International Journal of Mathematical, Engineering and Management Sciences
Vol. 7, No. 5, 697-716, 2022
https://doi.org/10.33889/IJMEMS.2022.7.5.046

A Layer & Request Priority-based Framework for Dynamic Resource Allocation in Cloud- Fog - Edge Hybrid Computing Environment

Sandip Kumar Patel
U & P.U. Patel Department of Computer Engineering,
Charotar University of Science & Technology, CHARUSAT Campus, Gujarat, India.
Corresponding author: 17drce006@charusat.edu.in, sandippatel.it@charusat.ac.in

Ritesh Patel
U & P.U. Patel Department of Computer Engineering,
Charotar University of Science & Technology, CHARUSAT Campus, Gujarat, India.
E-mail: riteshpatel.ce@charusat.ac.in

(Received on March 02, 2022; Accepted on July 18, 2022)

Abstract
One of the most promising frameworks is the fog computing paradigm for time-sensitive applications such as IoT (Internet of Things). Though it is an extended type of computing paradigm, which is mainly used to support cloud computing for executing deadline-based user requirements in IoT applications. However, there are certain challenges related to the hybrid IoT -cloud environment such as poor latency, increased execution time, computational burden and overload on the computing nodes. This paper offers A Layer & Request priority-based framework for Dynamic Resource Allocation Method (LP-DRAM), a new approach based on layer priority for ensuring effective resource allocation in a fog-cloud architecture. By performing load balancing across the computer nodes, the suggested method achieves an effective resource allocation. Unlike conventional resource allocation techniques, the proposed work assumes that the node type and the location are not fixed. The tasks are allocated based on two constrain, duration and layer priority basis i.e, the tasks are initially assigned to edge computing nodes and based on the resource availability in edge nodes, the tasks are further allocated to fog and cloud computing nodes. The proposed approach's performance was analyzed by comparing it to existing methodologies such as First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD), Best Fit Decreasing (BFD), and DRAM techniques to validate the effectiveness of the proposed LP-DRAM.

Keywords- Fog computing, Cloud computing, Edge computing, Dynamic resource allocation, Load balancing.

1. Introduction

Over the past few decades, the IoT has gained huge significance among various researchers and the need for developing new IoT devices for different domains is increasing extensively (Jo & Kim, 2019). Since IoT is an integrated network of various devices, it generates a significant amount of large data. It's a challenging task to compute such a vast amount of data derived from a variety of sources like sensors and actuators in IoT applications using conventional data processing environments (Kong et al., 2017). Despite the traditional computing approach is efficient in computing the data for IoT applications, the ever-increasing demand and resources lead to excessive energy consumption which ultimately results in the performance deterioration of the computation process (Sarkar et al., 2018). Hence, how to enhance the computational efficiency of IoT applications is still a prominent issue to be resolved (Azam et al., 2018; Atlam et al., 2018). Instead of using exclusively cloud environments, the fog computing paradigm increases the computational process by employing a decentralized computing architecture where storage and applications are distributed across the IoT and the cloud nodes (Bonomi et al., 2012). To minimize the latency, computational time, energy utilization, space, and operational costs, the fog nodes are situated nearest to the smart nodes. (Pande et al., 2016; Manasrah & Gupta, 2019). The decentralized architecture will enable the fog nodes to process the data faster compared to cloud environments which forward the

request to the cloud for further computation. A fog environment provides user mobility support and allows location tracking along with interoperability, high scalability and low latency which is not feasible in cloud computing systems (Mouradian et al., 2017; Dastjerdi et al., 2016). Despite the advantages, fog computation suffers from the problem of high-power utilization. The power utilized by fog nodes is higher and an efficient resource allocation might have an impact on the lifetime of fog nodes. Besides, due to the uncertainties associated with fog environments such as high variability and unpredictable nature of fog nodes, proper resource allocation becomes a difficult job in fog computing. Hence, appropriate control of fog devices is necessary for improving the efficiency of fog computing (Jian et al., 2019). Some of the prominent resource allocation and management issues such as allocation, scheduling and provisioning of resources need to be discussed.

In fog systems, the routers are considered to be the possible latent physical servers capable of providing resources to fog systems at the network's edge (Wang et al., 2016; Yi et al., 2015). These routers are capable of enhancing the computational efficiency and storage capabilities in the fog systems which can be completely used as the computing nodes. Varied IoT devices have different application requirements, particularly for real-time IoT applications; as a result, these applications rely on edge computing nodes to host them. (Xu et al., 2017; Xu et al., 2017). Users of fog systems can access and utilize network resources in a similar way to a cloud environment and this technique is also applicable for on-demand dynamic resources (Bonomi et al., 2014). For IoT based applications, the resource allocation should be performed for both conditions i.e., centralized and the geo-distributed computing nodes. For appropriate resource allocation, the resource managers should select potential computing nodes for hosting fog services using various techniques for allocating resources.

There are two important techniques, resource allocation and scheduling which are used widely to control the data centres, which are responsible for reducing carbon emissions, increasing resource utilization, and ensuring data centre load balance (Xu et al., 2020; Yu et al., 2016). In cloud computing, the purpose of resource allocation is to increase the number of machines that are physically active by distributing them in a balanced manner and to manage the workload of the machines effectively to prevent any overhead of resource usage (Sahu et al., 2013; Soni & Kalra, 2014; Xu et al., 2015; Li & Zhang, 2016). Whereas in fog computing systems, the complexity of resource allocation increases since the applications can only respond to the compute nodes in fog computing which are distributed in a decentralized manner. As a result, developing an effective resource distribution strategy for IoT based applications is critical. This paper presents a new framework for managing dynamic resource distribution in cloud-fog-edge computing environments.

2. Review of Fog Computing Environment

Fog computing incorporates various accessible system components of IoT applications such as servers, smart gateways, routers, switches, and other devices integrated with the cloud data centre (Miah et al., 2018; Deng et al., 2018). Due to the heterogeneity associated with various IoT devices and due to dynamic requirements and unpredictable fog nodes, it requires a potentially stable dynamic resource allocation strategy for improving the performance of fog computing systems (Aazam & Huh, 2015; Ni et al., 2017). The system structure of the fog environment is discussed in this section, as well as a summary of existing literature on resource allocation challenges in the fog environment.

2.1 Fog Computing Architecture

The fog environment architecture is made up of four levels in general namely: the cloud, fog, edge or service and the IoT application level. The cloud tier consists of various active virtual machines as well as a

significant quantity of physical resources making it an appropriate layer for hosting the activities with high storage abilities and with high-density computation (Xu et al., 2018). The tasks which require less computing resources and have the jobs with the highest time urgency are handled at the edge, while jobs with lower time urgency are handled by fog and cloud layers. The service tier is integrated with the IoT tier for processing a larger amount of requests. This is accomplished by determining the most appropriate intermediate computing nodes and the requests are processed on a priority basis by the fog and cloud environments.

2.2 Related Works

Feng et al. (2019) presented a novel mechanism for managing risks associated with fog computing environments. The service provider present in fog computing performs dynamic optimization of computational resources for enhancing the sustainability of fog systems with a huge amount of fog nodes. The provider makes dynamic judgments for each fog node in the system to prevent it from losses caused by cyber-attacks. Furthermore, a Stackelberg game was used to demonstrate a variety of cognitive judgement dilemmas. It was observed from the experimental analysis that the Stackelberg equilibrium is a unique approach whose analytical and experimental results were analyzed for improving cyber security in fog computing systems.

Bashir et al. (2019) proposed a dynamic resource distribution method for cloud environments and fog nodes in IoT applications using logistic regression and multi decision approach. In the proposed approach, the ranks of the fog nodes are formulated using TOPSIS for identifying a potential and robust fog node for handling incoming service requests. The fog computing environment was designed using a different set of fog nodes which processes the service requests of various users. To estimate the load balancing ability of fog nodes and to update the results to make the next decision, logistic regression is utilized. The simulation evaluation shows that the proposed strategy greatly improves performance with a 98.25 % accuracy. Table 1 is representing a summary of existing resource allocation strategies for computing in a cloud-fog environment.

Chang et al. (2020) suggested a differential evolution methodology for a fog platform based on IoT and using several mobile devices. In these devices, the resources for computation and offloading decisions are coordinated dynamically and are optimally allocated for satisfying the ever-increasing computational requirements. The preliminary objective of this research was to reduce the operational costs by minimizing the energy utilization, latency, and weights of the multiple devices. To achieve this, the study used a resource provisioning approach which was based on the Lyapunov optimization. The main issue was divided into various multiple subproblems by deriving the arbitrary cap of the drift-plus-penalty equation and was addressed appropriately. The proposed approach's performance was measured, and the findings confirmed that it was effective.

Naha et al. (2020) addressed the issue of fulfilling the dynamic user requirements by using a novel approach to resource allocation and provisioning techniques. The resources are ranked in a hierarchical manner. The proposed algorithms were analyzed using a platform for modelling a real-time fog environment by expanding the CloudSim toolset. The experimental findings indicate that the presented methodology outperformed other current approaches in terms of operational cost, network delay and overall data processing time. In comparison to other algorithms, the operational cost and overall processing time were both lowered by 15% and 12%.

Table 1. Summary of relevant works.

Year	Method Applied/ Proposed algorithms	Infrastructure	Resource Allocation	Load Balancing	Evaluation Process
Gawali & Shinde et al. (2018)	1. Analytical hierarchy process 2. BATS + BAR Algorithm 3. Longest expected processing time preemption 4. divide-and-conquer methods	Cloud	Yes	No	Simulation (CloudSim)
Zheng et al. (2014)	Description of issues in Multiobjective Virtual Machine Development	Cloud	Yes	No	Simulation (CloudSim)
Liu et al. (2019)	ϵ-greedy Q-Learning and MonteCarlo	Edge	Yes	No	Simulation
Saraswathi et al. (2015)	Allocate resources based on Job Priority	Cloud	Yes	No	Simulation (CloudSim)
Di et al. (2018)	Resource Allocation on different Power splitting receiver architecture	Fog	Yes	No	Simulation
Dam et al. (2018)	Used Simulated Annealing for the best possible solution	Cloud	Yes	Yes	CloudAnalyst
Khattak et al. (2019)	Heart rate, Patient categories and utilization of foglets	Fog	Yes	Yes	CloudSim & iFogSim
Rafique et al. (2019)	Used Hybrid approch of MPSO & MCSO	Fog	Yes	Yes	iFogSim
Li et al. (2020)	Intermediary node to get information of a node and Naive Bayes for classification of node	Edge	Yes	Yes	Simulation*
Bukhsh et al. (2018)	Used PSO-SA for resource distribution	Fog	Yes	Yes	CloudAnalyst
Xu et al. (2018)	DRAM (Based on computing node, Static and Dynamic allocation)	Fog	Yes	Yes	CloudSim
Deng et al. (2016)	Generalized Benders Decomposition (GBD), Hungarian Algorithm	Fog + Cloud	Yes	Yes	Simulation*
Yin et al. (2017)	Hybrid alternating direction method of multipliers (HADMM)	Fog	Yes	No	Simulation*
Lakzaei et al. (2022)	MapReduce and DVFS techniques are used for optimal resource and energy management.	Fog + Cloud	Yes	Yes	CloudSim
Wadhwa & Aron (2021)	TRAM is used along with the expectation-maximization (EM) algorithm to level existing tasks	Fog + Cloud	Yes	No	iFogSim

In fog computing, an LB and optimization strategy (LBOS) was presented by (Talaat et al., 2020), which included dynamic resource allocation approaches, genetic algorithms, and reinforcement learning. The LBOS monitor in the network traffic aggregates the data related to the load on each data server using the dynamic resource allocation technique. This method improves the performance of the resource allocation. Accordingly, it was noted that the proposed LBOS framework was effective for live applications in fog environments such as healthcare systems. The proposed methodology included three layers: an Internet of Things, fog, and cloud. The proposed technique increases QoS, according to the findings of an experimental investigation in the hybrid computing environment by reducing response time and cost of allocation, and it achieves 85.71 % of the best load balancing level.

Farooq & Zhu (2020) presented an optimal progressive resource sharing and pricing method for IoT applications in fog environments according to QoE. A pricing policy was used in this approach which is dependent on the IoT apps' quality of experience, and as a result of this distribution, the computational requests were processed with reduced time compared to other approaches. Based on the findings of

simulation analysis, the effectiveness of the proposed strategy was confirmed, and the findings reveal that the proposed strategy greatly outperformed alternative techniques.

3. Problem Formulation

The proposed work's preliminary goal is to effectively distribute the resources in a fog environment for satisfying the dynamic user requirements with reduced execution time, cost, and delay. However, there are certain challenges which restrict the optimal resource provision in a fog environment. Since the nature of the fog environment is highly distributed in nature It is challenging to process with a finite set of resources and vast volumes of data. Furthermore, fog computing has a finite number of resources. The focus of this research is on resource allocation in time-constrained applications like IoT networks, where user requests cannot be fulfilled without the usage of fog servers in a cloud tier. According to the changing pattern of the requirements and the processing time, the requests need to be processed in the cloud-fog hybrid environment. It can be inferred that; existing literary works have not analysed the resource allocation considering the dynamic users' requirements. It is not practically feasible to perform efficient resource allocation for time-constraint applications without managing the frequent changes in user requirements with limited resources at the fog layer.

This research identifies some of the resource distribution issues in a fog environment:

- Resource management: How effectively the resources are used during resource allocation? This is one of the primary issues considered in the resource distribution process.
- Scalability: In a fog computing environment, the proposed technique should be scalable enough to handle resource allocation.
- Energy efficiency: To improve the performance of fog devices, energy consumption must be optimized during resource allocation.
- Load balancing: To increase the execution time and efficiency of the resource allocation process, the overhead on computing fog nodes should be lowered.

4. Research Methodology

The proposed work aims to offer a dynamic resource provisioning method for maximizing the use of available resources at the fog and edge layers to reduce fog node latency and execution time in fog computing. The proposed research work used an optimal resource allocation strategy where the request is processed by a three-layer processing network. The tasks are initially distributed to the edge devices and if there is a lack of availability of required resources the tasks are further transferred to the fog computing layer for computation. If the fog devices are not compatible in processing a high number of responsibilities due to limited resource availability, then the jobs are processed by the cloud computing layer. This technique aims to cut the computational strain on the cloud computing tier, as well as the execution time and latency. The study identifies certain prominent drawbacks associated with existing works associated to load balancing and resource provision in fog environments: As observed in (Xu et al., 2018) the node type is fixed for each request for execution and the request priority is not defined. Also, the migration is fixed for specific layers only i.e., (fog-to-fog layer). The proposed work aims to overcome the limitations by not fixing the node types and by optimizing the request priority. The following sub-sections cover a thorough description of the proposed work:

4.1 System Description

In this hybrid architecture of cloud, fog, edge & IoT, the user requirements or requests are forwarded to the edge devices for computation. Then edge devices sent the data to the fog layer for further computation instead of the cloud which will significantly improve the performance. This scenario is more suitable for real-time

analysis-based applications. The fog layer will handle all the processes of requests and jobs by authenticating user identities and scheduling jobs to both fog-cloud layers. In fog computing, fog nodes play a crucial role in data acquisition. When fog devices get data from the edge, they store it and analyze it based on the user's demand. After processing, the raw data is forward to the fog environment for execution which is again forward to the cloud environment in case of limited resource availability at the fog layer. Fog devices have their operating system and must support virtualization to execute multiple requests simultaneously. The fog devices are incorporated with restricted storage, memory capacity, bandwidth, and processing capability. According to Figure 1, the calculations are spread over three layers: edge, fog and cloud.

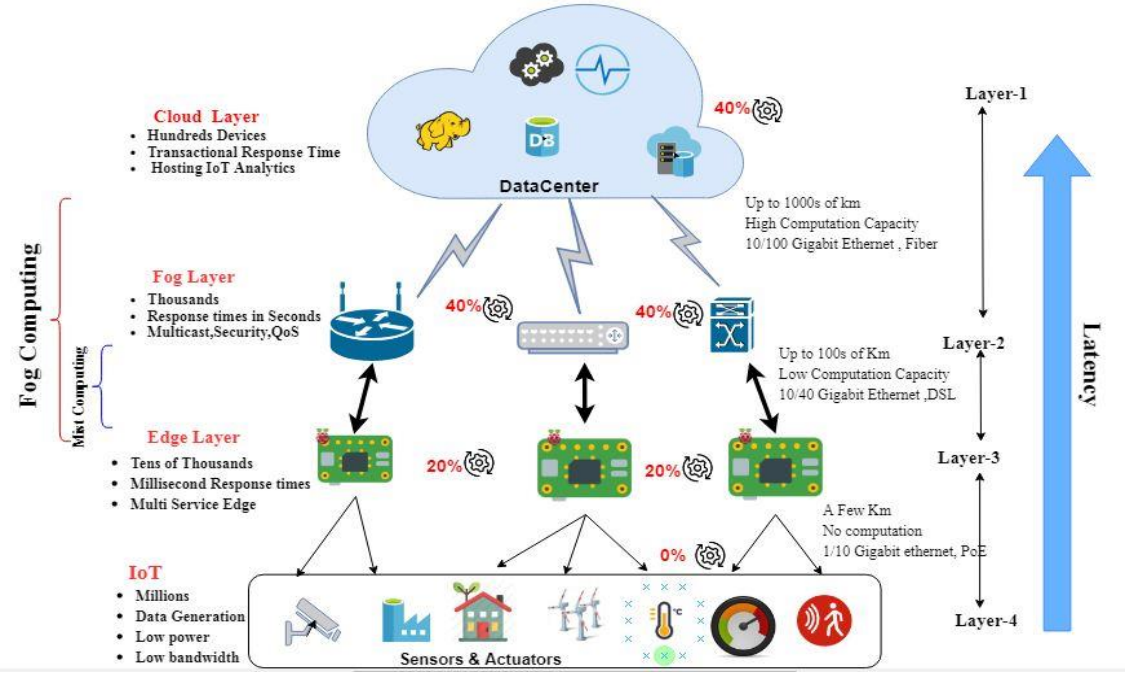

Figure 1. Proposed system architecture for fog computing.

Latency will be increased as the computation task will start to execute from the bottom to the top layers. For instance, when layer four will generate data, it will forward it to layer three, where it has some computing capacity. If requested resource requirements are not satisfied, then it will be sent to respectively layer 2 or 1. Ultimately, it is depended on the type of application and their needs. Any device can act as fog or edge, which had minimum computing resources like the raspberry pi, Arduino, layer-3 switch and nanocomputer. These four layers are classified based on the computational capacity of each.

4.2 Fog execution Model

The fog nodes that communicate to the application are able to handle the offloaded queries and perform the processing work. When the fog device offloads the tasks 't' to either the fog or cloud layer, the requirements of the task are distributed to the fog or cloud layer via wireless sensors. If the job 't' is allocated to the fog device, then according to the proposed layer priority the fog node decides whether the task can be executed in the fog layer or cloud layer, depending on the time it took to execute. The processing time taken by the fog node to run a task depends on two main parameters namely, the computation time (T^{i}_{comp}) and the time

taken for transmission (T^i_{data}). Assuming that transmission delay is negligible the execution time in fog node 'i' for a tsak 't' is evaluated as:

$$T_t^i = T^i_{comp} + T^i_{data} \quad (1)$$

Where T_t^i is the execution time of the fog nodes for computing task 't' by a fog node 'i'. Another important parameter that defines the efficacy of the fog execution model is the energy used by the fog nodes which implies the transmission rate of the fog devices. Moreover, it is noticed that the fog nodes must have sufficient computational resources to process the jobs offloaded by the fog devices else the jobs are assigned to the cloud. Hence the computation requirements of the fog environment must be well satisfied with appropriate resource allocation and load balancing as shown in equation 8. In this case, the actual processing time and the energy consumed by the fog devices can be evaluated respectively as shown in equations 2 and 3.

$$T_t^{i.} = I\,(-M \le S_n \le -1)\, T^i_{t,m} + I\,(S_n < -M)\, T^i_{t,\,m+1} \quad (2)$$

$$E_n = I\,(-M \le S_n \le -1)\, E_{n,m} + I\,(S_n < -M)\, E_{n,m+1} \quad (3)$$

where, $T_t^{i.}$ is the total execution time of the fog node 'i', '$T^i_{t,m}$ and '$T^i_{t,\,m+1}$' is the time cost of offloading the task from fog nodes to either fog or cloud layer. E_n is the amount of energy used by the fog nodes, where n is the number of nodes in the fog.

4.3 Task Allocation Approach in the Fog Computing Environment

The computational framework for fog performs efficient task distributions among all the nodes present in both the cloud layer and fog layer. Since the fog computing environment is highly diversified in nature, and the execution time differs from one commuting node to another, it is difficult to completely utilize the available resources. The main intent of the suggested work is to efficiently balance the workload among computing nodes in order to reduce overutilization and prevent the underutilization of existing resources.

Let $X_n^m(t)$ be the binary variable to determine whether b_n $(1 \le n \le N)$ is allocated to the Computation devices j_m $(1 \le m \le M)$ for instantaneous time t, which is evaluated as:

$$X_n^m(t) = \begin{cases} 1, & \text{if } b_n \text{ is assigned to } j_m \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where, B is the number of services set $B = \{b_1, b_2, \ldots b_{N`}\}$ of the requests and N is the total fog services produced by IoT applications. The physical machines and computing nodes on the cloud, fog and edge layers are used to leverage the allocation of available resources for the fog services. Consider that both fog and cloud levels have 'm' computing nodes defined as $J = \{j_1, j_2, \ldots, j_M\}$, then the fog services' resource capability and resource requirements are determined as;

$$\beta_\omega^m = \begin{cases} 1, & \text{if } j_m \text{ is } w^{\text{th}} \text{ compute node type} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$q_m(t) = \begin{cases} 1, & \text{if } nb_n > 0 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

As illustrated in the equation below, optimal usage of available resources for compute nodes in a fog environment of w^{th} type for a specific time 't' is determined.

$$PU_{\omega}(t) = \frac{1}{a_{\omega}(t)} \sum_{m=1}^{M} \sum_{n=1}^{N} X_n^m(t) \cdot pu_m(t) \cdot q_m(t) \quad (7)$$

The difference of load distribution of j_m for a time 't' which is determined by resource utilization variance is given as:

$$Lb_m(t) = \left(pu_m(t) - \sum_{\omega=1}^{W} PU_w(t) \cdot \beta_{\omega}^m \right)^2 \quad (8)$$

The average value of the measured variance for all computing nodes is given as:

$$LB_{\omega}(t) = \frac{1}{a_{\omega}(t)} \sum_{m=1}^{M} \sum_{n=1}^{N} X_{nm}(t) \cdot Lb_m(t) \cdot q_m(t) \quad (9)$$

Similarly, the load balance variance between the execution time interval (T, T_0) is evaluated.

$$LB_{\omega} = \frac{1}{T \cdot T_0} \int_{T_0}^{T} LB_{\omega}(t) dt \quad (10)$$

Based on these values, the constraint for reducing the variance of equal load distribution is given

$$\min \quad LB_w, \forall w = 1, \ldots . W \quad (11)$$

$$\text{With the condition} \quad a_{\omega}(t) \leq \sum_{m=1}^{m} \beta_{\omega}^m \quad (12)$$

$$\sum_{n=1}^{N} a_n \leq \sum_{m=1}^{M} C_m \quad (13)$$

where, $\sum_{n=1}^{N} q_n$ defines the resource demands for all fog services and $\sum_{m=1}^{M} c_m$ defines the capacity of the nodes.

$$\sum_{n=1}^{N} x_n^m a_n \leq C_m \quad (14)$$

where, $\sum_{n=1}^{N} X_n^m$ q_n defines the resource needs for fog services assigned to m number of computational nodes.

4.4 Dynamic Resource Allocation in Fog Environment using LP-DRAM

The research aims to accomplish effective resource provisioning at the fog layer and to equal distribution of workload amongst processing devices in the three layers hybrid environment. For effective resource allocation, the study proposes LP-DRAM. The novelty of the proposed LP- DRAM architecture is that it does not fix the node type and number of nodes for execution, unlike the previous DRAM process. Also,

the study considers the possibility of fog-to-cloud migration for non-critical interests so that the fog layer can handle maximum critical requests and there is less burden on the cloud layer. This is mainly done to improve the latency and to decrease the execution duration and computational burden on the cloud environment. The process flow of the proposed LP-DRAM is illustrated in Figure 2.

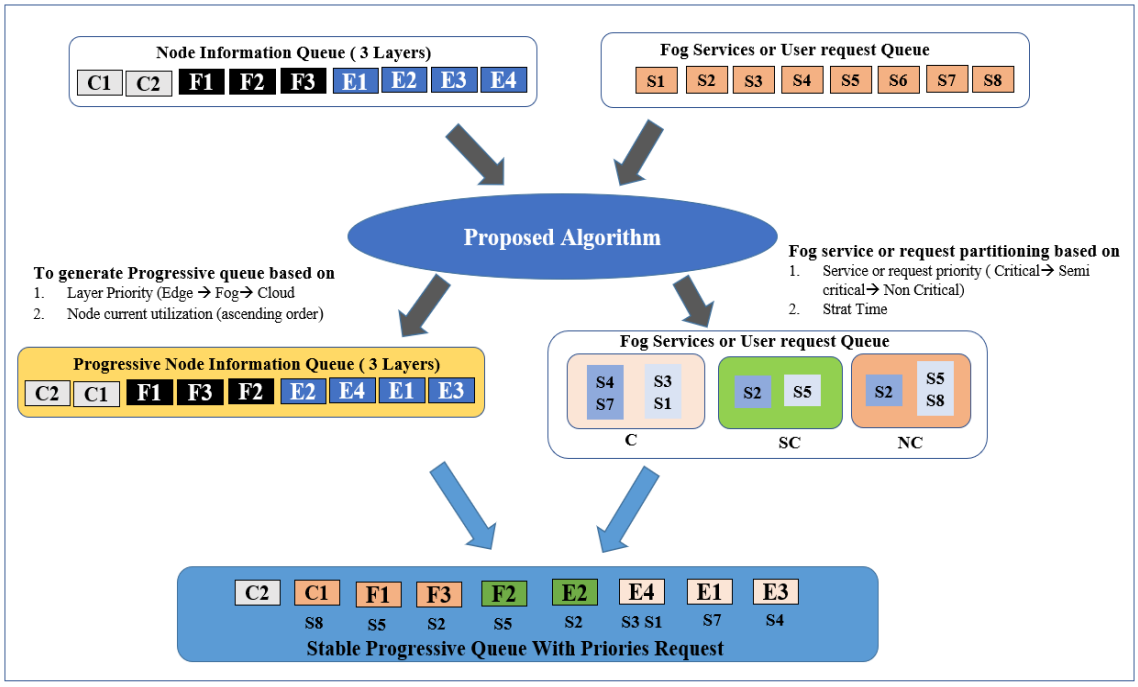

Figure 2. Proposed system architecture for fog computing.

Table 2. Used notations.

T_t^i :	Total execution Time
E_n :	Energy Consumed
M :	The no of computational nodes
J :	a collection of processing elements
B :	Set of services $B = \{ b_1, b_2, \ldots b_N \}$
AW :	Adaptive weight
$PU_w(t)$:	Use of Resources for w^{th} class of computational node at moment t
$Lb_m(t)$:	Task scheduling Variation j_m at moment t
$LB_w(t)$:	Task scheduling Variation w^{th} type of computational node at moment t
W_{nj} :	The existing workload of j^{th} node (initially whose value is 0)
ET_{njRj} :	The upcoming workload of request
Cap_{nj} :	The capacity of the node
S :	*Classified set of fog request* $S= (S_1, S_2, S_3, \ldots S_N)$.
R :	Request Data Set
P_t :	Progressive queue at time t

The requests analyzer is priority-based and is capable of analyzing the requests according to the requirements and to either choose to accept or reject the tasks. It processes the data based on the priority assigned; Normal, critical, and highly demanded. The task scheduler assigns the tasks according to the hierarchy of the proposed work i.e., it is initially assigned to edge, fog and then cloud devices. As mentioned earlier, the resource manager will allocate the resources of each task to either fog or cloud nodes and the optimal utilization of the available resources for the fog computing nodes is computed as shown in equation 7. If the fog device is not able to compute the task at a given time, due to a lack of resources available for processing this task then the fog device will reject the task and the task is processed by the cloud computing device. However, this process consumes more time and impacts the actual quality of the processing system. Hence in the proposed work, the priority is not fixed, and the request analyzer will evaluate the requests based on the order of arrival. Besides, the cloud layer must possess the necessary resources for executing the task. The analyzer will quickly determine whether the task can be processed by the fog or not and correspondingly allocate the resource to each task which is activated in the fog node as shown in equation 14.

Hypothesis 1: Based on Layer Priority

This hypothesis is based on the priority assigned to three tiers of the system including, Edge, fog and cloud. To execute fog services and effectively distribute resources, the network has several types of computing nodes. Fog services are classed according to the resource requirements of the node type. Here, in this work, the node type is not fixed and at a specific layer, only adaptive weights are used without fixed nodes. The adaptive weight is calculated as:

$$AW = \frac{W_{n_j} + ET_{n_j R_j}}{Cap_{nj}} \tag{15}$$

where, W_{nj} is defined as the existing workload of j^{th} node (initially whose value is 0), ET_{njRj} represents the upcoming workload of request, Cap_{nj} is the capacity of the node.

4.4.2 Fog Service Segmentation

Fog services derived through diverse IoT applications have a wide range of user needs and require a huge spectrum of processing resources. In order to process these requests, the fog services must select different types of computing nodes. Actual machines, intermediate (fog), and edge nodes near the sensors are among the N types of virtual machines in the cloud infrastructure. In the proposed work, it is assumed that the node type is not fixed, so fog services can be partitioned as follows: $S= (S_1, S_2, S_3, ...S_N)$. Furthermore, fog services that are part of the same set have different resource requirements and execution times. To assess the resource distribution for a similar group of fog application requests must be divided into various subgroups based on starting time of the nodes to occupy the resource elements, as indicated in equation 8, in order to provide load balancing for fog application requests in the similar group.

The acquisition of fog services subsets is depicted in Algorithm 1. Here, the resource requirements acquired from IoT applications are the input, and the classified set of fog requests 'S' is the output. Once the fog operations are differentiated and the resources are granted at the fog layer, the processed requests are evaluated based on certain QoS dimensions such as execution time, makespan time, TAT, execution cost and throughput. The proposed LP-DRAM framework is analyzed with respect to the hypothesis; Based on layer priority and using adaptive weight without fixed nodes.

Algorithm:- 1 *Request subset partitioning based on priority and Start time*

Input:- *IoT application resource needs (Request Dataset) R*

Output:- *Classified set of fog request* ***S***

```
1   for i = 0 to N do
2       for j = 0 to P do
    // P= priority of request (Critical, Non-Critical)
3     if Priority_i == j then
4       add r_i to S_i
5     end if
6    end for
7   end for
8  for i = 0 to P do
9    nn = 0, q = 0, S = stim_0
10     While nn < |S_si| do
11         if Stim_q ≤ S then
12           Add m^th request to set Ss_i,nn
13         else nn = nn + 1, q = q + 1, f = stim_q
14           Add the q^th request to set Ss_i,nn
15       end
16   end
17 end
18 Return S
```

Algorithm: - 2

Input: - *Available node information (current utilization of memory and CPU)*

Output: - ***Progressive*** *queue of nodes on each layer-by-layer priority with available capacity of nodes*

Two main parameters are required to design a queue

1. Current load on node
2. Current performance of node

L:- Load factor, W_t:- Weight factor, Q_t:- Performance factor, $Q_{t(avg)}$:- Average of current response time

P_t:- Progressive queue

Step:-1

- Calculate the load factor L for each layer node
- L= Total resources of respective layer (e.g edge layer) – currently used resource

Step:- 2

- Calculate Q_t of node
- response time = finish time -arrival time + transmission time
- $Q_t = Q_{t(avg)}$ - Previously calculated $Q_{t(avg)}$

Algorithm: - 3

Input: - *Each layer nodes progressive queue, Classified set of fog requests subsets S*

Output: - *Resource allocation to specific fog requests, resource utilization of each layer*

0. The request is a map with resources based upon the available resources in the resource pool
1. ***for*** *i = 0 to (E, f & C)*
2. Calculate *AW for an individual node on each layer* from ***Algorithm 2***
3. end
4. *E, f & C* = arrange all the nodes in ascending order of ***AW***
5. **for** *i*=0 to *E* **do**
6. **if** ($R_{RAM} < E_{i\,(Ava\_RAM)}$ **&&** $R_{Capacity} < E_{i\,(Ava\_Capacity)}$ **&&** $R_{CPU} < E_{i\,(Ava\_CPU)}$) **then**
7. Allocate Request to E_i
8. Edgecounter ++
9. Calculate resource utilization from ***Eq. 7***
10. Break

    end if
11. **end for**

- $Q_t = Q_t / (Q_{t(avg)}) * 100$ // Q_t in terms of %

Step:-3

- Calculate W_t
- $W_t = L - Q_t$
- If $W_t < 0$ then $W_t = 0$

Step:-4

- Now find out the minimum W_t from all existing nodes
- Node with W_t=0 should not consider for in calculation
- $Min\_W_t = min(all\ W_t)$
- $Min\_factor = Min\_W_t$

Step:- 5

- $P_t = W_t / Min\_factor$

Now design a progressive queue based on the value of P_t.

12. **Return** E_i
13. **for** i=0 to f **do**
14. **if** ($R_{RAM} < f_{i\ (Availble\_RAM)}$ **&&** $R_{Capacity} < f_{i\ (Ava\_Capacity)}$ **&&** $R_{CPU} < f_{i\ (Ava\_CPU)}$) **then**
15. Allocate Request to f_i
16. Fogcounter++
17. Calculate resource utilization from ***Eq.7***
18. Break
19. **end if**
20. **end for**
21. **Return** f_i
22. **for** i=0 to C **do**
23. **if** ($R_{RAM} < C_{i\ (Availble\_RAM)}$ **&&** $R_{Capacity} < C_{i\ (Ava\_Capacity)}$ **&&** $R_{CPU} < C_{i\ (Ava\_CPU)}$) **then**
24. Allocate Request to C_i
25. Cloudcounter++
26. Calculate resource utilization from ***Eq. 7***
27. Break
28. **end if**
29. **end for**

Return C_i

5. Results and Discussion

In this section, we have discussed the results and experimental setup. The cloud simulator ifogsim is used to carry out a result analysis of the whole proposed work with three distinct kinds of processing nodes. like edge, fog and cloud. To measure the efficiency of our proposed approach, we compare and analyzed it with existing resource allocation techniques such as FF, BF, FFD, BFD, and DRAM.

5.1 Experimental Analysis

The proposed LP-DRAM (hypothesis) approach was evaluated using the ifogsim simulator. The results were evaluated for two different conditions namely with fix node type (DRAM) and without fix node type (LP-DRAM). The scenario was carried out for a different number of fog services data sets such as 500, 100, 1500, and 2000 for an edge, fog and cloud computing nodes. The reference data set used for this experiment is shared & designed by Xu et al., 2018. In existing data set is modified with two additional columns, *Priority (**C**ritical, **Non**-Critical) and Process Type*. Fog services data set include information about *Request no, Node Id, Start Time(ms), and Duration of execution (ms),* Tables 3, 4, 5, 6, and 7 show the configuration settings for experimental analysis.

Table 3. Configuration parameters for experiment.

Configuration Parameters	Value
Date set of Fog Services	{500,1000,1500,2000}
Computing Node type	{edge, fog, cloud}
Number of computing node	{11, 8, 5}
Execution time for each service	[0.1, 5]
Process type (resource requirement)	6

Table 4. Edge server configuration.

ES ID	*Capacity (MB)*	*RAM (MB)*	*CPU (MHz)*
ES1	100	70	20
ES2	190	115	50
ES3	200	125	27
ES4	240	140	50
ES5	160	100	38
ES6	290	135	67
ES7	270	130	70
ES8	160	120	68
ES9	150	88	40
ES10	280	148	55
ES11	195	125	32

Table 5. Fog server configuration.

FS_ID	*Capacity (MB)*	*RAM (MB)*	*CPU (MHz)*
FS1	300	175	72
FS2	490	155	83
FS3	320	185	97
FS4	384	200	70
FS5	425	215	100
FS6	450	170	115
FS7	500	158	94
FS8	550	225	120

Table 6. Cloud server configuration.

CS_ID	**Capacity (MB)**	**RAM (MB)**	**CPU (MHz)**
CS1	700	250	120
CS2	850	380	135
CS3	775	415	150
CS4	915	315	210
CS5	1000	512	200

Table 7. Sample process set.

Process	**Process Size (PS)**	**RAM (MB)**	**CPU Usage (%)**
P1	50	30	20
P2	43	27	18
P3	32	18	15
P4	26	15	13
P5	14	10	11
P6	12	9	8

5.2 Performance Evaluation

The performance evaluation was done based on the number of resources utilized by the computing nodes in three different layers. The existing results with fixed nodes and the proposed LP-DRAM without fixed nodes are presented in Figures 3 and 4 respectively.

Figure 3 represents the existing dynamic resource allocation method with fixed nodes. While in the proposed work, resource allocation without fixed nodes is represented in Figure 4. It can be inferred from

Figure 4 that the number of resources allocated for edge computing nodes is greater than the intermediate (fog) and cloud computing nodes. This signifies that maximum requests are processed by the edge nodes and there is a less computational burden on the intermediate and cloud nodes. This improves the execution time significantly since the majority of the tasks are processed by the edge nodes themselves.

Measurement of average resource consumption is shown in Figure 5 for different numbers of fog services using different resource allocation approaches such as FF, BF, FFD, BFD, and DRAM, by Xu et al., 2018 and the proposed methodology. It is clearly identified that the proposed LP-DRAM has the highest average resource utilization compared to other paradigms irrespective of different datasets.

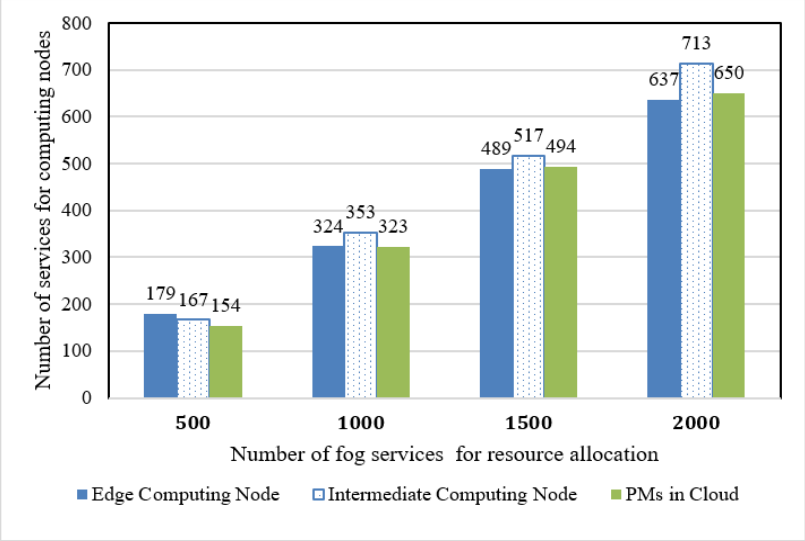

Figure 3. Fog services count for three categories of processing devices with fixed nodes type for each fog service execution (existing work by Xu et al., 2018).

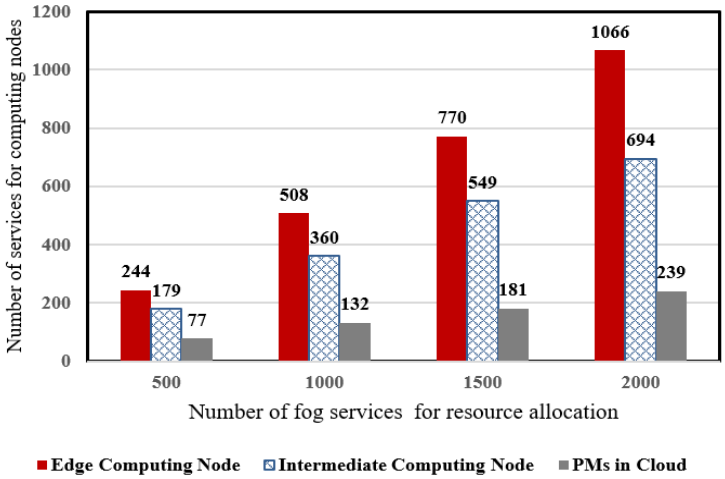

Figure 4. Fog services count for three categories of processing devices without fixed nodes (Proposed).

As observed in Figure 5, the proposed approach can utilize fewer computing nodes compared to other approaches. When 1000 fog services are used, the suggested method achieves a resource utilization efficiency of 85.89 %, which is higher than the FF method.

The suggested LP-DRAM is subjected to experimental assessment for several types of computing nodes as indicated in Figures (6,7,8,9) since the proposed technique assesses three categories of processing devices in the experimental analysis. The Figures (6, 7, 8, 9) are representing resource utilization for four different datasets which are having 500,1000,1500 and 2000 requests (fog services).

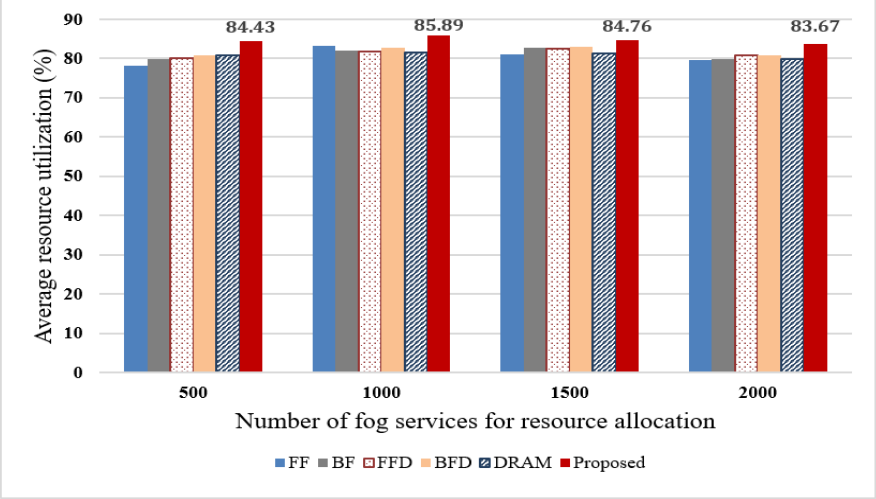

Figure 5. Analysis of average consumption of resource by existing approach vs proposed technique along with diverse statistics and without fixed nodes.

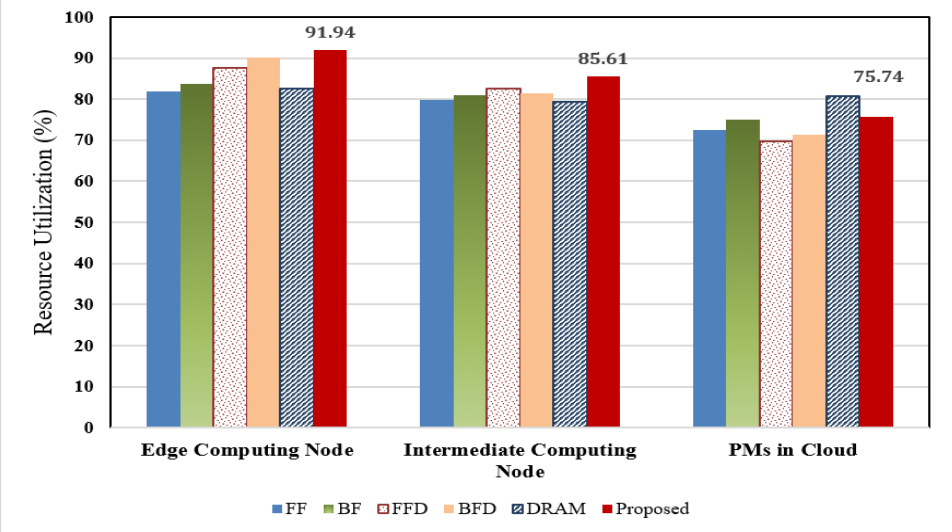

Figure 6. For 500 services of fog (dataset).

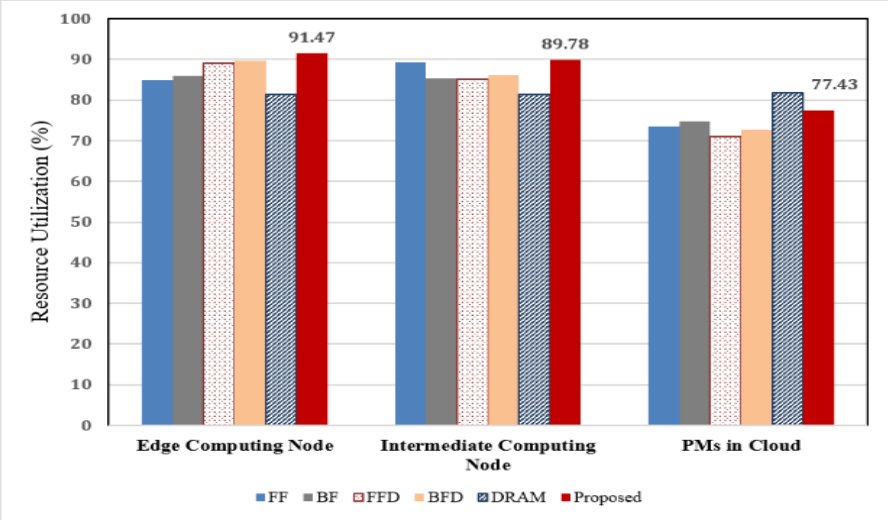

Figure 7. For 1000 services of fog (dataset).

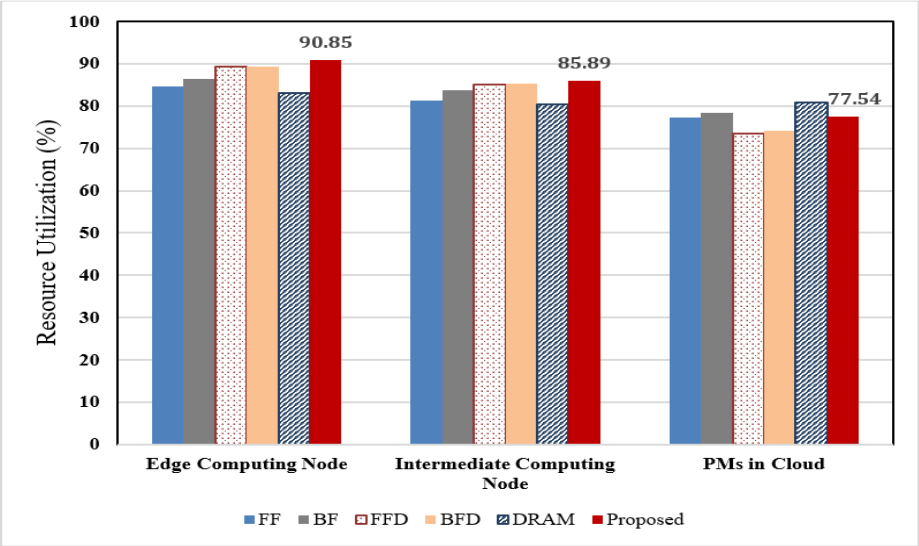

Figure 8. For 1500 services of fog (dataset).

As observed from above Figures 6, 7, 8 & 9, There is the highest resource utilization at the nearest computing edge nodes than intermediate(fog) nodes and the lowest resource utilization at cloud nodes. For instance, in Figure 9 where fog services are 2000, there is 91% resource utilization at edge layers nodes which is directly connected with Figure 4 in which out of 2000 fog services 1066 are executed at the edge layer only. So if the number of executed fog services at a particular layer is high then resource utilization will also high. Similar behaviour has been observed in other datasets as well.

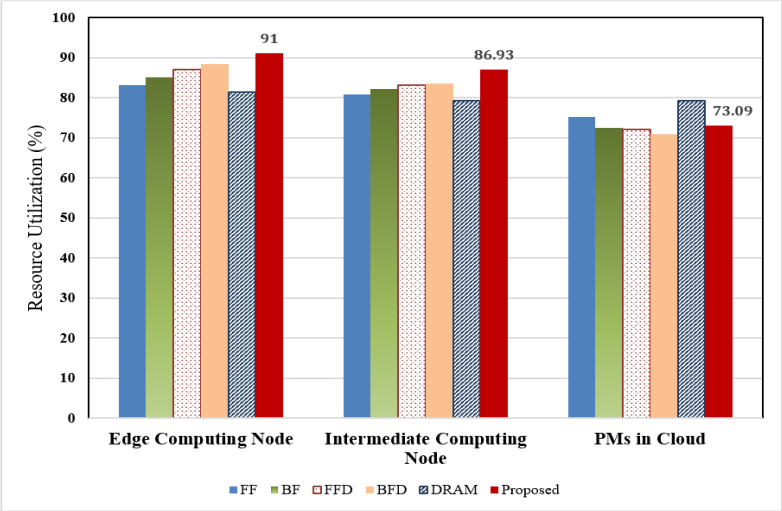

Figure 9. For 2000 services of fog (dataset).

6. Conclusion

The proposed research aims to achieve effective resource utilization by establishing a proper load balancing between the computing nodes. The study proposes a layer priority-based dynamic resource allocation method known as LP-DRAM for achieving optimal resource allocation among different nodes such as edge computing nodes, fog commuting nodes and cloud computing nodes. One of the novelties of the proposed approach is that the study does not consider any fixed node types unlike other existing resource allocation approaches and the proposed work also employs adaptive weight without fixed node and location for finding available resources for the computing nodes. The performance of the proposed approach was tested with existing methodologies such as First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD), Best Fit Decreasing (BFD), and DRAM techniques with a different set of fog services such as 500, 1000, 1500, and 2000. In comparison to fixed node strategies, the results of the experimental research reveal that the proposed strategy archives superior performance in terms of effective resource utilization (without having fixed nodes). The proposed approach surpasses earlier strategies in terms of resource allocation at the edge computing layer, especially when the number of fog services reaches 500.

This work addresses the best capacity planning in a fog computing environment with improved execution time and reduced computational burden. However, the study does not address the issue of cost optimization during resource allocation. Hence the study intends to perform cost optimization of the resource allocation and also consider the location of the fog nodes along with layer priority during resource allocation in this hybrid cloud, fog and edge architecture with real time data as a part of future work.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank the editor and anonymous reviewers for their comments that help improve the quality of this work.

References

Aazam, M., & Huh, E.N. (2015, March). Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications* (pp. 687-694). IEEE, South Korea.

Atlam, H., Walters, R., & Wills, G. (2018). Fog computing and the internet of things: A review. *Big Data and Cognitive Computing*, *2*(2), 10. https://doi.org/10.3390/bdcc2020010.

Bashir, H., Lee, S., & Kim, K.H. (2019). Resource allocation through logistic regression and multicriteria decision making method in IoT fog computing. *Transactions on Emerging Telecommunications Technologies*, *33*(2). https://doi.org/10.1002/ett.3824.

Bonomi, F., Milito, R., Natarajan, P., & Zhu, J. (2014). Fog computing: A platform for Internet of Things and Analytics. *Big Data and Internet of Things: A Roadmap for Smart Environments*, 169-186. https://doi.org/10.1007/978-3-319-05029-4_7.

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog Computing and its role in the Internet of Things. In *IEEE Workshop on Mobile Cloud Computing* (MCC) (pp. 13-16). https://doi.org/10.1145/2342509.2342513.

Bukhsh, R., Javaid, N., Ali Khan, Z., Ishmanov, F., Afzal, M.K., & Wadud, Z. (2018). Towards fast response, reduced processing and balanced load in fog-based data-driven smart grid. *Energies*, *11*(12), 3345. https://doi.org/10.3390/en11123345.

Chang, Z., Liu, L., Guo, X., & Sheng, Q. (2020). Dynamic resource allocation and computation offloading for IoT fog computing system. *IEEE Transactions on Industrial Informatics*, *17*(5), 3348-3357. https://doi.org/10.1109/tii.2020.2978946.

Dastjerdi, A.V., Gupta, H., Calheiros, R.N., Ghosh, S.K., & Buyya, R. (2016). Fog computing: Principles, architectures, and applications. In *Internet of Things* (pp. 61-75). Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-805395-9.00004-6.

Deng, R., Lu, R., Lai, C., Luan, T.H., & Liang, H. (2016). Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE Internet of Things Journal*, *3*(6), 1171-1181. https://doi.org/10.1109/jiot.2016.2565516.

Deng, Y., Chen, Z., Zhang, D., & Zhao, M. (2018). Workload scheduling toward worst-case delay and optimal utility for single-hop Fog-IoT architecture. *IET Communications*, *12*(17), 2164-2173. https://doi.org/10.1049/iet-com.2018.5077.

Di, X., Zhang, Y., Liu, T., Kang, S., & Zhao, Y. (2018). Mobile fog computing-assisted resource allocation for two-hop SWIPT OFDM networks. *Wireless Communications and Mobile Computing*, *2018*, 1-11. https://doi.org/10.1155/2018/7606513.

Feng, S., Xiong, Z., Niyato, D., & Wang, P. (2019). Dynamic resource management to defend against advanced persistent threats in fog computing: A game theoretic approach. *IEEE Transactions on Cloud Computing*, *9*(3), 995-1007. https://doi.org/10.1109/tcc.2019.2896632.

Gawali, M.B., & Shinde, S.K. (2018). Task scheduling and resource allocation in cloud computing using a heuristic approach. *Journal of Cloud Computing*, *7*(1), 1-16. https://doi.org/10.1186/s13677-018-0105-8.

Jian, C., Li, M., & Kuang, X. (2019). Edge cloud computing service composition based on modified bird swarm optimization in the internet of things. *Cluster Computing*, *22*(4), 8079-8087. https://doi.org/10.1007/s10586-017-1630-9.

Jo, D., & Kim, G.J. (2019). IoT+ AR: pervasive and augmented environments for "Digi-log" shopping experience. *Human-centric Computing and Information Sciences*, *9*(1), 1-17. https://doi.org/10.1186/s13673-018-0162-5.

Farooq, M.J., & Zhu, Q. (2020). Qoe based revenue maximizing dynamic resource allocation and pricing for fog-enabled mission-critical iot applications. *IEEE Transactions on Mobile Computing*, *20*(12), 3395-3408. https://doi.org/10.1109/tmc.2020.2999895.

Khattak, H.A., Arshad, H., Ahmed, G., Jabbar, S., Sharif, A.M., & Khalid, S. (2019). Utilization and load balancing in fog servers for health applications. *EURASIP Journal on Wireless Communications and Networking, 2019*(1), 1-12. https://doi.org/10.1186/s13638-019-1395-3.

Kong, Y., Zhang, M., & Ye, D. (2017). A belief propagation-based method for task allocation in open and dynamic cloud environments. *Knowledge-Based Systems, 115*, 123-132. https://doi.org/10.1016/j.knosys.2016.10.016.

Lakzaei, M., Sattari-Naeini, V., Sabbagh Molahosseini, A., & Javadpour, A. (2022). A joint computational and resource allocation model for fast parallel data processing in fog computing. *The Journal of Supercomputing*, *78*, 12662-12685. https://doi.org/10.1007/s11227-022-04374-x.

Li, G., Yao, Y., Wu, J., Liu, X., Sheng, X., & Lin, Q. (2020). A new load balancing strategy by task allocation in edge computing based on intermediary nodes. *EURASIP Journal on Wireless Communications and Networking, 2020*(1), 1-10. https://doi.org/10.1186/s13638-019-1624-9.

Li, S., & Zhang, Y. (2016). On-line scheduling on parallel machines to minimize the makespan. *Journal of Systems Science and Complexity, 29*(2), 472-477. https://doi.org/10.1007/s11424-015-3252-8.

Liu, X., Qin, Z., & Gao, Y. (2019, May). Resource allocation for edge computing in IoT networks via reinforcement learning. In *ICC 2019-2019 IEEE International Conference on Communications* (ICC) (pp. 1-6). IEEE. https://doi.org/10.1109/icc.2019.8761385.

Manasrah, A.M., & Gupta, B.B. (2019). An optimized service broker routing policy based on differential evolution algorithm in fog/cloud environment. *Cluster Computing, 22*(1), 1639-1653.

Mandal, G., Dam, S., Dasgupta, K., & Dutta, P. (2018, July). Load balancing strategy in cloud computing using simulated annealing. In *International Conference on Computational Intelligence, Communications, and Business Analytics* (pp. 67-81). Springer, Singapore.

Miah, M.S., Schukat, M., & Barrett, E. (2018). An enhanced sum rate in the cluster based cognitive radio relay network using the sequential approach for the future Internet of Things. *Human-centric Computing and Information Sciences, 8*(1), 1-27.

Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R.H., Morrow, M.J., & Polakos, P.A. (2017). A comprehensive survey on fog computing: State-of-the-art and research challenges. *IEEE Communications Surveys & Tutorials, 20*(1), 416-464.

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.

Ni, L., Zhang, J., Jiang, C., Yan, C., & Yu, K. (2017). Resource allocation strategy in fog computing based on priced timed petri nets. *IEEE Internet of Things Journal, 4*(5), 1216-1228.

Pande, V., Marlecha, C., & Kayte, S. (2016). A review-fog computing and its role in the internet of things. *International Journal of Engineering Research and Applications, 6*(10), 2248-96227.

Rafique, H., Shah, M.A., Islam, S.U., Maqsood, T., Khan, S., & Maple, C. (2019). A novel bio-inspired hybrid algorithm (NBIHA) for efficient resource management in fog computing. *IEEE Access, 7*, 115760-115773. https://doi.org/10.1109/access.2019.2924958.

Sahu, Y., Pateriya, R.K., & Gupta, R.K. (2013, September). Cloud server optimization with load balancing and green computing techniques using dynamic compare and balance algorithm. In *2013 5th International Conference and Computational Intelligence and Communication Networks* (pp. 527-531). IEEE. https://doi.org/10.1109/ cicn.2013.114.

Saraswathi, A.T., Kalaashri, Y.R., & Padmavathi, S. (2015). Dynamic resource allocation scheme in cloud computing. *Procedia Computer Science, 47*, 30-36. https://doi.org/10.1016/j.procs.2015.03.180.

Sarkar, S., Chatterjee, S., & Misra, S. (2018). Assessment of the suitability of fog computing in the context of Internet of Things. *IEEE Transactions on Cloud Computing, 6*(1), 46–59. https://doi.org/10.1109/ tcc.2015.2485206.

Soni, G., & Kalra, M. (2014, February). A novel approach for load balancing in cloud data center. In *2014 IEEE International Advance Computing Conference* (IACC) (pp. 807-812). IEEE, India. https://doi.org/10.1109/iadcc.2014.6779427.

Talaat, F.M., Saraya, M.S., Saleh, A.I., Ali, H.A., & Ali, S.H. (2020). A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment. *Journal of Ambient Intelligence and Humanized Computing, 11*(11), 4951-4966. https://doi.org/10.1007/s12652-020-01768-8.

Wang, S., Lei, T., Zhang, L., Hsu, C.H., & Yang, F. (2016). Offloading mobile data traffic for QoS-aware service provision in vehicular cyber-physical systems. *Future Generation Computer Systems, 61*, 118-127. https://doi.org/10.1016/j.future.2015.10.004.

Wadhwa, H., & Aron, R. (2021). TRAM: Technique for resource allocation and management in fog computing environment. *The Journal of Supercomputing, 78*(1), 667-690. https://doi.org/10.1007/s11227-021-03885-3.

Xu, X., Dou, W., Zhang, X., & Chen, J. (2015). EnReal: An energy-aware resource allocation method for scientific workflow executions in cloud environment. *IEEE Transactions on Cloud Computing, 4*(2), 166-179. https://doi.org/10.1109/tcc.2015.2453966.

Xu, X., Dou, W., Zhang, X., Hu, C., & Chen, J. (2017). A traffic hotline discovery method over cloud of things using big taxi GPS data. *Software: Practice and Experience, 47*(3), 361-377. https://doi.org/10.1002/spe.2412.

Xu, X., Fu, S., Cai, Q., Tian, W., Liu, W., Dou, W., Sun, X., & Liu, A.X. (2018). Dynamic resource allocation for load balancing in fog environment. *Wireless Communications and Mobile Computing, 2018*, 1-15. https://doi.org/10.1155/2018/6421607.

Xu, X., Zhang, X., Khan, M., Dou, W., Xue, S., & Yu, S. (2020). A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems. *Future Generation Computer Systems, 105*, 789-799. https://doi.org/10.1016/j.future.2017.08.057.

Xu, X., Zhao, X., Ruan, F., Zhang, J., Tian, W., Dou, W., & Liu, A.X. (2017). Data placement for privacy-aware applications over big data in hybrid clouds. *Security and Communication Networks, 2017*, 1-15. https://doi.org/10.1155/2017/2376484.

Yi, S., Li, C., & Li, Q. (2015, June). A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 Workshop on Mobile Big Data* (pp. 37-42). https://doi.org/10.1145/2757384.2757397.

Yin, B., Shen, W., Cheng, Y., Cai, L.X., & Li, Q. (2017, May). Distributed resource sharing in fog-assisted big data streaming. In *2017 IEEE International Conference on Communications* (ICC) (pp. 1-6). IEEE. https://doi.org/10.1109/ icc.2017.7996724.

Yu, L., Chen, L., Cai, Z., Shen, H., Liang, Y., & Pan, Y. (2016). Stochastic load balancing for virtual resource management in datacenters. *IEEE Transactions on Cloud Computing, 8*(2), 459-472. https://doi.org/10.1109/TCC.2016.2525984.

Zheng, L. (2014). Virtual machine resource allocation algorithm in cloud environment. *Computer Modelling & New Technologies, 18*(11), 279-284.