International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 07 | July 2022 www.irjet.net p-ISSN: 2395-0072

Job Scheduling Mechanisms in Fog Computing Using Soft Computing Techniques.

[1]Mehraj Maqbool, [2]Harwant Singh Arri

[1]M. Tech, School of Computer Science and Engineering, Lovely Professional University, Punjab
[2]Assistant professor, Dept. of computer science and engineering, Lovely Professional university, Punjab

---

Abstract-*The internet of things may be a flexible and emerging technology, it's large and complicated network of devices during which fog plays a crucial role, and fog nodes handle the information flow of such an oversized and sophisticated network, because the cloud and fog provides the on-demand virtualized resources like computing resources and storage resources to its users. To allocate the resources to the IoT tasks, dynamic and efficient load balancing algorithms is wont to improve the general performance and reduce energy consumption, i.e. the target is to a way to allocate the suitable resources to the tasks and dispatching the computing tasks from the available resource pool. the choice may affect the execution time, energy consumption, load distribution and can affect the value incurred by the user .this paper specialize in the dynamic workflow scheduling mechanisms in fog computing and can give the comparison of various workflow algorithms on the idea of various parameters and can classify the various workflow and task scheduling algorithms and at the top of the paper we'll determine the realm of research and supply some directions of future work.*

Key words**: fog computing, cloud computing, PSO algorithm, Monkey algorithm, genetic algorithm.**

1. INTRODUCTION

Many Technologies are on trend these days , In the same way IoT has gained a much market and has become an important and core of many applications, there are many application where IoT has gained the popularity like ,Smart grids , smart health care ,5G connectivity ,transportation and supply chain smart cities ,wearables ,Traffic monitoring, Agriculture ,Hospitality ,and Maintenance management . According to one of the research work done by Abbasi and sha [1], 50 billion devices would be connected to the internet by 2020 and if this number can be divides it will gives us that about 6.38 devices would be using per user in 2020. And the number would by changing rapidly and will reach 500 billion devices by 2025. The IoT device may be anything which can share, process, generate data, and have a connectivity feature is called IoT device [2]. As when these heterogeneous devices will be connected they will produce a huge amount of data, to handle this large number of data we need a large capacity of storage, bandwidth and computational power. Keeping all the requirements of IoT environment into consideration the organizations developed the infrastructure known as cloud computing. American psychologist and computer scientist J.C.R. Licklider (Joseph Carl Robnett Licklider) developed the idea of cloud computing in the early 1960s. In an effort to connect people and data globally, he worked on the ARPANet (Advanced Research Project Agency Network) and introduced the Cloud Computing concept that is so well-known today. [3] .IN cloud computing the data centres are geographically centralized, and the IoT devices taking services from these data cloud. As the data centres are located at different fixed locations ,to handle all the data of IoT application which is in zeta bytes makes some of the IoT application slow ,therefore managing and processing such a huge data was a challenging and facing a lot of issues with the applications which needs a real time processing such as heath ,traffic monitoring transportations and wearables ,due to these challenges the organisation move into the new network infrastructure which overcomes the loopholes of the cloud computing which is called the Fog computing.

Fog Computing was first initiated by Cisco to extend the cloud computing to the edge of the network [4].Thus enables a new breed of applications and services. the motivation behind the Fog computing was that it should not incur with the loop holes which was in cloud computing, and it should reduce the burden of processing in cloud computing ,it should reduce the high latency which the cloud takes for processing the data ,so that the applications which needs the real-time responses will operate smoothly. It should also reduce the distributed environmental complexity [5].The fog devices would be installed closer to the end devices so that the most of the data processing would be done at these devices which will in turn will reduces the burden on cloud computing and will reduces the bandwidth, computing and storage of the cloud [6] .to adopt these features in fog computing architecture most of the IoT applications can use fog computing services, like health care centres ,traffic monitoring systems ,smart grids and wearables etc.

Alike of cloud computing fog is a decentralized infrastructure, where fog nodes can join and leave the network arbitrary [7]. As the fog nodes are closer to the IoT end devices by this fog computing reduces the latency which was a challenge in cloud computing,apart from this fog nodes can give a local storage and can also some of the computation locally which can enhance the efficiency of applications which needs the real time response, by giving these services closer distance fog cannot replace the cloud as the fog devices are contained by resources like storage capacity ,process power , and energy . Therefore, the fog is a supplement to cloud computation rather than a replacement for it. The connection between fog and cloud is important [8]. fog nodes gives real time processing and pushes the data into the cloud for storage ,as most of the process is done at fog nodes the only useful data is send towards the cloud for storage, which will use the overall bandwidth ,processing and storage efficiently .

From the above discussion, We have stated that fog is a complement to cloud computing rather than a replacement for it., but it can just leverage the efficiency to the time sensitive applications, the computational operations of time sensitive applications are carried out at the fog layer, through the resources which are available at fog nodes .by doing this the latency of the applications is reduced and the bandwidth is saved.

It will use less energy by using fog computing. In fog computing, the design will utilise the potential and resources present in fog devices to the greatest extent possible; but, if we require efficient resources that are not present in the fog environment, we will transfer to the cloud, with associated higher costs. Continued usage of fog resources will result in lower prices, shorter wait times, higher levels of security and confidentiality, less network traffic, and greener computing.

In this architecture the fog layer nodes will provide, resources to the IoT devices [9][10] .by providing these resources in an efficient way the resource scheduler and manager come into picture, it would be the responsibility of the scheduler which will provide the resources in an efficient manner. When IoT devices submit the requests to the fog nodes ,the scheduler will provide the approximate resources to the tasks[11] After assigning the jobs, the fog nodes will either be overcrowded or under loaded. [12][13]so we should have a dynamic and efficient load balancing algorithms which will increase the overall efficiency ,performance and reduce the power energy consumption.

Different optimization strategies have been suggested in order to solve the scheduling algorithms in a fog cloud setting. Researchers have categorised optimization approaches in a number of ways, with deterministic and stochastic optimization being one of them. In deterministic optimization, the data for the given issue are precisely known. But occasionally, the data cannot be known with absolute certainty for a number of reasons. A straightforward measuring error can be to blame. Another reason is that some statistics describe future knowledge, which makes it impossible to know with certainty. Stochastic optimization is used in optimization under uncertainty when the uncertainty is built into the model.

The key contributions of this study are highlighted as follows:

• Introducing a various optimization techniques based on metaheuristic optimization algorithms in fog cloud environment

• Classification of many methods that have been proposed on the fundamentals of job, task, workflow, and resource scheduling.

• Discussing and comparing the existing scheduling algorithms on the basis of performance and efficiency.

The remainder of the document is structured as follows: The concept of scheduling algorithms is covered in Section 2. The Concepts and Methods of Optimization are provided in Section 3. Section 4 presents a review of related research on scheduling algorithms. Discussion and comparison in Section 5. Conclusion and Research Gaps in Section 6. Citations for Section 7.

2. CONCEPTS

This section outlines some key terms used in this research and provides a brief explanation of the logical justifications that drive research toward fog computation.

As IoT has changed very fast ,it is now present in everywhere in our surroundings stated from universities ,vehicles ,homes ,health care centres smart grids ,traffic monitoring systems and wearables etc and is in fact increasing continuously[14][15]

,but these IoT devices have not too much capacity to store large data and cannot perform complex computational operations, because as if they will perform by its own they will consume energy very soon ,therefore there is a need to offload the task where needs a intense resource outs the higher level layers with more resources [16] there is where fog and cloud come into picture[17][18].

Fog is a decentralized computational scheme which is between IoT devices and cloud .before existence of fog computation architecture, IoT devices were taking services directly from cloud ,but Fog computing is being used, which raises serious concerns because the cloud was unable to meet the needs of the majority of IoT applications because cloud data centres were geographically centralised, which dramatically reduced QoS [19][20]. According to the definition of fog provided by Cisco, fog is a geographically dispersed computer architecture that provides a pool of resources at the network's edge that are not dependent on the cloud and are used for elastic computation, archiving, and computation [21]. As a result, the goal of fog computing is to locally offer cloud-like services to serve IoT applications, particularly delay-sensitive applications. To accomplish these goals and enhance the effectiveness of IoT devices and fog computing

There are five types of scheduling mechanisms as:

1. Resource scheduling.
2. Task scheduling.
3. Resource allocation.
4. Workflow scheduling.
5. Job scheduling.

2.1 Resource scheduling

The objective of the resource scheduling is to provide the best machine resources for the customers to obtain best scheduling goal, When scheduling resources, the scheduler aims to increase resource utilisation, cut down on delays, and improve service quality. QoS [23]

2.2 Task scheduler:

The goal of task scheduling is to distribute a set of jobs among the available fog nodes in order to satisfy the quality of service QoS requirement while optimizing task execution and transmission times [24].

2.3 Resource allocation:

The goal of resource allocation is to match the available resources to client needs over the internet in a systematic manner. [25].

2.4 Workflow scheduling:

The goal of workflow scheduling is to assign computing resources with various processing powers to work flow application jobs in order to reduce make time and cost [26]. An application's price includes its computational, data transfer, and storage costs.

2.5 Job scheduling

The goal of task scheduling is to give a collection of jobs to the least amount of fog resources so that they can be completed in the least amount of CPU time.

3. OPTIMIZATION

In the simple sense optimization is a procedure of maximizing or minimizing the objective function problem, so for a simple function like optimization is a simple task, and can be done in an easy way. but in case a function is a nonlinear multimodal function, multivariate like the task scheduling in fog computing[] the optimization is not an easy task, this problem may lead to many challenges and cannot be solved using the traditional methods. For those problems, efficient algorithms needed to be proposed.

Optimization Algorithms.

There have been numerous optimization algorithms developed, and one of the categories is as deterministic or stochastic. Efficient search methods are required to handle the optimization issues of fog clouds. Deterministic algorithms are those that operate mechanically deterministically, without the use of chance, and will always arrive to the same conclusion if they begin from the same starting point. Examples of this type of algorithm are hill climbing and downhill. However, stochastic algorithms, which might achieve various conclusion positions even while utilizing the same starting point, have some element of randomness. Examples of stochastic algorithms are genetic and PSO algorithms.

Metaheuristic Algorithms with stochastic components are often referred to as a heuristic in the past, as recent literature tends to refer to them as metaheuristic since all nature-inspired algorithms proposed for optimization is called meta heuristic algorithms according to glovers convention [].Metaheuristic algorithms may be thought of as a master strategy that directs and changes other heuristics to yield answers beyond those that are typically generated in a hunt for local optimality. Literally, the heuristic means to find out to discover the solution by hit and trial approach. There is a particular trade-off that all metaheuristic algorithms employ in order to find solutions to randomization issues in an acceptable period of time, but there is no assurance that the algorithms will always succeed in doing so. Exploitation and exploration, also known as intensification and diversification, are two key elements of every metaheuristic algorithm. Exploitation is the search's local emphasis, while exploration involves producing a variety of answers to explore the search space globally.

Different Meta heuristic approaches for optimization in fog cloud environment.

3.1 PSO Algorithm (Particle Swarm Optimization)

Kenned and Earhart created PSO, an evolutionary computational method, in 1995. Given that the algorithm was influenced by a flock of birds, it is a Meta heuristic. Being a meta-heuristic algorithm, the PSO starts with a population of random solutions, assigns a random velocity to each random solution, and then sends the possible solutions, known as particles, flying around the problem space. Each particle records its coordinates in the issue space at each stage, along with the best solution (fitness). It has so far succeeded. The global version of the particle swarm optimizer tracks another best value in addition to the fitness value, which is also saved and is known as pbest.

1. In the issue space, initialize the population (array) of particles with random positions and speeds on a d-dimensional grid.
2. Assess the fitness function and goal function in d dimensions.
3. If the present position in d-dimensional space is preferable, update the pbest and pbest location and compare the particles fitness evaluation with the particles pbest.

4. If the new evaluation of fitness is better than the population's prior best, update the gbest.

5. Modify the particle's location and velocity in accordance with Equations (1) and (2).

$$Vi(t+1) = wvi(t) + c1r1\,(pi - xi(t)) + c2r2\,(pg - xi(t)). \qquad \ldots\ldots\ldots\ldots 1$$

$$x(t+1) = x(t) + vi\,(t) \qquad \ldots\ldots\ldots\ldots 2$$

6. Repeat step 2 until a requirement is satisfied.

In the velocity equation of PSO there are some parameters which users can set according to the problem statement and different

3.2 GENETIC ALGORITHM

John Holland initially presented genetic algorithms [1], which are search and optimization algorithms based on the principles of natural evolution, in 1970. By replicating the development of species through natural selection, genetic algorithms also carry out the optimization procedures. Typically, a genetic algorithm consists of two steps. The first step is choosing an individual to produce the next generation, and the second is using crossover and mutation procedures to manipulate the chosen person to produce the following generation [2]. Which individuals are chosen for reproduction and how many children each selected person produces are determined by the selection process. The fundamental tenet of selection strategy is that a person's likelihood of having children increases with their level of competence.

Using a genetic algorithm, a population of starting people is evolved.

3.2.1 Selection operation: is to select elitist individuals as a parents in current population which will be used for further generation, for the selection procedure the fitness values are used as a criteria to judge whether individuals are elitist ,various approaches are used to select the individuals eg Boltzmann selection, tournament selection ,roulette wheel selection.

3.2.2 Crossover operation: The cross over operator may create two new children from two parent strings by copying two selected bits from each parent. In a genetic algorithm, the generation of successors is governed by a series of operators that recombine and alter selected members of the current population.

3.2.3 Mutation operation: In addition to the recombination operation, which creates offspring by fusing elements of two parents, there is another type of operator that creates offspring from a single parent, particularly the mutation operator, which creates small random changes to bit strings by crossing a single bit at random and changing its various values after crossover.

3.3 MONKEY ALGORITHM

The basic monkey algorithm, which has four processes (initialization process, climb process, watch-jump process, and somersault process), was the inspiration for the monkey algorithm, which is an improved version of the basic algorithm. In this improved version, a fifth process called random perturbation is added. The algorithms function as

3.3.1 Initialization process: Since M denotes population size (number of monkeys), it creates a random position for each monkey, with ith being as follows: Xi=(i=1,2,3,4,5...M)for each monkey as xi =(x1,x2.....,xm) (1)

After that the position of each monkey is evaluated in objective function.

3.3.2 Climb process: in this the monkey changes its position from initial position step by step that can make improvements in the objective function , The following equation describes the length of the monkey's stride while shifting positions:

$$\Delta x_{ij} = \begin{cases} a, & with\ probability\ 1/2 \\ -a, & with\ probability\ 1/2 \end{cases}$$

Where a is positive no step length in the climb process with a=10-3, xij is updating the location of the monkey (j=1, 2, 3, 4,..., n).

3.3.3 Watch-jump process: This procedure inspects each monkey's position following the climb. It determines whether or not their position has reached the top by having each monkey scan the area for a position that is higher than their present one. If they find one, they will leap from it; if not, it implies that their position has not reached the top.

3.3.4. Somersault process: They will locate locations near the barycentre of all monkeys' existing positions, which is described as a pivot, and monkeys will tumble along the direction leading to the pivot thanks to this method for finding new positions (searching domain).

4 RELATED WORKS

Task or Workflow scheduling is an NP-hard issue in a distributed computer system, as is well known [27]. To ensure that programmes run as efficiently and automatically as possible, scheduling must be done. There are several studies that are now available on task or workflow scheduling in a distributed context, with task scheduling under cloud computing platforms receiving the greatest attention [28][29].

The author in [35] has implemented an efficient algorithm that can minimize the energy consumption in IoT workflow on heterogeneous fog computing architecture in this work the integer linear programming model is built and designed a performance effective algorithm based on the model. It takes into account the two types of edge nodes, of which one has greater performance and the other uses less power. The experiment's findings demonstrate that, when compared to the Longest Time First (LTF), Integer Linear Programming (ILP), and Random Algorithm, the EMS algorithm may provide the best energy usage.

[38]. In this paper the author proposed A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments it takes into account both computational cost and data transmission cost, the heuristic function optimizes the cost of the task resources mapping based on the solution given by the particle swarm optimization technique, the mapping optimizes the overall cost of computing and workflow application.

[30]. The author of Improved Particle Swarm Optimization Based Workflow Scheduling describes our scheduling solution for Workflow applications that is based on IPSO and primarily focuses on Workflow scheduling in a cloud-fog environment. The original PSO's change in inertia weight is employed as an unique nonlinear decreasing function in IPSO. The goal is to balance and adapt each particle's capacity to seek during the search process. Finally, taking into account the actual issue of Workflow applications in a cloud-fog environment, a scheduling method is developed.

[33]. The proposed algorithm, called the "bee life algorithm," is a type of job scheduling algorithm. Its goal is to find the best way to distribute a set of tasks among available fog computing nodes in order to achieve the best possible trade-off between CPU execution time and allocated memory needed by fog computing services set up by mobile users. When compared to other algorithms like the genetic algorithm (GA) and PSO particle swarm algorithm, its efficiency is improved in terms of execution time and the allocated memory. In this algorithm, they used two performance metrics to evaluate CPU execution time and the total amount of memory needed to complete execution in a fog environment.

[39]. for business workflows that require a lot of parallel instances and are instance-intensive, suggest a nearly ideal dynamic priority scheduling (DPS) technique. [31]. The HH algorithm, or hyper-heuristic algorithm, uses machine learning techniques to choose, combine, produce, or match several sample heuristics in order to solve computational search problems. The suggested approach locates the ideal response to the workflow scheduling issues. In comparison to PSO, ACO, and SA, this HH algorithm provides the best average energy usage. By providing resources to users with precise limits, this strategy shortens simulation time and saves energy. It also gives users more control over resource allocation.

[36]. the goal of this [paper] is to decrease the average response time and to optimize resource utilization efficiently by scheduling tasks and managing fog nodes that are available. The author proposed a novel bio-inspired hybrid algorithm in which the resources are managed according to the incoming requests of the users. ABIHA is a hybrid of modified particle swarm optimization and modified cat swarm optimization MCSO. In this algorithm MPSO algorithm scheduled the tasks among the fog nodes and the hybrid of MPSO and MCSO are managing the resources at the fog device level.

[34]. the suggested methodology, called Dynamic Energy-Efficient Resource Allocation, is the method for distributing the workload among fog infrastructure. The tasks that the user submits are handled by the task manager and passed to the resource information provider RIP, which will register the resources for the tasks. The third step is resource scheduler, which collects information about the tasks from task manager and also gathers information about the resources from RIP. Tasks and resources are then sent to the resource engine, which assigns tasks to the resources as per sorted list, resource load manager examines the resource status during execution and transfers the status to resource power manager which in turn manages the resource on/off power status. The result of the proposed algorithm is compared with the DRAM result shows that it reduces the energy consumption and computational cost by 8.67% and 16.77%.

[40]. in this paper the author has proposed an algorithm called DAPSO (Dynamic adaptive particle swarm optimization algorithm) To improve the functionality of the fundamental PSO method, they have optimize the parameters run time of tasks by reducing the make span of a particular task set and it has also taken into consideration the resource utilization ,which is the maximizing the algorithm is need to cloud computing to schedule the independent task ,in this paper the author has taken the amalgamation of CROOKOO search algorithm and DAPSO and proposed the MDAPSO ,in this algorithm , They have found a solution to the PSO affinity problem where more inertia weight helps with global searches and less inertia weight helps with local searches.

[41].In this paper they developed a task scheduling technique for IoT requested in cloud fog environment ,this algorithm is depend on a modified artificial-based optimization (AEO) .the modification is an attempt to enhance the exploitation ability of AEO to get the optimal solution for the problems under investigation ,the algorithms is called as AEOSSA , this modification is developed using the operators of the salp swarm algorithm(SSA) ,the proposed algorithms is compared with other proposed algorithm of metaheuristic on the bases of performance metric and the proposed algorithm shows better results as compared to the other methods accordingly to the performance metrics like make span time ,throughput etc.

[42].This proposed algorithm solves the job scheduling problems in cloud computing by modifying the basic PSO approach, in this algorithm they have removed some components of the basic PSO algorithm and analyze the effect of that components in performance of algorithm, they removed the personal memory term from the velocity update formula and reinforce the case for PSO being mostly reliant on social interaction rather than personal experience.

The velocity equation of modified basic PSO approach is:

$V_i(t+1) = WV_i(t) + C_2X_2(p_g - x_i(t))$

Rather than

$V_i(t+1) = WV_i(t) + C_1R_1(p_i - xi(t)) + C_2X_2(p_g - x_i(t))$

5 DISCUSSION AND COMPARISON:

This section will give brief comparisons of various algorithms of fog computing and we will discuss some important insights of the workflow algorithms, resource allocation, task scheduling, and job scheduling algorithms Table 1 shows the comparison of different algorithms on the basis of various evaluation parameters like energy consumption, cost, response time, make-span, bandwidth utilization, memory allocated and data transmission cost. From the given table most of the algorithms tried to solve the problem of make span, energy consumption, and cost but only a few have taken the response time , memory allocation and data transmission cost into consideration.

In table 2 we have compared the various algorithms on the basis of environment i.e. for which infrastructure the algorithms are built may be fog or cloud and which algorithm is what type whether job scheduling, resource allocation, workflow scheduling, or task scheduling, and from which algorithms are performing better when compared with the existed once.

In table 3 we have given the simulation techniques of each algorithm whether ifogsim , MATLAB, CloudSim, or c++ and for each algorithm, we have figured out some advantages and disadvantages which will give us the picture of the future work .advantages are discussed on the basis of different parameters of evaluation when compared with other algorithms.

As a result, workflow scheduling in fog is a crucial study field that has to be pursued. The fog plays a significant role in the implementation of IoT scheduling algorithms. According to the process scheduling algorithms, new approaches to maximize make-span, cost, energy consumption, resource usage, or bandwidth are required. Due to the fact that time-sensitive IoT applications employ fog computing. The overall execution time (make-span) that the jobs require is a crucial element. Additionally, because the fog has limited resources, energy wastage and battery life have become more of a problem. By employing an effective strategy for data scheduling, these measurements may be preserved in the fog infrastructure.. Other metrics and applications, such as dynamic task scheduling, periodic tasks, task migration between edge nodes, heterogeneous fog nodes, applications with soft or hard deadlines, virtual machine migration, and energy consumption in fog nodes, must be taken into account by the researchers in order for their new proposed scheduling algorithm to be effective.

Table-1: Comparison of various algorithms on the basis of different parameters

Algorithms	**Make-Span**	**Cost**	**Bandwidth Utilization**	**Response Time**	**Energy Consumption**	**Allocated Memory**	**Data Transfer Cost**
HH	YES	YES	NO	NO	YES	NO	NO
EMS	NO	NO	NO	NO	YES	NO	NO
PSO	NO	YES	NO	NO	NO	NO	YES
IPSO	YES	YES	NO	NO	NO	NO	NO
BLA	YES	NO	YES	NO	NO	YES	NO
ERA	NO	NO	NO	YES	NO	NO	YES
MARKET	YES	YES	NO	NO	NO	NO	NO
DEER	NO	YES	NO	NO	YES	NO	NO
MDAPSO	YES	NO	YES	NO	NO	NO	NO
AEOSSA	YES	NO	NO	NO	NO	NO	NO
SPSO	YES	NO	NO	NO	NO	NO	NO

Table-2: Comparison of various algorithms on the basis of environment, scheduling type and result compared

Reference	Algorithm	Scheduling Type	Parameter Taken Into Consideration	Environment	Result Compared
[31]	HH (Hyper-heuristic)	Workflow-scheduling	Make-span, usage of energy network usage, cost	fog	Particle swarm optimization, genetic algorithm, ant colony optimization,
[35]	EMS (Energy Efficient Scheduling)	Workflow scheduling	Energy consumption	fog	Longest first time ,ICP, random algorithm
[38]	PSO (particle swarm optimization)	Workflow scheduling	Computational cost ,data transmission cost	Cloud	Best resource scheduling
[30]	IPSO (improved particle swarm optimization)	Workflow scheduling	Cost and make span	Fog-cloud	particle swarm optimization
[33]	BLA (Bees Life Algorithm)	Job scheduling	Make span ,allocated memory	fog	particle swarm optimization , genetic algorithm
[36]	NBIHA (Novel Bio-inspired Hybrid Algorithm)	Task scheduling and resource allocation	Make span ,response time, energy consumption	fog	First come first serve, SJE,MPSO
[32]	ERA (Efficient Resource Allocation)	Resource allocation	Response time, data transmission expense, and bandwidth use	fog	RDLB,ORT
[37]	Market-oriented hierarchical Strategy)	Workflow scheduling	Cost, make span, cpu time	Cloud	PSO, genetic algorithm
[34]	DEER (Dynamic Energy Efficient Resource Allocation)	Resource allocation	Energy consumption computational cost	fog	DRAM
[40]	MDAPSO (Modified dynamic adaptive particle swarm optimization)	Task scheduling	Make-span ,Resource utilization	Cloud	PSO
[41]	AEOSSA (Eco-system based and salp swam algorithm)	Task scheduling	Make-span and Throughput	Cloud-fog	AEO ,SSA ,PSO
[42]	SPSO(Simplified particle swarm optimization)	Job scheduling	Make-span	Cloud	GA ,PSO

Table-3: Advantages and Limitations of various algorithms and simulated tools

Reference	Algorithm	Simulated by	Advantages	Limitations
[31]	HH	ifogSim	HH algorithm improved the average energy consumption as compared to PSO , ACO , SA algorithm by 69.99 cost to cost 59,62%	In this algorithm VM (virtual machine) migration concept is not address. It has low scalability.
[36]	NBIHA	ifogSim	First come, first served, shortest task first, and modified particle swarm optimization methods are inferior to the suggested algorithms in terms of execution time, energy consumption, and average response time.	Communication cost is not addressed, the intend to use reinforcement learning techniques to managing resources in fog IoT environment.
[35]	EMS	SPEC CPU 2006	EMS algorithm uses less energy than the longest job first, integer linear programming, and random algorithms.	Not taking different nodes into consideration it taken only two types of nodes. No consideration is given to task migration across edge nodes. It does not consider periodic task which should be finished within their periods
[30]	IPSO	MATLAB 2016a	The overall completion time and economical cost is reduced as compared to PSO	It takes only two metrics into consideration i.e. computational and economical cost. It gives the trade-off between make span and cost
[34]	DEER	CloudSim	In comparison to DRAM, the DEER is effective at load balancing to cut down on energy usage and computational costs by 8.67 percent and 1.677 percent, respectively. There is no overhead for sorting.	In this proposed algorithm there is no fault tolerance mechanism implemented. High complexity.
[33]	BLA	C++	When the proposed algorithm is compared with particle swarm optimization and genetic algorithm it performs better in terms of processing speed and memory allotted	The proposed algorithm is tested on small dataset and it has low scalability. The dynamic scheduling technique is not addressed, and the task execution reaction time is long.
[32]	ERA	Cloud Analyst	The proposed algorithm shows better optimized way of resource allocation when compared with RDLB,ORT and in terms of overall response time ,data transfer cost and better bandwidth utilization in the fog computing environment	Resources are only allotted to users who have requested them prior to processing because the needs for the resources during the execution of the request are not satisfactory. It only applies to instances that have been reserved. poor accessibility.

				It is really complicated. and lengthy execution
[38]	PSO	JSwarm package	The computational cost and communicational cost savings are 3 times more in PSO when compared with the best resource selection algorithm BRS . It gives good work load distribution onto resources.	It does not work well in scheduling workflow of real applications.
[40]	MDAPSO	CloudSim 3.3	Results show that it outperforms the default PSO, ADPSO, and PSOCS algorithms by 38.63 and 25.30 percent, respectively.	It implementation is complex as it the combination of two algorithms
[42]	Simplified PSO	CloudSim	Easy to implement	It didn't taken consideration most the performance parameters

6. CONCLUSIONS

The workflow scheduling in fog computing is one of the most significant contemporary scheduling methods that is thoroughly reviewed and analyzed in this study. The best scheduling methods are picked after a thorough reading and analysis of the majority of the recent publications released on scheduling algorithms in fog computing. Task scheduling, resource scheduling, resource allocation, job scheduling, and workflow scheduling are the five key categories used in this research to group the scheduling methods. From the comparison outcome, the majority of researches focused on energy usage and make-span. However, other factors like cost, reaction time, and memory allocation are the most important metrics that researchers must take into account in order to optimize the efficiency of IOT applications. Additionally, because the resources available to the fog infrastructure are few, make-span and energy usage are crucial parameters in the fog computing scheduling algorithms.

REFRENCES

[1] B. Z. Abbasi and M. A. Shah, "Fog computing: Security issues, solutions and robust practices," 2017 23rd International Conference on Automation and Computing (ICAC), 2017, pp. 1-6, doi: 10.23919/IConAC.2017.8082079.

[2] https://www.oracle.com/in/internet-of-things/what-is-iot/

[3] https://www.educba.com/history-of-cloud-computing/

[4] Bonomi, Flavio & Milito, Rodolfo. (2012). Fog Computing and its Role in the Internet of Things. Proceedings of the MCC workshop on Mobile Cloud Computing. 10.1145/2342509.2342513.

[5] XiaoM, ZhouJ, LiuX, JiangM.A hybrid scheme for fine grained search and access authorization in fog computing environment. Sensors. 2017;17:1423.

[6] TariqN, AsimM, AlObeidatF ,et al. The security of big data in fog enabled IoT applications including block chain: asurvey.Sensors. 2019;19:1788.

[7] Mohanta, Bhabendu & Jena, Debasish & Panda, Soumyashree & Gountia, Debasis. (2019). Decentralized Secure Fog Computing in Cloud-Fog-IoT Infrastructure Using Blockchain.

[8] Dastjerdi AV, GuptaH, Calheiros RN,Ghosh SK,Buyya R.Fog computing: principles, architectures, and applications. Internet of Things. Netherlands: Elsevier; 2016:61-75.

[9] M.OguraandV.M.Preciado,‘‘Stability of spreading processes overtime varying large-scale networks,'' IEEE Trans. Netw. Sci. Eng., vol. 3, no. 1, pp. 44–57, Jan. 2016.

[11] H. Yuan and M. Zhou, ‘‘Profit-maximized collaborative computation offloading and resource allocation in distributed cloud and edge computing systems, ''IEEET rans. Automat. Sci. Eng., EarlyA ccess, Jul.14,2020,doi: 10.1109/TASE.2020.3000946.

[12] Y. Kong, M. Zhang, and D. Ye, ‘‘A belief propagation-based method for task allocation in open and dynamic cloud environments,'' Knowl.-Based Syst., vol. 115, pp. 123–132, Jan. 2017.

[13] Li, Peng, Shengli Zhao, and Runchu Zhang, ‘‘A cluster analysis selection strategy for supersaturated designs,'' Comput. Statist. Data Anal., vol. 54, no. 6, pp. 1605–1612, 2010

72 K. Matrouk and K. Alatoun / International Journal of Networked and Distributed Computing 9(1) 59–74

[14] Muralidharan S, Roy A, Saxena N. MDP-IoT: MDP based interest forwarding for heterogeneous traffic in IoT-NDN environment. Future Gen Comp Syst. 2018;79:892-9

[15] Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gen Comp Syst. 2013;29(7):1645-1660.

[16] Díaz M, Martín C, Rubio B. State-of-the-art, challenges, and open issues in the integration of Internet of Things and cloud computing. J Network Comp Appl. 2016;67:99-117.

[17] Botta A, De Donato W, Persico V, Pescapé A. On the integration of cloud computing and Internet of Things. In: 2014 International Conference on Future Internet of Things and Cloud. IEEE; 2014, August:23-30

[18] Roca D, Milito R, Nemirovsky M, Valero M. Tackling IoT ultra large scale systems: fog computing in support of hierarchical emergent behaviors. In: Fog Computing in the Internet of Things. Cham: Springer; 2018:33-48.

[19] Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R. Fog computing: principles, architectures, and applications. In: Internet of Things. Morgan Kaufmann; 2016:61-75.

[21] Cisco Systems. Fog computing and the Internet of Things: extend the cloud to where the things are. 2016. [Online]. Available: http:// www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computingoverview.pdf

[22] Mahmud R, Srirama SN, Ramamohanarao K, Buyya R. Quality of experience (QoE)-aware placement of applications in fog computing environments. J Parallel Distrib Comp. 2018:190-203.

[23] G. Li, Y. Liu, J. Wu, D. Lin, S. Zhao, Methods of resource scheduling based on optimized fuzzy clustering in fog computing, Sensors (Basel) 19 (2019), 2122.

[24] Z. Hao, E. Novak, S. Yi, Q. Li, Challenges and software architecture for fog computing, IEEE Internet Comput. 21(2017), 44–53.

[25] S. Agarwal, S. Yadav, A.K.Yadav, An efficient architecture and algorithm for resource provisioning in fog computing, Int. J. Inform.Eng.Electron.Bus. 8 (2016), 48–61.

[26] R. Xu, Y. Wang, Y. Cheng, Y. Zhu, Y. Xie, A.S. Sani, et al., Improved particle swarm optimization based workflow scheduling in cloud-fog environment, International Conference on Business Process Management, Springer, Cham, 2018, pp. 337–347.

[27] Jennings,D.;Sabbagh,M.;Pfefferer,C.;Fristch,A.;Russell,D.;Seibyl,J.;Marek,K.;Reininger,C.Aframework for resourceal location strategies in cloud computing environment. In Proceedings of the Computer Software & Applications Conference Workshops, Munich, Germany, 18–22 July 2011

[28] Wu, Z., Liu, X., Ni, Z., et al.: A market-oriented hierarchical scheduling strategy in cloud workflow systems. J. Supercomput. 63(1), 256–293 (2013)

[29] Verma, A., Kaushal, S.: A hybrid multi-objective particle swarm optimization for scientific workflow scheduling. Parallel Comput. 62,1-19(2017).

[30] Xu, Rongbin & Wang, Yeguo & Cheng, Yongliang & Zhu, Yuanwei & Xie, Ying & Sani, Abubakar & Yuan, Dong. (2019). Improved Particle Swarm Optimization Based Workflow Scheduling in Cloud-Fog Environment: BPM 2018 International Workshops, Sydney, NSW, Australia, September 9-14, 2018, Revised Papers. 10.1007/978-3-030-11641-5_27.

[31] S. Kabirzadeh, D. Rahbari and M. Nickray, "A hyper heuristic algorithm for scheduling of fog networks," 2017 21st Conference of Open Innovations Association (FRUCT), 2017, pp. 148-155, doi: 10.23919/FRUCT.2017.8250177.

[32] Agarwal, Swati et al. "An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing." *International Journal of Information Engineering and Electronic Business* 8 (2016): 48-61.

[33] Salim Bitam, Sherali Zeadally & Abdelhamid Mellouk (2018) Fog computing job scheduling optimization based on bees swarm, Enterprise Information Systems, 12:4, 373-397, DOI: 10.1080/17517575.2017.1304579

[34] A. U. Rehman et al., "Dynamic Energy Efficient Resource Allocation Strategy for Load Balancing in Fog Environment," in IEEE Access, vol. 8, pp. 199829-199839, 2020, doi: 10.1109/ACCESS.2020.3035181.

[35] H. -Y. Wu and C. -R. Lee, "Energy Efficient Scheduling for Heterogeneous Fog Computing Architectures," 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), 2018, pp. 555-560, doi: 10.1109/COMPSAC.2018.00085.

[36] H. Rafique, M. A. Shah, S. U. Islam, T. Maqsood, S. Khan and C. Maple, "A Novel Bio-Inspired Hybrid Algorithm (NBIHA) for Efficient Resource Management in Fog Computing," in IEEE Access, vol. 7, pp. 115760-115773, 2019, doi: 10.1109/ACCESS.2019.2924958.

[37] Wu, Zhangjun & Liu, Xiao & Ni, Zhiwei & Yuan, Dong & Yang, Yun. (2011). A market-oriented hierarchical scheduling strategy in cloud workflow systems. The Journal of Supercomputing. 63. 1-38. 10.1007/s11227-011-0578-4.

[38] S. Pandey, L. Wu, S. M. Guru and R. Buyya, "A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," 2010 24th IEEE International Conference on Advanced Information Networking and Applications, 2010, pp. 400-407, doi: 10.1109/AINA.2010.31.

[39] Xu, Rongbin & Wang, Yeguo & Huang, Wei & Yuan, Dong & Xie, Ying & Yang, Yun. (2017). Near-optimal dynamic priority scheduling strategy for instance-intensive business workflows in cloud computing. Concurrency Computation. 29. 10.1002/cpe.4167.

[40] Attiya, Ibrahim & Zhang, Xiaotong. (2017). A Simplified Particle Swarm Optimization for Job Scheduling in Cloud Computing. International Journal of Computer Applications. 163. 34-41. 10.5120/ijca2017913744.

[41] Al-Maamari, Ali, and Fatma A. Omara. "Task scheduling using PSO algorithm in cloud computing environments." International Journal of Grid and Distributed Computing 8.5 (2015): 245-256.

[42] Mohamed Abd Elaziz, Laith Abualigah, Ibrahim Attiya, Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments, Future Generation Compute Systems,Volume 124,2021,Pages 142-154,ISSN 0167-739X, https://doi.org/10.1016/j.future.2021.05.026.