Task management in IoT-Fog-Cloud environment employing static scheduling Techniques

Gaurav Goel and Rajeev Tiwari

Abstract— In a distributed computing system, there are limited resources, which needs to be utilized effectively. Then for improving QoS Fog computing paradigm is an effective way, with suitable allocations. Thus, different resource scheduling and optimization algorithms exist. However, still, there is a scope to improve bandwidth, latency, energy consumption, and total communication cost in the Fog environment. In this work investigation is done to show significance of task management in such resource constrained environment. Various heuristics and meta-heuristic algorithms are evaluated using simulations, to show the task placement and their impacts by using 5 different Montage datasets from work flow sim tool kit for Fog-Computing environment. Then QoS parameters like cost, makespan, and energy consumptions are computed for various state-of-the-art techniques like Min-max, PSO, GA, ACO, and BLA. This shows the behaviour of these techniques with such different tasks and allocation environment configurations. Evaluated result parameters are collected and presented in the result section. This work shows the effectiveness of heuristics and meta-heuristics techniques to manage the tasks and their allocations in the Fog environment.

***Keywords*— Task management, IoT, Fog, Cloud, Resource Scheduling.**

I. INTRODUCTION

IoT devices are expanding rapidly with an increase in networking technologies. Large applications are being developed by a group of users. These applications require more computation, large resources, and an intelligence system. Resources available to execute this application are insufficient with cloud devices. The problem of latency, high cost, and high makespan is faced by users while transferring their application data to the cloud for computation [1].

To overcome these problems a Fog environment is introduced by CISCO. The Fog paradigm is providing the same environment as a cloud in contrast to computation by using Fog nodes. Fog nodes can be a router, switches, gateway, modems, etc. Fog nodes are heterogeneous and distributed with limited storage and computation capacity. However, some applications require high storage and computation capacities, for this cloud can be used at the backend. Today's Fog paradigm is used in many applications like health monitoring, traffic management, industries, and farming. The benefit of using a Fog environment in this application is that Fog nodes are established in proximity to the end devices. Because of this benefit problems of latency, and bandwidth can be reduced [2,3]. Many researchers have discussed the problem of optimization in this scenario. Many optimization algorithms have been designed to achieve Quality of Service (QoS) in the IoT-Fog-Cloud environment. Many researchers have focussed on the technique of task management but still, there is a problem with latency, bandwidth, and the cost is raised.

Manuscript received November 20, 2021; revised June 9, 2022.

Gaurav Goel , Department of Computer Science and Engineering, Chandigarh Group of Colleges, Landran (Mohali), India, and Research Scholar, University of Petroleum and Energy studies, Dehradun, India, (email: gaurav.coecse@cgc.edu.in)
Rajeev Tiwari, School of Computer Science, University of Petroleum and Energy studies, Dehradun, India, (email: errajeev.tiwari@gmail.com)

Digital Object Identifier (DOI): 10.53907/enpesj.v2i1.76

A. RESOURCE MANAGEMENT IN FOG COMPUTING

Resource distribution and planning is an important technology for managing resource utilization, and obtaining load balancing for the data centers. In the cloud paradigm, the resource allocation is done in a centralized manner, to balance the workload of running physical machines and to avoid congestion. But the cloud is placed away from the IoT devices, so the problem of Latency and Bandwidth arises [4] . In the fog computing environment, resource allocation is now days a bottleneck task. The Fog nodes are distributed desperately in the Fog environment, while in the cloud the computing nodes are distributed in a centralized data center. The resource requirements for the IoT applications are distinguished because the applications have different requirements of bandwidth, computing power, and storage capacity. Therefore, it is requisite to achieve resource allocation for the static resource requirements to attain the target of QoS. The purpose of resource planning is to find better resources as per the demand of applications for achieving lower processing delay [5].

B. HOW DO OPTIMIZATION ALGORITHMS IMPACT TASK MANAGEMENT?

Task management [6] is a strategy for allocating tasks to fog nodes/cloud servers efficiently. In the IoT-Fog Cloud environment number of tasks may be raised by devices, which further undergo computation. By efficient task management techniques, one can schedule the Fog/cloud resources successfully. Suppose "n=5 "is representing the number of tasks, which is generated by the IoT devices [7]. These five tasks need to allocate to fog devices or maybe to the cloud server for computation. Devices in the fog environment are highly heterogeneous. so, there is a requirement for a useful strategy that helps task management in the Fog environment. Following any designed task scheduling technique helps the efficient tasks management in Fog environment. If task management strategy is chosen wisely then QoS parameters like cost, makespan, energy consumption, latency [8], and security can be achieved.

Optimization algorithms iteratively search for optimal solutions from a set of solutions. By comparing a new solution with the previous one optimal solution can be achieved. Similarly, in a Fog environment, searching for the best machines for the allocation of tasks may occur. So that problem of task management can be overcome [9].

Many traditional optimization algorithms like Min-Min, Max-Min, PSO, GA, RoundRobin, and FCFS are the basis of later/upcoming optimization algorithms. Most of the researchers have provided Bio-inspired optimization algorithms like PSO, GA, Cat-Swarm, Moth-flame optimization algorithm, Dolphin Partner Optimization algorithm, Grey-wolf, firefly, Bees-swarm, Cuckoo, lion optimization algorithms, etc.

C. FOG COMPUTING ARCHITECTURE

Fig. 1 is showing the architecture of the IoT-Fog-Cloud environment. It consists of a three-tier structure which includes: the IoT device layer, Fog computing layer, and Cloud layer. The fog layer is in between the cloud layer and the IoT layer.

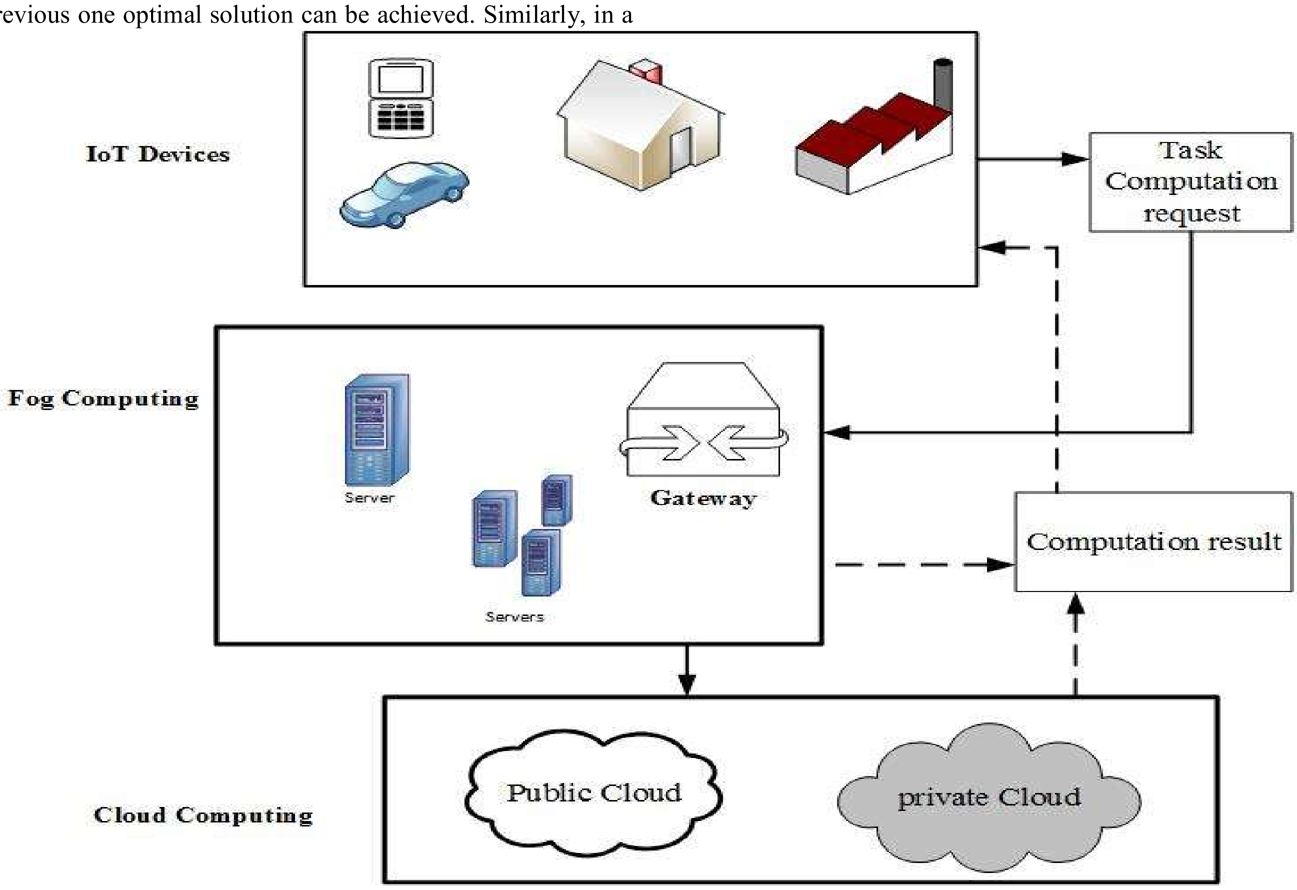

Fig. 1: Fog Computing Architecture

Cloud layer: Cloud layer consists of large computing servers and storage computation devices. Cloud can be any form like a public cloud for anybody access, it can be a private cloud for limited user access or it can be a hybrid cloud combination of both.

Fog layer: The fog layer consists of heterogeneous Fog nodes, having limited computation and storage capacity. It exists at the edge of the network, which helps it for improving bandwidth, computation cost, etc.

IoT device layer: This layer occupies the end-user interface, and includes devices and sensors. Sensors device collect user information. This information sends to the upper layer for processing.

D. THE MAJOR CONTRIBUTION OF THE WORK

1. Heuristic and meta-heuristic algorithms are investigated,

simulated and evaluated for QoS parameters using work Flow Simulator.

1. Investigation of 5 types of montage data set is done with all techniques, which depicts the performance execution with smaller to larger tasks in the system.
2. Comparative performance evaluation of state-of-the-art techniques is presented on cost, makespan, and energy consumption QoS parameters.

E. ORGANIZATION OF THE WORK

In this work, section II describes the related work of task management in the IoT-Fog-Cloud environment is done. Section III provides a taxonomy for Task Management optimization techniques to attain QoS parameters. In section IV description of simulation and parameter setting is provided. The result and discussion part are discussed in section V. A conclusion of this work is provided in section VI.

II. RELATED WORK

Salim Bitam et al. [10] described an algorithm based on Bees life named BLA. Researchers addressed the problem of Job scheduling in the Fog environment in their work. The authors discussed the trade-off between memory utilization and execution time of the CPU. The authors made a comparison of the proposed technique with PSO and GA and successfully outperforms both techniques in terms of memory and CPU execution time. Saniah Rehman et al. [11] proposed a technique of load balancing for efficient utilization of resources. Researchers used the technique of the Min-Min algorithm for efficient management. Resources are allocated first to those tasks which have minimum execution time as per the protocol of the Min-Min algorithm. The proposed approach is compared with the Round-Robin algorithm. The proposed technique outperforms the Round-Robin algorithm on parameter cost. Bushra jamil et al. [12] provided a technique of job scheduling for achieving optimal QoS parameters delay, energy consumption, and network usage. Researches provide a case study on health management to show the efficiency of the proposed algorithm. The authors achieved optimal results in comparison with the FCFS approach by 32% of delay and 16% of network usage. Mostafa Ghobaei-Arani et al. [13] described a technique of moth-flame for task scheduling in a Fog environment. By using this technique authors focussed on efficient task allocation to achieve optimal QoS requirements in the Fog environment. The objective function of work is to minimize the transfer time and task execution time. For showing the efficiency of work, the comparison is in contrast with PSO, BLA, and NSGA-2 techniques and validated the results of the proposed technique. Claudia Canali et al. [14] addressed the problem of distribution of data stream to fog nodes received from sensors. For the solution to this problem, the authors worked in two folds. The first optimization model considers not only the load on the fog nodes but also considers communication latency between fog nodes and sensors. second, a scalable genetic algorithm is proposed to address the problem. for the validation of the results, experiments were conducted on the smart cities problem. Narayana Potu et al. [15] proposed an extended PSO technique for optimizing resource scheduling in the Fog computing environment. Researchers used the extra gradient method in this technique to optimize the problem of task scheduling. the proposed technique is compared with TCaS, BLA, ideal PSO, and MPSO on parameter cost and makespan. Researchers have noticed the improvement in doing a comparison with existing techniques. Amit Kishor et al. [16] discussed the technique of task offloading by proposing a smart ant colony optimization algorithm. In this work, Researchers addressed the problem of latency when the task is going to offload from IoT devices to Fog nodes. The proposed technique is compared with modified PSO, BLA, RoundRobin, and throttled algorithm. The smart ant colony technique conquers all the existing techniques on parameter task offloading time. Fatma M. Talaat et al. [17] proposed a method of EPRAM (Effective prediction and resource allocation method) in the healthcare system. To control EPRAM, the authors concentrated on the method of Resource allocation method, Data processing method, and effective prediction method. These all techniques assist to preserve persons from high-threat diseases by the method of deep reinforcement learning and PNN. This designed method helps to reduce the makespan, and enhance load balancing and resource allocation. Noé Godinho et al. [18] describe the idea of services offloading and communication to the Fog environment for efficient QoS. In the proposed method researchers described the idea of MILP (Mixed integer linear programming) and mapping of VN to the network for leading energy and bandwidth in the Fog environment. The designed method successfully gets the optimal results. Jyoti Bisht et al. [19] proposed the method of extended min-min scheduling algorithm which assist the researcher to enhance makespan, cost, load balancing, and energy utilization in a Fog-edge environment. For the validation of the result, the designed method in contrast with the ELBMM & min-min algorithm, and the proposed method conquer both the techniques on the mentioned parameters.

III. TASK MANAGEMENT OPTIMIZATION TECHNIQUES TO ATTAIN QOS PARAMETERS

QoS in a system may be achieved through techniques of heuristic and meta-heuristic as shown in Fig. 2. Many researchers have provided various methods [20] under these categorized techniques. Min-Min, Max-Min, FCFS, Round-Robin, etc. fall under heuristic categories. Similarly, PSO, GA, BLA, ACO, etc. fall under Metaheuristic categories. Following heuristic and metaheuristic techniques, many researchers have done task management to attain QoS parameters like cost, makespan, time, response time, throughput, etc. in the Fog environment [21-23].

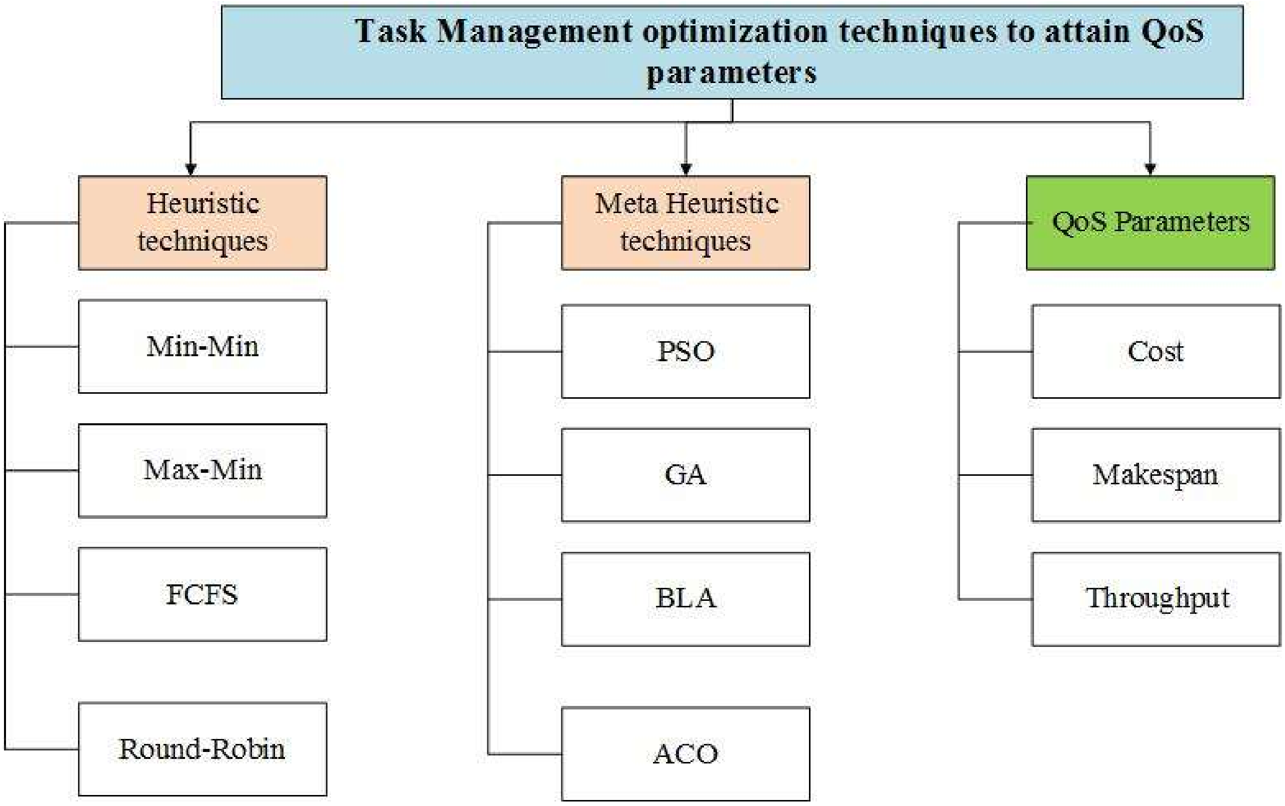

Fig. 2: Taxonomy of Task management

a) Heuristic technique: This technique is meant to be problem dependent. Heuristic algorithms may be used only for specified problems. Heuristic algorithms are more focused on speed rather than accuracy in the system. In the heuristic technique, one can guess a solution to a problem but he does not know how close this is to a solution to any problem [24]. Traditionally heuristic algorithms are used in many applications. Selecting the random number to pivot in quicksort is one example of the application of the heuristic algorithm. In Figure, 2 authors have mentioned some of the traditional heuristic algorithms.

i. Min-Min Technique**:** Min-Min is a type of heuristic technique; in a Fog environment this technique is used by many researchers for task management. The problem of resource management is still an issue in the Fog system. Many authors allocated VM to task as per policies of the Min-Min algorithm those tasks have minimum execution time, the machine will be allocated to that task first. The execution time of a task is computed based on the number of instructions in a task. As an application Min-Min algorithm is used in Smart-cities during power consumption in cities.

ii. Max-Min Technique**:** This technique is providing priority to larger tasks over smaller tasks. In the Fog system, those tasks that have a large number of instructions will get the priority first and those tasks that have a smaller number of instructions will get the lowest priority. The computation of Makespan in a system is examined by the implementation of the longest tasks.

iii. FCFS Approach**:** FCFS works on the process of FIFO queue. Jobs that get the resources request first will get the allocation of resources first. In the Fog environment, optimal QoS can be achieved through the FCFS heuristic technique by managing the request queue of tasks. A drawback of the technique is if some longest tasks get in the request queue first then some small tasks have to wait for a long time until the longest tasks get finished.

iv. Round-Robin technique: Round-Robin technique is a pre-emptive type FCFS approach.in Round-Robin technique CPU is allocated for quantum or for a particular time to tasks. If tasks fail to be completed in a given time then tasks need to wait in queue for the next turn. in the IoT-Fog-Cloud system, many researchers successfully get optimal cost and makespan by computing the expected completion time of the tasks.

b) Metaheuristic Techniques: This technique is meant to be problem independent. Metaheuristic techniques are not for specified problems. Using Meta-heuristic techniques researchers are solving many NP-hard problems. Unlike the heuristic technique, the meta-heuristic technique provides some optimal solutions to the problem in the next steps. An example of Hill-Climb is one of the finest applications of metaheuristic technique, where during a hill climb as moves to the next steps, there is the assurance of reaching the target. Authors have provided various algorithms for these techniques like PSO, ACO, GA, BLA, etc.

i. PSO optimization algorithm: The particle swarm optimization technique is a type of meta-heuristic technique. It's based on the nature of swarm or particle agents that moves around in search space for solving the problems to obtain an optimal solution. The motion of a particle is advised through the particle's own best position and other's global best solution. Many applications like healthcare, agriculture, smart industries, etc. use the PSO technique for solutions in their field.

ii. GA optimization algorithm: The technique is based on the process of natural election. In this technique, there is a selection of individuals for reproduction to generate offspring for an upcoming generation. Genetic algorithm is working on techniques of crossover and mutation. Application of GA algorithms like DNA analysis, economics, aircraft design, etc.

iii. Bees life optimization algorithm: Bees life algorithm is inspired by bees life when they are searching for their food and find the best food out of search space. Multiple bees looking at multiple places for their daily food. Out of multiple choices best, one is selected an optimal solution from the search space. Similarly, the concept is used in a fog environment, out of given resources which best available resource is allocated to the task for execution.

iv. Ant Colony optimization algorithm: it's a part of the optimization algorithm that depends upon the behavior of the ant colony. This optimization technique is used in finding an optimal path in a given solution search space. Ants pull out a pheromone to find a path for food. Ants roam randomly in search space, and when they find food on the path. Ants use that path as a source path for food with the help of pheromone. The same technique is used in the fog environment on smart city applications where the task is allocated to the Fog nodes in a distributed manner.

C. QoS parameters**:** In a Fog environment, Fog devices are heterogeneous in terms of memory, CPU, and other resources. Fog nodes in the Fog environment can be considered also mobile nodes. Nodes are distributed in an environment for efficient computation. But heterogeneity, mobility, and distributed structure make it difficult to achieve QoS parameters in the Fog environment [25,26]. Some of the QoS parameters are mentioned in **Figure 2** like cost, makespan, time, response time, etc. Efficient calculation of these parameters eventually depends on how resources are distributed to tasks [27,28]. Discussion on QoS parameters cost, makespan, time, and response time are done below:

I. Cost**:** The cost parameter is computed based on the computation done in the Fog system by machines. Cost is included based on memory cost, bandwidth cost, and processing cost or CPU utilization cost. Cost of computation of Task ($Ʈ_k^i$) may be considered using equation 1:

$$cost\,(Ʈ_k^i) = cp(Ʈ_k^i) + cm(Ʈ_k^i) + cb(Ʈ_k^i) \quad (1)$$

computation cost is computed by the sum of processing cost, memory cost, and bandwidth cost. processing cost using equation 3, is a cost of CPU usage cost ($\boldsymbol{\zeta 1}$) of each node n_i, and execution time is defined using equation 2:

$$Execution\text{-}time = \frac{Length-of-task}{cpu-rate} \quad (2)$$

Execution time in a system is computed by the number of instructions computed by the CPU based on fixed CPU frequency.

$$cp(Ʈ_k^i) = \zeta 1 * Execution\text{-}time \quad (3)$$

Cost of memory usage may be computed by the amount of memory required by tasks ($Mem Ʈ_k^i$) for computation with memory usage cost ($\boldsymbol{\zeta 2}$). Memory usage cost may be computed as using equation 4 below:

$$cm(Ʈ_k^i) = \zeta 2 * Mem(Ʈ_k^i) \quad (4)$$

Task requires bandwidth for computation in a Fog environment. The amount of bandwidth requires depends upon the size of input and output files. The cost of bandwidth usage may be computed as the bandwidth required by the tasks $band(Ʈ_k^i)$ at each node n_i with bandwidth usage cost parameter ($\boldsymbol{\zeta 3}$)**.** Bandwidth usage cost may be computed as using equation 5 below:

$$cb(Ʈ_k^i) = \zeta 3 * band(Ʈ_k^i) \quad (5)$$

II. Makespan: Makespan is a time between when tasks arrive for computation to all tasks completed. By improving the execution time of tasks, authors can improve the makespan of the system. Makespan may be calculated using equation 6:

Makespan= min {Execution-time} (6)

Execution time in a system may be calculated as mentioned in equation 2. Minimum makespan may also be considered as each task execute at the same time using equation 7.

$$Makespan_{Min} = Execution\text{-}time(N_1) = \text{------} = Execution\text{-}time\,(N_m) \quad (7)$$

III. Throughput (ϒ): Throughput is meant to be how the system is efficiently performing for a given bandwidth. Efficiency will be calculated by useful time (U_{time}) over the Total time (T_{time}) of the system. Useful time (U_{time})will be the time the system performs without any delay. Efficiency of system may be computed using equation 8.

$$\varepsilon = \frac{Utime}{Ttime} \quad (8)$$

Then throughput is defined as system efficiency **ε** over a provided bandwidth **β** using equation 9.

$$\text{Throughput } (\Upsilon) = \varepsilon * \beta \quad (9)$$

IV. SIMULATION PARAMETER AND SETTING

This section elaborates on the simulation strategies and parameter settings used in this work. Simulation has been performed on the Fog-workflow sim toolkit [29] with a CPU Core i3-2370M @ 2.40GHz, 8GB RAM, and operating system windows 7. For the whole scenario to work, each server and node has its own set of memory and processing capacity in MIPS. For simulation, researchers set the parameters, Number of Fog nodes=10 nodes, Number of cloud servers= 3, processing capacity of Fog nodes= 1300 MIPS, and processing capacity of Cloud Nodes = 1600 MIPS.
in this work, Parameter setting for PSO technique are Number of particles=20, iteration=100, c1=2.05, c2=20.3, and inertia weight=0.5. For GA technique population size=50, No. of iterations=100, cross rate=0.8, and mutation=0.1.

V. RESULT AND DISCUSSION

This section describes the comparison of parameter makespan, cost, and energy consumption between Min-Min, Max-Min, FCFS, RoundRobin, PSO, and GA.

As shown in Fig. 3, a comparison of parameter makespan is done between all heuristic and meta-heuristic techniques for 60-300 tasks. All comparison is done on 5 types of montage data sets. Heuristic technique FCSF outperforms all other Min-Min, Max-Min, and Round-Robin techniques. Improvement of 3-6% is shown with the FCFS technique in comparison to all heuristic techniques. For a higher number of tasks 200 tasks and 300 tasks, the FCFS technique performs much better in comparison to a lower number of tasks. Besides, for meta-heuristic techniques PSO and GA. PSO technique is performing better concerning GA. An improvement of 1.5%-2.5% is noticed with PSO in comparison to GA. The Metaheuristic technique is performing better for the lower number of tasks. For a higher number of tasks, less improvement is seen with PSO on GA.

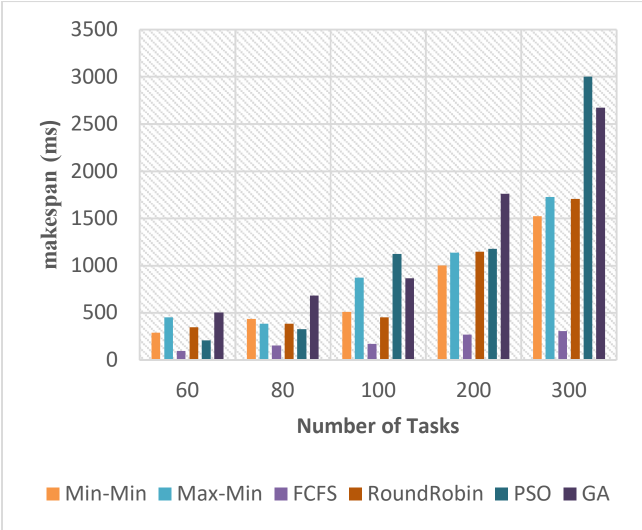

Fig. 3: Makespan Comparison between techniques

A comparison of cost parameters is done for all techniques as shown in Fig. 4. Total cost is depending on processing cost, bandwidth cost, and memory cost as shown in equation 1. Total cost in heuristic technique is less for FCFS technique w.r.t other techniques. Cost is computed in Grid $ in exchange for currency unit. Cost is less up to 1.5%-2% for FCFS technique in comparison with Min-Min, Max-Min, and Round-Robin. Whereas in the Meta-heuristic technique, GA outperforms PSO by about 1.5%-2%. For a high number of tasks such as 200 and 300, less cost is seen for GA on PSO. A comparison of energy consumed by heuristic and meta-heuristic techniques is shown in Fig. 5. The Min-Min technique consumes much less energy in contrast to Max-Min, FCFS, and Round-Robin techniques. Min-Min consumes approx. 20-22% less energy in comparison to all techniques for heuristic strategy. Besides for metaheuristic technique, the GA technique is consuming less energy in a Fog environment in contrast to the PSO technique. In the IoT-Fog-Cloud environment, GA consumes approx. 6-7% less energy in contrast to the PSO technique

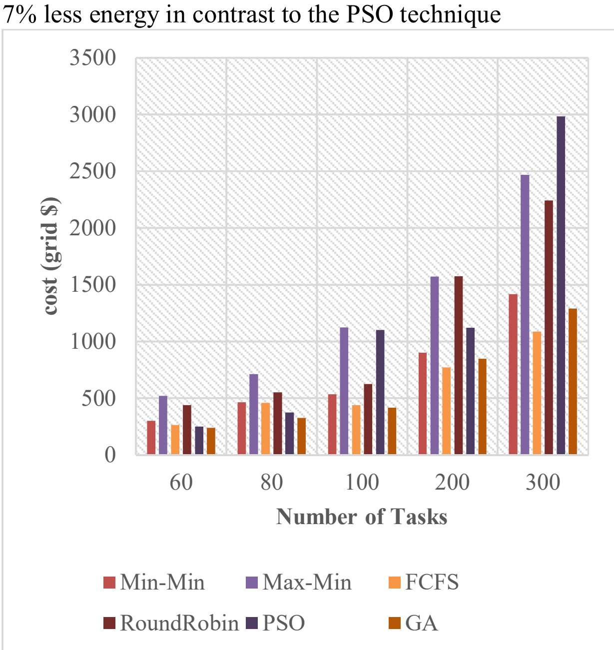

Fig. 4: Cost Comparison between techniques

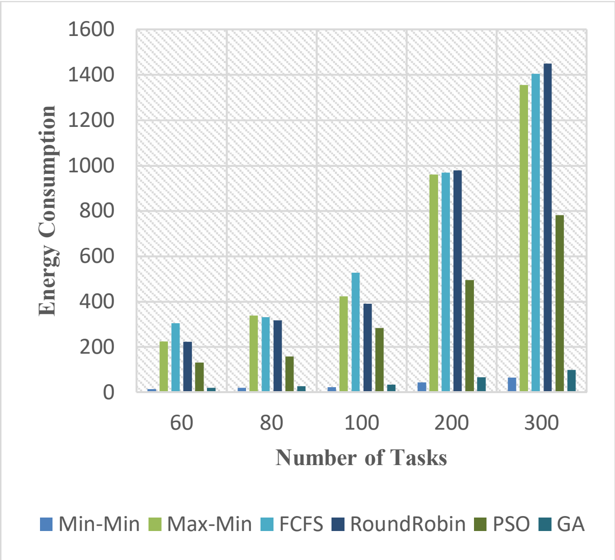

Fig. 5: Energy consumed Comparison between techniques

VI. CONCLUSION

This work focuses on, the discussion on some of task management and resource scheduling techniques. Their various QoS parameters and their impact is also discussed in Fog environment. Montage workload for tasks is considered to simulate the Fog environment. Various heuristics and meta-heuristic techniques are implemented on workflow sim to show the effectiveness of state-of-the-art techniques. These techniques are evaluated on makespan, cost and energy consumed. The experimental setup shows that the techniques, like PSO and GA, has shown less significant improvements with larger number of tasks for makespan. For cost computation of all execution of tasks on Fog nodes is optimal in case of GA based heuristics techniques *w.r.t* Min-Max, Max-Min, FCFS, Round robin and PSO. Then for energy consumption, PSO and GA techniques has out-performed the other techniques with lesser to larger number of tasks. Thus, heuristics and meta heuristics are significant in task management in Fog environment to utilize the limited resources. Thus, in future more task management techniques are required for optimizing the utilization of limited resources so that computing environment can support the real-life network traffic requests.

DECLARATION OF INTERESTS

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] Elgendy, I. A., Zhang, W. Z., He, H., Gupta, B. B., El-Latif, A., & Ahmed, A. (2021). Joint computation offloading and task caching for multi-user and multi-task MEC systems: reinforcement learning-based algorithms. Wireless Networks, 27(3), 2023-2038.

[2] Kumar, Sumit, and Rajeev Tiwari. "An efficient content placement scheme based on normalized node degree in content centric networking." Cluster Computing 24.2 (2021): 1277-1291.

[3] Tiwari, Rajeev, and Neeraj Kumar. "A novel hybrid approach for web caching." 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. IEEE, 2012.

[4] Tiwari, Rajeev, et al. "Automated parking system-cloud and IoT based technique." International Journal of Engineering and Advanced Technology (IJEAT) 8.4C (2019): 116-123.

[5] Wang, S., Zhao, T., & Pang, S. (2020). Task scheduling algorithm based on improved firework algorithm in fog computing. IEEE Access, 8, 32385-32394. DOI: 10.1109/ACCESS.2020.2973758

[6] Naha, R. K., Garg, S., Chan, A., & Battula, S. K. (2020). Deadline-based dynamic resource allocation and provisioning algorithms in fog-cloud environment.,Future Generation Computer Systems, 104 , 131-141.

[7] Khan, E., Garg, D., Tiwari, R., & Upadhyay, S. (2018, February). Automated toll tax collection system using cloud database. In 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU) (pp. 1-5). IEEE.

[8] Tiwari, R., & Kumar, N. (2012, December). Dynamic Web caching: For robustness, low latency & disconnection handling. In 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing (pp. 909-914). IEEE.

[9] Tiwari, R., Kumar, K., & Khan, G. (2010, November). Load balancing in distributed web caching: a novel clustering approach. In AIP Conference Proceedings (Vol. 1324, No. 1, pp. 341-345). American Institute of Physics.

[10] 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.

[11] Rehman, S., Javaid, N., Rasheed, S., Hassan, K., Zafar, F., & Naeem, M. (2018, October). Min-min scheduling algorithm for efficient resource distribution using cloud and fog in smart buildings. In International Conference on Broadband and Wireless Computing, Communication and Applications (pp. 15-27). Springer, Cham.

[12] Jamil B, ShojafarM, Ahmed I, Ullah A, Munir K, Ijaz H. A job scheduling algorithm for delay and performance optimization in fog computing. Concurrency Computat Pract Exper. 2019;e5581. https://doi.org/10.1002/cpe.5581.

[13] Ghobaei-Arani M, Souri A, Safara F, Norouzi M. An efficient task scheduling approach using moth-flame optimization algorithm for cyber-physical system applications in fog computing. Trans Emerging Tel Tech. 2019;e3770. DOI: 10.1002/ett.3770.

[14] Canali, C., & Lancellotti, R. (2019). GASP: genetic algorithms for service placement in fog computing systems. Algorithms, 12(10), 201.

[15] Potu N, Jatoth C, Parvataneni P. Optimizing resource scheduling based on extended particle swarm optimization in fog computing environments. Concurrency Computat Pract Exper. 2021;e6163. DOI: 10.1002/cpe.6163.

[16] Kishor, A., & Chakarbarty, C. (2021). Task offloading in fog computing for using smart ant colony optimization. Wireless Personal Communications, 1-22.

[17] Talaat, F. M. (2022). Effective prediction and resource allocation method (EPRAM) in fog computing environment for smart healthcare system. Multimedia Tools and Applications, 81(6), 8235-8258.

[18] Godinho, N., Silva, H., Curado, M., & Paquete, L. (2022). A reconfigurable resource management framework for fog environments. Future Generation Computer Systems.

[19] Bisht, J., & Vampugani, V. S. (2022). Load and Cost-Aware Min-Min Workflow Scheduling Algorithm for Heterogeneous Resources in Fog, Cloud, and Edge Scenarios. International Journal of Cloud Applications and Computing (IJCAC), 12(1), 1-20.

[20] Singh, S. P. (2022). Effective Load Balancing Strategy Using Fuzzy Golden Eagle Optimization in Fog Computing Environment. Sustainable Computing: Informatics and Systems, 100766.

[21] Tiwari, R. (2010). Gulista khan,"Load Balancing through distributed Web Caching with clusters". Proceeding of the CSNA, 46-54.

[22] Kumar, S., & Tiwari, R. (2021). Dynamic popularity window and distance-based efficient caching for fast content delivery applications in CCN. Engineering Science and Technology, an International Journal, 24(3), 829-837.

[23] Chithaluru, P., Tiwari, R., & Kumar, K. (2021). Performance analysis of energy efficient opportunistic routing protocols in wireless sensor network. International Journal of Sensors Wireless Communications and Control, 11(1), 24-41.
[24] Sangeetha, A., & Rajendran, T. (2022). Supervised Vector Machine Learning with Brown Boost Energy Efficient Data Delivery In MANET. Sustainable Computing: Informatics and Systems, 100761.
[25] Chithaluru, P., Tiwari, R., & Kumar, K. (2021). Arior: Adaptive ranking based improved opportunistic routing in wireless sensor networks. Wireless Personal Communications, 116(1), 153-176.
[26] Tiwari, R., Sille, R., Salankar, N., & Singh, P. (2022). Utilization and Energy Consumption Optimization for Cloud Computing Environment. In Cyber Security and Digital Forensics (pp. 609-619). Springer, Singapore.
[27] Tiwari, R., Mittal, M., Garg, S., & Kumar, S. (2022). Energy-Aware Resource Scheduling in FoG Environment for IoT-Based Applications. In Energy Conservation Solutions for Fog-Edge Computing Paradigms (pp. 1-19). Springer, Singapore.
[28] Varshney, S., & Singh, S. (2018). A survey on resource scheduling algorithms in cloud computing.,International Journal of Applied Engineering Research, 13(9), 6839-6845.
[29] Liu, Xiao, et al. "FogWorkflowSim: an automated simulation toolkit for workflow performance evaluation in fog computing." 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2019.

Gaurav Goel was born on January 09, 1987, in India. He has done his B.Tech from Haryana Engineering College Jagadhri, India in the stream Computer Science and Engineering from 2004-to 2008. Further has completed his MTech in Computer Science and Engineering in 2014. Currently, the Author pursuing his Ph.D. in Computer Science and Engineering from UPES Dehradun, India in the area of Fog Computing. He has held the position of Assistant professor in Chandigarh Group of Colleges, Landran, India. He has a total Experience of 14 years.

Rajeev Tiwari is working as a Professor in SCS in UPES, Dehradun (India). He is a senior IEEE Member. He has done Ph.D. in CSE from Thapar Univeristy, Patiala (Punjab). He has more than 16 years of research and teaching experience. His broad area of research is Cloud Computing, MANET, VANET, QoS in wireless networks, Cache Invalidation Techniques, Internet of Things (IoT), Big Data Analytics and Machine Learning. He is also an expert in simulation and design of scenarios on NS2, SUMO and MOVE. He is an active member of ACM, IEI, IACSIT and IAENG. He has 40 + international publications in SCI, and Scopus indexed journals and conferences. His publications are majorly indexed in SCI, Scopus and DBLP. He has published two Patents also. He has enriched experience of organizing international conferences successfully. He has major roles as Organizing Chair, TPC chair, Co-Chair Secretory etc in NGCT 2016, NGCT 2017, NGCT 2018, NGCT 2019, ICACCA 2019, IC4S 2019 etc with Springer and IEEE. He has chaired many sessions in international conferences. He is a lead reviewer in SCI indexed journal of Springer, IEEE and ScienceDirect and international conferences. He has edited books and published book chapters also with international publishers. He has contributions in funded projects in field of IoT, Cloud and wireless network QoS.