The Resource Allocation Using Weighted Greedy Knapsack Based Algorithm in an Educational Fog Computing Environment

https://doi.org/10.3991/ijet.v17i18.32363

Shruthi G.[1(✉)], Monica R. Mundada[2], Supreeth S.[3]
[1]Research Scholar, Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, VTU, Bengaluru and School of CSE, REVA University, Bengaluru, India
[2]Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bengaluru, India
[3]School of CSE, REVA University, Bengaluru, India
shruthi.g@reva.edu.in

Abstract—The Internet of Things ecosystem pertains to the web-enabled connected devices that operate built-in processors to record, send, and act on information from their surroundings via embedded communication hardware. IoT applications span from education, healthcare to self-driving cars. The high delay supplied through the connecting network to the data centers and huge data traffic can cause the system to become congested. Hence, the cloud is not suggested for the delay-sensitive applications and it is extremely difficult to provide educational applications, particularly in a mix of cloud and fog conditions. Fog computing was created to address this problem and improve time-sensitive applications by considering quality of service (QoS). The allocation of resources and scheduling of tasks are challenging issues for IoT applications in a fog environment. The resources are required for each educational application that includes several modules to run. In this paper, we used Weighted Greedy Knapsack (WGK) based algorithm for the resource allocation to the modules/components in the fog system. We have considered the smart parade application to provide certain services/resources and the proposed method was experimented in iFogSim. The proposed method shows a better energy consumption and execution cost than that of the concurrent, First-Come-First-Served (FCFS) and Delay-Priority algorithms.

Keywords—fog computing, cloud computing, resource allocation, time-sensitive applications

1 Introduction

In the present era, there is a rising number of things and gadgets that are linked to the Internet to gather and disseminate features derived from the physical medium which create a concept called the Internet of Things (IoT) [1]. In accordance with the International Data Corporation (IDC), by 2025, the amount of linked IoT gadgets will have surpassed 41 billion, creating more than 79 ZB of data [2]. IoT storage and computational requirements can be handled depending on Cloud [3] as an operation based on information and communication technology-based concepts. For all the applications based on IoT, cloud computing [4] is not suitable despite its widespread use. The cloud computing is not recommended to be used for applications that require low latency, such as healthcare, smart homes, and smart transportation. This is owing to the significant delay introduced by network connections to data centers, as well as excessive data flow that may cause the network to become congested. Fog computing has emerged as a new approach for performing latency-sensitive applications in response to this problem. Fog computing is a systemic-level horizontal structure that spreads processing, storage, control, services, and resources in networking anywhere from the cloud as shown in the Figure 1.

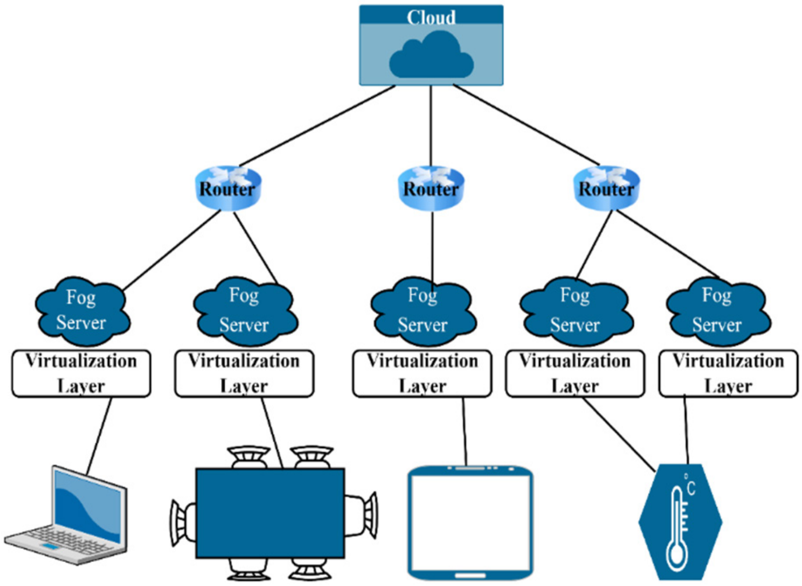

Fig. 1. Architecture for fog computing

Fog is a geo-distributed layer [5] of servers having computation, memory, and capabilities of the network that act as a bridge between the IoT and cloud levels. Fog servers are closer in proximity to IoT devices than cloud servers, as a result, there is a reduction in response time, latency and the ability to support the most latency-sensitive IoT applications [6]. Fog devices are widely dispersed and have limited resources. In a Fog Environment, to run the IoT applications the most difficult challenge is the resource allocation problem. To handle resource allocation problems efficiently, Fog computing needs to take an account of the requested Service Quality (QoS).

The recent research has conducted a detailed survey of fog computing which is focused on allocation of resources and problems in scheduling [13] [23–25]. The resource allocation plays a major issue by considering the different metrics used in fog and cloud computing. We inspected the existing technology to resolve the allocation of resource problems in the Fog environment. The survey has given insight into the current research work on resource allocation. To deal with the resource allocation in our work, we have considered that the energy consumption and execution cost has a metrics.

In our approach, we have considered the smart parade application. The concept is to obtain people's information involved in a parade and to provide certain services/ resources. We have experimented in an iFogSim [20] library using the WGK algorithm for resource allocation to the modules in the fog network and analyzed the outcome with another methods such as Concurrent, First-Come-First-Served (FCFS) and Delay-Priority algorithms [22]. We run the simulation by 6 conditions of zones, cameras/ mobile devices, and Fog devices.

Using smart parade case study, weighted greedy knapsack algorithm was used to replicate our allocation technique in the fog and cloud system. Such case study has been based on the gathering of data relating to object and people tracking, data transfer to the edge node, processing, and cloud storage. The following are the major contributions we made in this study.

We used a weighted greedy knapsack based strategy to formulate the allocation of resource issue in Fog Computing. With multi objective approach, weighted sum method was employed to generate the objective function. The suggested algorithm is processed quickly using this way. After the module installation in Fog Devices, the best Physical Resources of the Fog Device are assigned to the desired modules, according to Weighted greedy knapsack (WGK) based Allocation.

The amount of executed modules in Fog Devices, transmission duration, and management of resource interval were investigated in Smart Parade Application.

The proposed approach is analyzed with existing algorithms based on various configurations such as zones, cameras/mobile devices, and Fog Device. For the smart parade application, simulation findings reveal that WGK outperforms Concurrent, First-Come-First-Served (FCFS), and Delay-Priority algorithms [22] in terms of consumption of energy and cost of total execution.

The remaining parts of the paper is ordered as follows: The literature review in section 2 summarizes the current resource allocation method. In section 3, we present a detailed methodology based on the smart parade applications. Section 4 exposes the results of developed approach by considering the energy consumption and total execution cost as a metric that is compared with the other existing techniques. The conclusion of this study is presented in Section 5.

2 Literature review

An Internet of Everything concept is increasingly being used in the development of new applications for many fields such as smart agriculture, smart cities, and big data streaming, and so on. For execution, these IoE apps make use of cloud computing resources. A Fog computing that allows flexibility, heterogeneity, geographic spread, context awareness, and applications including storage, computing, analytics, and networking on nearby fog nodes, is an extension of cloud computing. Resource Allocation is a significant difficulty in the fog environment, which is resource constrained, diverse, dynamic, and unpredictable. This section explains the Resource Allocation algorithms and highlights the articles' outcomes.

With the demands of latency-sensitive applications, Fog computing has become a building block of 5G networks [7]. In fog Radio Access Network (F-RAN) there are intermediate network devices that are mainly equipped with virtualized computing resources. These resources are inefficient and limited that could obstruct F-RAN nodes. To solve this problem a scheme called autonomous and dynamic resource allocation for F-RAN is considered. This uses an algorithm called Reinforcement Learning that calculates the minimum requirement of resources for each Fog node by optimizing latency, cost, and energy consumption in F-RAN. This model is evaluated and compared using a simulator.

The current contributions concerning resource allocation and computation offloading [8] are inefficient with the growing requirement for reduced services with high throughput and low latency is very challenging in F-RANs. To address this, Deep Reinforcement Learning (DRL) has been developed which is based on joint resource allocation and computation offloading schemes that attain low latency in F-RANs. In the DRL technique, the controller allocates the job to be processed at the system level locally or to offload the job to a cloud server or fog access point and allocates resources based on the serving level. The use of this approach is to decrease the delay and improve the system's throughput and it is simulated.

An auction is a popular technique for allocating resources and pricing in the fog and cloud computing scenario [9]. The difficult problem from the past two decades in Fog computing is the pricing mechanism and resource allocation. The Fog users are facing a problem related to resource allocation and administration in collecting the permanent computing resources in a given time frame. To solve this problem two algorithms are introduced, Fixed-priced fog node allocation and combinatorial auction-based fog service allocation mechanisms. In these two algorithms, the combinatorial auction-based mechanism has resulted in a high resource allocation proficiency. This mechanism is applied for various applications and validated with the time complexity.

The increased number of IoT devices and the production of a large amount of information using only cloud or fog could not satisfy the requirements of the users. For the delay-sensitive applications, both fog and cloud play a major role. In [10] it addressed an algorithm which is a task distribution AI-based algorithm between cloud and fog servers. This algorithm aims to reduce the traffic on the internet and the response time by issuing tasks between the cloud and fog servers. The approach is more understandable as the number of tasks in the broker grows. The simulation is performed in MATLAB.

The content delivery networks [11] release a large data center's carbon footprint which is stationed in a distributed manner. A new paradigm has been introduced by Cisco to provide low-latency access called fog computing. In this new paradigm, for the video streaming service, there is a problem of minimizing the carbon footprint and resource allocation. To address this problem a distributed technique called Alternating direction method of multipliers (ADMM) has been used. This distributed technique is using a proximal algorithm to break down complex problems into many sub-problems so that they can be resolved speedily. To evaluate distributed technique performance, numerical results were conducted.

Fog computing can extend the resources from cloud to the edge networks for solving the issues such as (1) Intelligent devices might be inefficient in their memory, battery, storage, processing, resource allocation, and network resources, (2) A centralized cloud server may not be suitable for time-critical services, applications, and resource allocation requests. To solve these issues a Heuristic-based technique [12] called decision rules which is of a straightforward decision tree is considered on three criteria i.e., completion time, service size, Capacity of VMs for managing user requests, and balancing workload. Even in fog and cloud computing, the method is used to assign resources to satisfy service level agreements (SLA) and quality of service (QoS), and to distribute massive data more efficiently. The simulation results show that performance is superior when analyzed with other algorithms in balancing workload efficiently, improving the resource allocation, optimizing the distribution of big data.

Gupta et al. come up with iFogSim in [20], a Fog and IoT wide scale simulator. You can form infrastructures using millions of IoT artifacts or Fog nodes (and data centers). It allows users to identify and deploy their resource management strategies for allocating and scheduling IoT services across the entire facility, as well as to monitor the effect on connection latency, network congestion, energy, and costs. However, this simulator also does not provide data placement management. Ni, Zhang, et al. proposed Priced Timed Petri Nets [14] by considering the task as a parameter but not as a time factor. Zhang et al. proposed the game theory [15] by considering utility of the fog node but not considered time and cost factor [13]. Zhang et al. proposed the game theory [16] by considering energy consumption but not considered time. Nguyen et al. considered utility of fog node [17] but failed with time factor. They considered energy consumption [18] and latency but failed with the time factor.

In fog and cloud computing, resource allocation plays a major issue by considering the different metrics. We have reviewed the existing techniques to resolve an allocation of resource problem in the Fog environment. The literature survey provided insights into the current research work on resource allocation by considering any one of the metric. But it is restricted to the problem dealing with resource allocation by considering one (or) two metrics. In our work, we have considered energy consumption and execution cost as a metrics.

3 Methodology

3.1 Preliminaries

Fog environment architecture is represented in Figure 1 which consists of sensor nodes, edge devices, fog devices and cloud. The sensor nodes are placed at base. The gateways redirect data to higher level from these nodes. The actuators control environment at base of the architecture. The fog network performs data processing in Fog Devices, sending and sensing data, dividing applications into different components/modules, and assigning resources for execution. Each Fog Device includes Servers as $\{\text{Server}_1, \text{Server}_2\ldots\ \text{Server}_N\}$. The applications used to gather and retain data in data center for future processing and analysis. The properties of server's include bandwidth, storage, RAM, and processing elements (PEs). The server should always satisfy condition as specified in the Eqn. 1.

$$FB_{Lower} \leq \sum_{i=1}^{M} SB_i \leq FB_{Upper} \tag{1}$$

Where; FB:-Fog's Bandwidth; SB:-Server's Bandwidth; FB_{Lower}:-Lower Bandwidth and FB_{Upper}:-Upper Bandwidth of each Fog Device; M:-Number of servers.

The total bandwidth used by all servers within each Fog Device is between FB_{Lower} and FB_{Upper}.

PEs of servers are assigned to application components in Fog Device and they are executed. MIPS is still the most significant element of a PE. In all FDs, these values are set at the start of simulation. The Total Allotted MIPS (TAM) of all PEs is updated once a PE is assigned to an applications as shown in the Eqn. 2.

$$TAM = \sum_{i=1}^{M}\sum_{j=1}^{N} PEM_{ij} \tag{2}$$

Where; PEM:-PE's MIPS; M:-Number of servers; N:-Number of PEs in a Server; PEM_{ij}:-MIPS of jth PE in ith server.

A virtual machine (VM) is a sort of system which includes the properties like MIPS, Memory, and bandwidth are all attributes of the module. The Resource Allocation issue is NP-hard and takes a long time to solve. We offer a quick approach for allocating PEs to software modules in this work. The default resource allocation distributes server resources in FDs evenly between all running application modules and there is K number of PEs in FD as shown in the Eqn. 3.

$$\{PE_1, PE_2, PE_3, PE_4 \ldots PE_k\} \tag{3}$$

PEs are allocated to modules as part of the knapsack issue. As with the knapsack issue, every object is assigned a weight and profit. So, the Weights and Profits are as defined in the Eqn. 4 and 5.

$$\{W_1, W_2, W_3 \ldots W_n\} \tag{4}$$

$$\{P_1, P_2, P_3 \ldots P_n\} \tag{5}$$

We have considered two main objective i.e., total utilization of CPU and the bandwidth. An object is arranged in knapsack, if it satisfies two circumstances as shown in the Eqn. 6 and Eqn. 7.

$$\sum w_{1i} * \delta_i \leq FDe_{MIPS} \tag{6}$$

$$\sum w_{2i} * \delta_i \leq FD_{Bandwidth} \tag{7}$$

Where; where w_{1i}:-MIPS of $module_i$ and w_{2i}:-bandwidth of $module_i$, $\delta_i = 0, 1$.

The knapsack capacity is measured by the MIPS and bandwidth. Total modules MIPS is less than FD's MIPS and total modules bandwidth is less than FD's bandwidth. Purpose of the Weighted Greedy knapsack to keep the weight low and maximum profit i.e., Maximize ($\Sigma P_i * \delta_i$).

3.2 Scenario for a smart parade application

We have considered a Scenario for a Smart Parade Application as shown in Figure 2 [19] and this IoT application consists of several elements. The concept is to obtain people's information involved in a parade and to provide certain services/ resources. The application collects the parade footage and analyses them to detect such trends and/or risks to security, such as people or environment anomalies. The scenario for a Smart Parade was simulated by the technique called WGK (Weighted greedy knapsack) based algorithm in the network of fog. In the Record Parade Footage Part, recognizable features are extracted from parade footage and they are sent for processing and analysis and then the data will be uploaded on the cloud. This technique will be analyzed with the other techniques based on a variety of configurations. The simulation results show the best result for the energy consumption and total execution cost.

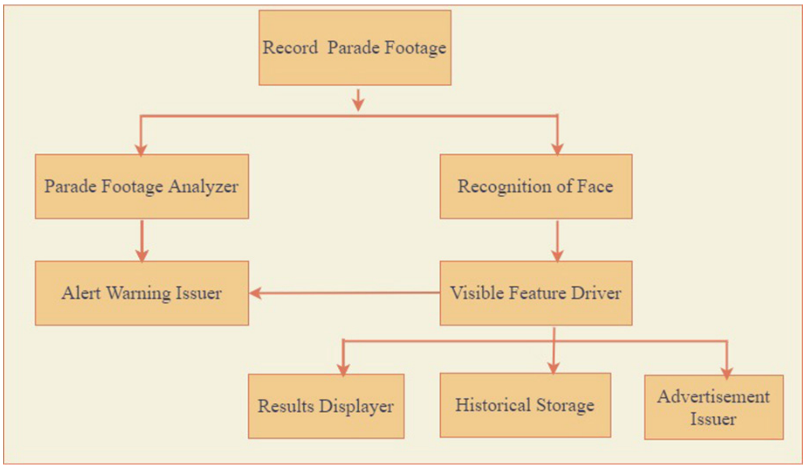

Fig. 2. The smart parade application [19]

Algorithm: Weighted Greedy knapsack (WGK) based allocation algorithm

```
Establish Fog Broker
Build an Application (smart parade)
        Smart Parade → Analyzer for parade footage → Recognition of Face →
        Visible Feature Driver → Alert Warning Issuer → Results Displayer →
        Historical Storage → Advertisement Issuer
for i=1 to Zone_max
    for j=1 to Camera_max
        Build Fog Device
    end for
end for
for i=1 to Fog Device_max
    Add Module to Fog Device_i
end for
Submit Application
Start iFogSim
pArray = Calculate earnings and module weights based on CPU use and network bandwidth
sort pArray P1/W1, P2/W2 …
i = 1;
sum_MIPS = sum_BW = 0
while sum_MIPS<= FD_MIPS and sum_BW<=FD_BW
    Choose module_i(C_i) in array
    if the module is not executing in the current FD,
            Allocate PE to module_i(C_i)
            sum_MIPS = sum_MIPS + MIPS_modulei
            sum_BW = sum_BW + BW_modulei
    end if
    i=i+1
end while
Update energy and price using Eqs, (8) and (9) respectively
Stop iFogSim
```

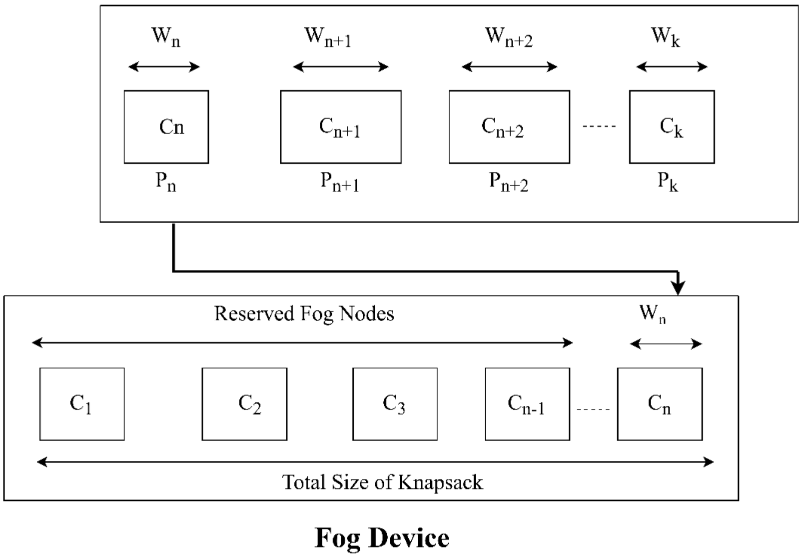

Fig. 3. Weighted greedy knapsack

Modules are entered into the system as it is shown in Figure 3. C_i is the ith object (or) module that will be in the knapsack. Fog Nodes include processing elements, these elements are assigned to the modules according to the algorithm. When a module cannot fit into the knapsack's array, it is moved to another Fog Device in iFogSim [21]. The Algorithm shows pseudo-code for our proposed method.

4 Results and discussions

The preliminary set up for the simulation are shown in the following tables. Table 1 explicates Fog Device (FD) outline, where each FDs acts as micro datacenters (MDCs) to provide the resources for the application modules. Each FD as an MDC possesses several attributes, comprising RAM (KB), MIPS, UpBW (Upper bandwidth in kilobytes per second), DownBW (Down bandwidth in kilobytes per second), level in the hierarchical structure, Rate per MIPS, busy, and idle power (MW). Table 2 explains the setup of the server, detailing its bandwidth, storage, architecture, VM model, operating system, cost, time zone, cost per storage and cost per memory, among other attributes. Table 3 explains the setup of application modules, including MIPS, bandwidth, RAM and module size.

Table 1. Fog device outline [21]

Name	MIPS	RAM	UpBw	DownBw	Level	Rate Per MIPS	Busy Power	Idle Power
Cloud	44,800	40,000	100	10,000	0	0.01	1648	1.332
Proxy-server	2800	4000	10,000	10,000	1	0	107.339	83.4333
Zones	2800	4000	10,000	10,000	1	0	107.339	83.4333
Cameras/Mobile devices	500	1000	10,000	10,000	3	0	87.53	82.44

Table 2. Server outline [21]

Storage	BW	Architecture	OS	VM Model	Time Zone	Cost	Cost Per Memory	Cost Per Storage
1,000,000 B	10,000 B/S	X86	Linux	Xen	10	3	0.05	0.01

Table 3. Module outline [21]

RAM	MIPS	Size	BW
10 B	1000	10,000 B	1000 B/S

The proposed approach is experimented in iFogSim tool with java programming with configuration as RAM with 3 GB and Microsoft Windows 10 OS framework with 32-bit are designed to run a PC with CPU features, including Intel Core i5 2.67 Giga Hertz. We ran simulations of the WGK algorithm and compared the results to other methods like Concurrent, First-Come-First-Served (FCFS) and Delay-Priority algorithms. Conducted the experiment by 6 zones, cameras/mobile devices and Fog Devices in the following formats: {1 = (2, 6, 16), 2 = (2, 7, 18), 3 = (3, 6, 23), 4 = (3, 7, 26), 5 = (4, 6, 30), 6 = (4, 7, 34)}.

4.1 Consumption of energy

The Eqn. 8 is used to calculate energy consumption.

$$E = CEC + (PT - LUUT) * SP \tag{8}$$

Where: E:-Energy, CEC:-Current Energy Consumption, PT:-Present Time, LUUT:-Last Utilization Update Time and SP:-Server Power.

The experimented results depicts that proposed algorithm consumes energy less than the other methods. The mean value of the energy consumed by the WGK, Concurrent, FCFS and Delay-Priority algorithms in smart parade scenario is $1.30*10^7$, $1.57*10^7$, $1.68*10^7$ and $1.47*10^7$ respectively. WGK algorithm reduces consumption of energy compared to Concurrent, FCFS, and Delay-Priority methods. The comparison of WGK, Concurrent, FCFS and Delay-Priority methods is presented in Figure 4.

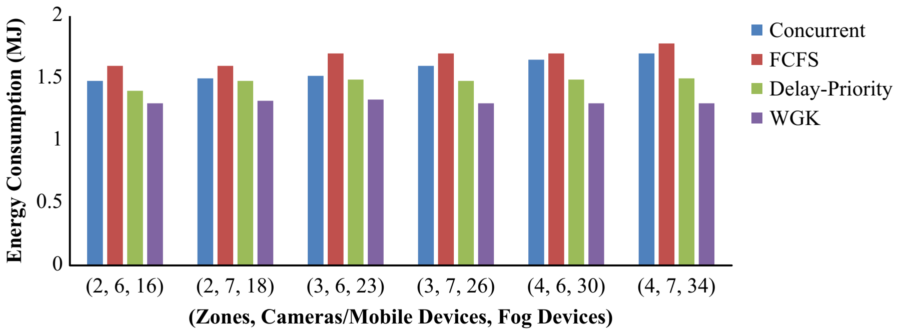

Fig. 4. Energy consumption of smart parade scenario

4.2 Total execution cost

$$\text{Cost} = \sum_{i=1}^{F} [EC + (CT - LUUT) * RPM * LU * TM] \quad (9)$$

The Total Execution Cost [21] is shown in Eqn. 9.

Where F:-Number of Fog devices, EC:-Cost for the Execution, CT:-Simulation Current Time (or) FogSim clock, LUUT:-Last Utilization Update Time, RPM:-Rate Per MIPS (Million Instruction per second), LU:-Last Utilization, TM:-Total MIPS of the server.

Last Utilization [21] is calculated as shown in Eqn. 10, where TMA: Total allocated MIPS of the server.

$$\text{LU=Min (1, TMA/TM)} \quad (10)$$

The mean value of the execution cost by the WGK, Concurrent, FCFS and Delay-Priority algorithms in smart parade scenario are equal to $1.58*10^6$, $2.55*10^6$, $3.03*10^6$, and $2.01*10^6$ respectively. The comparison of WGK, Concurrent, FCFS, and Delay-Priority methods is presented in Figure 5.

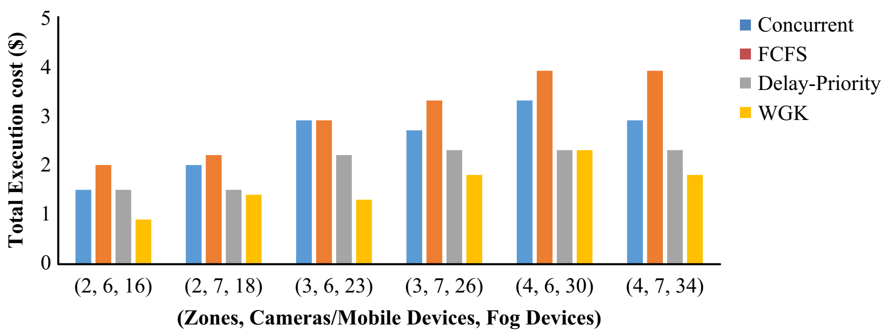

Fig. 5. Total execution cost of smart parade scenario

5 Conclusion

The IoT covers a wide variety of applications from healthcare to automated vehicles in different fields. In fog computing, effective and practical allocation of resources is a vital and difficult subject. This paper deliberates about WGK algorithm for the smart parade scenario to provide certain services/ resources. We have simulated/experimented using iFogSim by considering WGK and compared it with the Concurrent, FCFS, and Delay Priority algorithm for different configurations for the consumption of energy and cost of total execution as a metric. The mean value of the energy consumed by the WGK, Concurrent, FCFS, and Delay-Priority algorithms in smart parade scenario are $1.30*10^7$, $1.57*10^7$, $1.68*10^7$ and $1.47*10^7$ respectively so, the Energy consumed in WGK is less compared to other methods. The mean value of the execution cost by the WGK, Concurrent, FCFS, and Delay-Priority algorithms in smart parade scenario is equal to $1.58*10^6$, $2.55*10^6$, $3.03*10^6$, and $2.01*10^6$ respectively so, execution cost in WGK is less compared to the other methods.

6 References

[1] Lahmar, Imen Ben, and KhouloudBoukadi, "Resource allocation in fog computing: A systematic mapping study," In 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), pp. 86–93. IEEE, 2020. https://doi.org/10.1109/FMEC49853.2020.9144705
[2] "The growth in connected IoT devices is expected to generate 79.4ZB of data in 2025, according to a new IDC forecast," Int. Data Corp. (IDC), Framingham, MA, USA, Rep. prUS45213219, 2019. Accessed: Apr. 2020. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS45213219
[3] S. Supreeth, and K. K. Patil, "Virtual machine scheduling strategies in cloud computing: A review," Int. J. Emerg. Technol., vol. 10, no. 3, (2019): 181–188.
[4] S. Supreeth, and Biradar, "Scheduling Virtual Machines for Load balancing in Cloud Computing Platform," International Journal of Science and Research (IJSR), (2013).
[5] Martinez, Ismael, AbdelhakimSenhajiHafid, and AbdallahJarray, "Design, resource management, and evaluation of fog computing systems: A survey." IEEE Internet of Things Journal, vol. 8, no. 4, (2020): 2494–2516. https://doi.org/10.1109/JIOT.2020.3022699
[6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of things," in Proc. 1st ed. MCC Workshop Mobile Cloud Comput., (2012): 13–16. https://doi.org/10.1145/2342509.2342513
[7] N. Khumalo, O. Oyerinde and L. Mfupe, "Reinforcement Learning-based Computation Resource Allocation Scheme for 5G Fog-Radio Access Network," 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), (2020): 353–355, https://doi.org/10.1109/FMEC49853.2020.9144787
[8] G.M. Rahman, Shafiqur, Tian Dang, and Manzoor Ahmed, "Deep reinforcement learning-based computation offloading and resource allocation for low-latency fog radio access networks," Intelligent and Converged Networks, vol. 1, no. 3, (2020): 243–257. https://doi.org/10.23919/ICN.2020.0020

[9] Bandyopadhyay, Anjan, Taniya Sarkar Roy, Vaskar Sarkar, and Saurav Mallik, "Combinatorial auction-based fog service allocation mechanism for IoT applications," In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 518–524. IEEE, 2020. https://doi.org/10.1109/Confluence47617.2020.9058055
[10] Abedi, Masoud, and Mohammadreza Pourkiani, "Resource allocation in combined fog-cloud scenarios by using artificial intelligence," In 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), pp. 218–222. IEEE, 2020. https://doi.org/10.1109/FMEC49853.2020.9144693
[11] C. T. Do, N. H. Tran, Chuan Pham, M. G. R. Alam, Jae Hyeok Son and C. S. Hong, "A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing," 2015 International Conference on Information Networking (ICOIN), (2015): 324–329. https://doi.org/10.1109/ICOIN.2015.7057905
[12] Aymen Abdullah Alsaffar, Hung Phuoc Pham, Choong-Seon Hong, Eui-Nam Huh, and Mohammad Aazam, "An architecture of IoT service delegation and resource allocation based on collaboration between fog and cloud computing," Mobile Information Systems, vol. 2016, ArticleID 6123234, 15 pages, 2016. https://doi.org/10.1155/2016/6123234
[13] S. Supreeth, and Kirankumari Patil, "Hybrid genetic algorithm and modified-particle swarm optimization algorithm (GA-MPSO) for predicting scheduling virtual machines in educational cloud platforms," International Journal of Emerging Technologies in Learning (iJET), vol. 17, no. 7, (2022). https://doi.org/10.3991/ijet.v17i07.29223
[14] Ni, Lina, Jinquan Zhang, Changjun Jiang, Chungang Yan, and Kan Yu. "Resource allocation strategy in fog computing based on priced timed petri nets," IEEE Internet of Things Journal vol. 4, no. 5, (2017): 1216–1228. https://doi.org/10.1109/JIOT.2017.2709814
[15] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining stackelberg game and matching," in IEEE Internet of Things Journal, vol. 4, no. 5, (2017): 1204–1215. https://doi.org/10.1109/JIOT.2017.2688925
[16] Zhang, Yaomin, Haijun Zhang, Keping Long, XiaomingXie, and Victor CM Leung. "Resource allocation in software defined fog vehicular networks," In Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, (2017): 71–76. https://doi.org/10.1145/3132340.3132357
[17] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based resource allocation for edge computing: A market equilibrium approach," in IEEE Transactions on Cloud Computing, vol. 9, no. 1, (2021): 302–317. https://doi.org/10.1109/TCC.2018.2844379
[18] Sood, Sandeep K., and Kiran Deep Singh. "SNA based resource optimization in optical network using fog and cloud computing," Optical Switching and Networking, vol. 33, (2019): 114–121. https://doi.org/10.1016/j.osn.2017.12.007
[19] C. Mouradian et al., "An IoT platform-as-a-service for NFV-based hybrid cloud/fog systems," in IEEE Internet of Things Journal, vol. 7, no. 7, (2020): 6102–6115. https://doi.org/10.1109/JIOT.2020.2968235
[20] Harshit Gupta, Amir VahidDastjerdi, Soumya K. Ghosh, and Raj Kumar Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments." CoRR abs/1606.02007 (2016).
[21] Rahbari, Dadmehr, and Mohsen Nickray. "Low-latency and energy-efficient scheduling in fog-based IoT applications," Turkish Journal of Electrical Engineering & Computer Sciences, vol. 27, no. 2, (2019): 1406–1427. https://doi.org/10.3906/elk-1810-47
[22] Bittencourt LF, Diaz-Montes J, Buyya R, Rana OF, and Parashar M. "Mobility-aware application scheduling in fog computing," IEEE Cloud Computing, vol. 4, no. 2, (2017). https://doi.org/10.1109/MCC.2017.27

[23] S. Supreeth, and K. Patil. "VM Scheduling for Efficient Dynamically Migrated Virtual Machines (VMS-EDMVM) in Cloud Computing Environment," KSII Transactions on Internet and Information Systems, vol. 16, no. 6, pp. 1892–1912, (2022). https://doi.org/10.3837/tiis.2022.06.007
[24] S. Supreeth, Kirankumari Patil, Shantala Devi Patil, and Rohith S, "Comparative approach for VM Scheduling using Modified Particle Swarm Optimization and Genetic Algorithm in Cloud Computing," IEEE International Conference on Data Science and Information System (ICDSIS-2022), 29–30 July 2022.
[25] G. Shruthi, Monica R. Mundada, B. J. Sowmya, and S. Supreeth, "Mayfly Taylor Optimisation-Based Scheduling Algorithm with Deep Reinforcement Learning for Dynamic Scheduling in Fog-Cloud Computing," Applied Computational Intelligence and Soft Computing, vol. 2022, Article ID 2131699, 17 pages, 2022. https://doi.org/10.1155/2022/2131699

7 Authors

Ms. Shruthi G. holds M. Tech. degree in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belgaum, and currently she is pursuing Doctoral program (Ph. D.) in VTU and an Assistant Professor with the School of Computer Science and Engineering, REVA University, Bengaluru, India. E-mail: g.shruthi466@gmail.com

Dr. Monica R. Mundada is working as a Professor in Computer Science Department of Ramaiah Institute of Technology. Her areas of interest include theory of computation, wireless sensor networks, computer networks, data communications, and IOT.

Mr. Supreeth S. holds M. Tech. degree in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belgaum, and currently he is pursuing his Doctoral program (Ph. D.) in VTU and an Assistant Professor with the School of Computer Science and Engineering, REVA University, Bengaluru, India.

Article submitted 2022-05-10. Resubmitted 2022-06-22. Final acceptance 2022-06-25. Final version published as submitted by the authors.