International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (Mar-24)

http://dx.doi.org/10.12785/ijcds/150179

Optimizing Fog Computing Efficiency: Exploring the Role of Heterogeneity in Resource Allocation and Task Scheduling

Nikita Sehgal[1], Savina Bansal[2] and RK Bansal[3]

[1,2,3]*Department of Electronics & Communication Engineering*
[1,2,3]*Giani Zail Singh Campus College of Engineering & Technology, Maharaja Ranjit Singh Punjab Technical University, Bathinda-151001 (Punjab), India*

E-mail address: [1]nikita.ece@mrsptu.ac.in; [2]savina.bansal@gmail.com; [3]drrakeshkbansal@gmail.com

Received 13 Sep. 2023, Revised 15 Jan. 2024, Accepted 7 Feb. 2024, Published 1 Mar. 2024

Abstract: Fog computing is a promising solution for latency-sensitive applications in the Internet of Things (IoT) era. This research thoroughly investigates the performance characteristics of fog computing systems, specifically focusing on the impact of architectural heterogeneity. The study explores how architectural diversity, deadline constraints, and the count of Mobile Data Centers (MDCs) influence key performance metrics. Through experimental simulations, the research assesses success ratio, rejection ratio, and resource utilization across various architectural models. The findings emphasize the significance of adopting heterogeneous architectures and wider deadline ranges, leading to improved success ratio and reduced job rejection. Moreover, increasing the number of MDCs positively affects resource utilization and overall system performance. This research offers valuable insights for optimizing fog computing systems, enabling the development of efficient solutions for real-world applications.

Keywords: Internet of Things, fog computing, scheduling, resource utilization

1. INTRODUCTION

The rapid increase of Internet of Things (IoT) devices has led to an exponential increase in data generation and processing requirements. However, traditional cloud computing architectures face numerous challenges in effectively addressing the unique demands of IoT applications, including latency issues, limited bandwidth, and concerns regarding data privacy and security [3, 4, 12, 20]. To overcome these challenges, fog computing has emerged as a promising paradigm that brings computation, storage, and networking resources closer to the edge of the network. Fog computing extends the capabilities of cloud computing by deploying a distributed computing infrastructure at the network edge [5, 9, 21]. This proximity to the edge allows fog computing to provide low-latency and high-bandwidth services to IoT devices, making it an ideal solution for real-time and latency-sensitive applications. Cloud data centers (CDCs) have emerged as a dominant execution mode for processing the data generated by IoT devices. However, the latency involved in transmitting data from IoT devices to CDCs can cause real-time applications to miss their processing deadlines. To overcome this latency challenge, edge computing has been introduced, where computation is performed as close to the source as possible, reducing the need for data transmission to the cloud. Mobile data centers (MDCs), also known as cloudlets, are edge devices that can execute jobs that would have otherwise been scheduled to run at a CDC [1]. By leveraging MDCs, latency-sensitive real-time applications, can meet their processing requirements without the significant network latency to the CDC.

Efficient resource utilization and task scheduling are critical components of fog computing systems [7, 13, 15, 16]. With a large number ofIoT devices and tasks to manage, an intelligent and deadline-aware task scheduling mechanism becomes essential to optimize resource usage and improve system performance. Furthermore, considering the limited resources of edge devices and the importance of efficient resource utilization, fog computing systems must prioritize scheduling IoT tasks in a manner that maximizes resource efficiency [17,19,28]. Deadline-aware task scheduling ensures that tasks are executed within their specified time constraints, which is particularly crucial for time-sensitive IoT applications such as real-time monitoring, autonomous systems, and smart grids [8,23]. Utilizing smart algorithms and

optimization techniques, a fog computing system allocates tasks to suitable computing resources, ensuring timely execution and meeting stringent time requirements. Efficient resource utilization is closely intertwined with deadline-aware task scheduling in fog computing systems [24,25]. By optimizing computing resource allocation, load balancing, and task assignment strategies, the system can effectively utilize available resources, avoiding bottlenecks and underutilization [11,14,32,35]. This not only enhances overall system performance but also reduces operational costs by minimizing the need for additional resources and improving the scalability of the IoT infrastructure.

Fog computing presents a set of practical considerations and challenges that significantly influence its deployment in real-world scenarios. In terms of deployment considerations, optimizing the network topology is paramount to ensure efficient data processing and minimal latency [6,7,13,21]. Proper resource allocation among fog nodes, considering the diverse capabilities of devices, plays a crucial role in enhancing overall system performance. Effective data management strategies, encompassing storage and transfer, are essential to handle the substantial volume of information generated by IoT devices. Additionally, implementing robust security measures at the edge is critical to safeguard sensitive data and maintain the integrity of fog computing systems[1, 4,36].

Conversely, fog computing deployment also faces various challenges that need careful navigation. Interoperability issues must be addressed to seamlessly integrate fog computing into existing infrastructures, enabling communication across diverse devices and platforms. Scalability is a persistent challenge, requiring adaptive measures to meet the escalating demands for computational resources as the number of IoT devices grows [5, 28]. Privacy concerns emerge with the decentralized processing of data at the edge, necessitating the implementation of privacy-preserving mechanisms to comply with regulations and establish user trust. Furthermore, ensuring reliability and fault tolerance in dynamic and potentially unstable edge environments is crucial for sustained performance [29,30]. Lastly, optimizing power consumption becomes imperative for fog computing deployments in remote or resource-constrained areas, emphasizing the need for energy-efficient solutions. Successfully navigating these considerations and challenges is pivotal for leveraging the full potential of fog computing in enhancing latency-sensitive applications and optimizing resource utilization at the network's edge [20,27].

In conclusion, fog computing and the integration of MDCs and CDCs provide a powerful framework for addressing the challenges of IoT applications [1]. By bringing computation closer to the edge, fog computing enables low-latency and high-bandwidth processing, while efficient resource utilization and deadline-aware task scheduling optimize system performance and meet the real-time requirements of IoT applications. These advancements pave the way for the effective deployment of IoT technologies in various domains, including smart cities, healthcare, transportation, and industrial automation [10,30,37].

The structure of the paper is as follows: the next section describes the latest related works in the area of fog computing. Detailed description of the proposed and implemented scheduling approach is presented in Section 3. The performance metrics and simulation setup used for experimentation are explained in Section 4 and 5 respectively. Section 6 presents the exhaustive performance analysis of the Homogeneous Architecture, Fixed Heterogeneity Model (FHM), and Mixed Heterogeneity Model (MHM). Finally, the paper completes with an overall conclusion in Section 7.

2. LITERATURE SURVEY

In recent years, there has been significant research conducted in the field of fog computing, addressing various aspects such as security, privacy, energy efficiency, resource management, and task scheduling. This literature review aims to provide an overview of the key studies conducted by researchers in these areas, highlighting their contributions and findings. Yousefpour et al. [34] proposed a fog computing approach aimed at minimizing delays in cloud service provisioning. They introduced a fog-enabled architecture that utilized local fog nodes to host and deliver cloud services closer to end-users. The authors discussed the advantages of reduced service latency and network congestion, emphasizing the importance of fog computing in improving the performance of cloud services. Oma et al. [22] proposed a tree-based fog computing (TBFC) model for efficient distribution of processes and data in IoT environments. The aim was to minimize the total electric energy consumption of nodes in the IoT. Through evaluations, the authors concluded that the TBFC model outperformed traditional cloud computing models by reducing energy consumption.

Toor et al. [2] presented an adaptive performance and energy-aware scheme for Fog-IoT computational environments utilizing iFogSim. The experimental results demonstrated enhancements in energy consumption, particularly in the power saver mode, compared to the existing approach. Deng et al. [37] proposed an improved Cuckoo Search algorithm incorporating various modifications for task scheduling on heterogeneous multiprocessor systems. The algorithm addressed task scheduling using Dynamic Voltage and Frequency Scaling (DVFS) to achieve better scheduling performance. This study demonstrated the effectiveness of the improved algorithm in optimizing task scheduling in fog computing environments. Bansal et al. [26] explored

dynamic voltage scaling (DVS) and dynamic power management (DPM) techniques for energy management in fog computing. They proposed preference-oriented energy-aware rate-monotonic scheduling (PER) and preference-oriented extended energy-aware rate-monotonic scheduling (PEER) algorithms, which outperformed several related studies in terms of energy savings. This research provided valuable insights into energy management techniques for fog computing systems.

Singh et al. [1] proposed RT-SANE, a solution that enabled batch and interactive applications to run while considering deadlines and safety requirements. Their approach selected between different computing platforms based on network latency and security tags, prioritizing speed and reliability. However, energy-efficient task scheduling was not prioritized in their work. Zhang et al. [36] presented a comprehensive review of fog computing, focusing on its vision, architecture, and challenges. They discussed the benefits of fog computing compared to traditional cloud computing and highlighted the key design principles and architectural components of fog computing systems. The paper also addressed challenges related to resource management, security, and scalability in fog computing environments. Rizwan et al. [14] conducted an experimental evaluation of machine learning algorithms for resource management in fog computing. They compared different algorithms, including decision trees, random forests, and support vector machines, to optimize resource allocation and task scheduling. The experimental results demonstrated the effectiveness of machine learning algorithms in improving resource utilization and overall system performance. Zhang et al. [33] investigated IoT task offloading in fog computing environments. They proposed a dynamic task offloading algorithm based on multi-objective optimization to achieve efficient resource utilization and reduce latency. Experimental results validated the effectiveness of the algorithm in improving system performance.

Zhou et al. [30] focused on fog-to-cloud offloading in the context of mobile edge computing (MEC). They proposed a joint optimization approach that considered both energy consumption and latency in the decision-making process for offloading tasks from fog to cloud. Experimental evaluations showed the advantages of the proposed approach in achieving energy-efficient and low-latency fog-to-cloud offloading. Azizi et al. [23] conducted a recent study on energy-efficient scheduling in a fog environment. They introduced two semi-greedy approaches to minimize penalty for deadline violations of independent real-time jobs on a heterogeneous fog platform. However, their study did not include the cloud platform in the analysis. Finally, Wang et al. [29] proposed a trust management framework for fog computing in the Industrial Internet of Things (IIoT). They designed a trust evaluation model and developed a trust management mechanism to enhance security and reliability in fog computing environments. Experimental results demonstrated the effectiveness of the framework in mitigating trust-related risks in IIoT applications. Sehgal et al. [20] conducted an extensive review of task and resource scheduling techniques within the fog computing context. The study aimed to identify contributions and limitations of different approaches in the fog computing environment, providing insights into their practical applicability. The authors delved into the details of task scheduling and resource allocation, offering valuable perspectives in the field of fog computing. This work serves as a foundational resource for comprehending current solutions and acts as a guide for the development of new and enhanced techniques. Atiq et al.[6] introduced a novel framework called Reliable Resource Allocation and Management (R2AM) to efficiently manage resource allocation in IoT transportation through fog computing. The strategy involves queuing data from IoT devices for storage and processing, alongside separate queues for fog nodes. Available fog nodes were prioritized based on their processing time, and IoT data is assigned to them in the specified order. Upon successful execution, the results were delivered to the users. This innovative approach ensured reliable and effective resource utilization in the context of IoT transportation and fog computing. In conclusion, these studies contribute to the growing body of knowledge in fog computing and identify future research directions for further advancements in the field.

3. PROPOSED WORK

The scheduling algorithm introduced takes into account network delay and security tags to determine the appropriate assignment of jobs. User-submitted jobs are classified into three categories: private, semi-private, and public, based on their security requirements. Private jobs are exclusively executed on the user's local Mobile Data Centers (MDCs) to ensure the preservation of a trusted environment. Semi-private jobs have the flexibility to be processed either on a MDC or a CDC, depending on specific requirements. Public jobs, which have lower security demands, are assigned to CDCs and remote MDCs. This approach strikes a balance between security and resource utilization. Private jobs receive the necessary security measures by being processed locally, while semi-private and public jobs can leverage the resources available in remote locations, optimizing the allocation of resources and overall system performance. To evaluate the effectiveness of the proposed scheduling algorithm, the code implementation initializes essential variables and parameters, including the number of jobs, MDCs, and CDCs, along with their respective capacities and communication delays. Job descriptions are generated, taking into account execution time (et), size, deadline, arrival time, and job types. The scheduling process follows a priority-based approach, assigning private jobs to MDCs, semi-private jobs to either MDCs or the CDC, and public jobs to the CDC or remote MDC meeting deadline and capacity requirement. The deadline and

capacity conditions for MDCs and CDCs in the fog computing system are expressed using "(1)" to "(8)". "(1)" calculates the time (A) when a job can start processing at the MDC, considering factors like communication delay, job size, bandwidth, MDC's available time, execution time, and MDC's capacity. Equation (2) represents the deadline of the job (B), a predefined time limit for completion.

A. MDC Deadline Condition:

$$A = cd\_mdc + ((job\_size)/(bw\_mdc)) + mdc\_rdy\_time + ((job\_et) / (mdc\_cap)) \quad (1)$$

$$B = job\_dl \quad (2)$$

The condition for scheduling a job is determined by comparing A with B as given in "(3)". If A is less than or equal to B, the job status is denoted as 1 (scheduled within the deadline), otherwise, it is 0 (not scheduled within the deadline).

$$if, A \leq B: Job\ status = 1 \quad (3)$$

Similarly, "(4)" calculates the time (C) when a job can start processing at the CDC, considering similar factors as for MDCs.

B. CDC Deadline Condition:

$$C = cd\_cdc + ((job\_size)/(bw\_cdc)) + \text{mdc\_rdy\_time} + ((\text{job\_et}) / (\text{cdc\_cap})) \quad (4)$$

Similar to the MDC Deadline Condition, the calculated start time of the job at the CDC (C) is compared with the deadline of the job (B) (refer "(5)") to determine if the job can be scheduled and completed within its deadline. If C is less than or equal to B, the job can be scheduled within its deadline, and its status is denoted as job status = 1

$$if, C \leq B: Job\ status = 1 \quad (5)$$

The capacity condition "(6)" calculates the required processing capacity for a specific job.

$$cap\_req = (job\_size) / (job\_et) \quad (6)$$

For both MDCs and CDCs, "(7)" and "(8)" ensure that the available processing capacity is sufficient to handle the job's requirements.

C. MDC Capacity Condition:

$$MDC\_cap \geq cap\_req \quad (7)$$

D. CDC Capacity Condition

$$CDC\_cap \geq cap\_req \quad (8)$$

By validating these capacity conditions, the fog computing system can effectively allocate jobs to the appropriate data centers based on their processing capabilities. Jobs are assigned to MDCs if their capacity is adequate for the job's processing needs. If no suitable MDC is available, the job may be allocated to the CDC or remote MDC, provided its capacity can handle the job. Adhering to these capacity conditions is vital for optimizing the performance and resource utilization of fog computing systems. It helps avoid overload situations and ensures that jobs are efficiently processed by the most suitable data center within the system. The research paper then conducts a comprehensive performance analysis to assess the efficiency of the system. Performance metrics, such as success ratio, rejection ratio and MDC and CDC utilization are calculated. Additionally, the analysis explores the impact of different job types and scheduling strategy on the overall system performance.

4. PERFORMANCE METRICS

Performance metrics play a crucial role in evaluating the efficiency and effectiveness of fog computing systems. These metrics provide valuable insights into the system's task scheduling capabilities and resource utilization. The key performance metrics used in this experimentation work are briefly defined as follows:

Success Ratio (SR): The success ratio measures the percentage of tasks successfully completed within their specified constraints. It provides an indication of the efficiency and effectiveness of task scheduling. It is defined as the ratio of successfully scheduled jobs to the total number of jobs as given in "(9)".

$$SR = \frac{n(jobs\_completed)}{n(jobs)} * 100 \quad (9)$$

Rejection Ratio (RR): The rejection ratio measures the proportion of rejected jobs to the total number of jobs as given by "(10)". It refers to the proportion or percentage of incoming tasks or requests that are not successfully serviced or completed within the system's specified constraints or deadlines. In other words, it represents the rate of tasks or jobs that are rejected or not accommodated due to various reasons, such as resource limitations, exceeded deadlines, or system overload.

$$RR = \frac{n(jobs\_rejected)}{n(jobs)} * 100 \quad (10)$$

Here, the "n(jobs_rejected)" refers to the count of jobs that were not successfully scheduled or processed, while the "n(jobs)" represents the overall number of jobs considered. By calculating the rejection ratio, the efficiency of a scheduling algorithm or system in terms of job acceptance and resource utilization can be accessed. A lower rejection ratio indicates a higher success rate in job scheduling, while a higher rejection ratio suggests that a significant number of jobs were unable to be accommodated or processed.

Resource Utilization (RU): Resource utilization measures the efficiency of resource usage in the system "(11)". It quantifies the extent to which resources, such as processing capacity or bandwidth, are effectively utilized. In the context of fog computing or task scheduling, resource utilization is often measured for individual resources, such as MDCs (Mobile Data Centers) or CDCs (Central Data Centers).

$$RU = \frac{Total\ Busy\ Time}{Total\ Time} * 100 \tag{11}$$

The total busy time for MDCs (Mobile Data Centers) can be calculated as follows "(12)":

$$Total_{mdc\_busy\_time} = \sum \frac{mdc_{used_{cap}}}{mdc_{cap}} \tag{12}$$

Similarly, the total busy time for the CDC (Central Data Center) can be calculated using "(13)" as:

$$Total_{cdc\_busy\_time} = \sum \frac{cdc_{used\_cap}}{cdc_{cap}} \tag{13}$$

This calculation represents the ratio of used capacity to total capacity for the CDC, indicating its busy time.

5. EXPERIMENTAL SET-UP

The experimental setup encompasses a range of scenarios to investigate the performance of a fog computing system, considering various factors such as the number of jobs, MDCs, CDCs, communication bandwidth, communication delay, deadline range, α factor, task size, and three levels of heterogeneity. The selection of these parameters was deliberate, aiming to ensure a thorough assessment of the system's performance and behavior. The number of jobs considered in the experiment ranged from 100 to 500, providing a diverse workload for the system under evaluation. The number of MDCs varied from 4 to 12, representing the fog computing nodes in the system. Additionally, there was one CDC in the setup. Communication bandwidth played a crucial role in the experiment, with the bandwidth from the user to the MDC set at 2000 Mbps and the bandwidth from the user to the CDC set at 4000 Mbps. These bandwidth values ensured efficient data transfer between the users and the fog computing nodes, as well as between the users and the central data center.

The communication delay from a source to an MDC was set at 5 ms, and to the CDC was set at 105 ms. Deadline range was another important factor considered in the experiment. The tasks had a fixed deadline range, with a loose deadline range of 10% to 50% of the job's execution time and a tight deadline range of 10% to 20% of the job's execution time. The tasks consisted of a mixed bag of small and larger-sized tasks chosen randomly. Three levels of MDC Capacity Heterogeneity are defined based on the processing capacities of the Mobile Data Centers (MDCs). At Level 1, MDCs have processing capacities ranging from 3000 to 4000 Million Instructions Per Second (MIPS). These MDCs offer moderate processing power, suitable for handling a variety of tasks and workloads. Level 2, MDCs have processing capacities ranging from 2500 to 4500 MIPS, providing a wider range of processing power. This level accommodates tasks with higher computational demands while still being capable of handling less intensive workloads. At Level 3, MDCs have processing capacities ranging from 2000 to 5000 MIPS. MDCs at this level exhibit a broader spectrum of processing power, enabling them to handle both low and high-computational tasks effectively. This heterogeneity in MDC capacities allows the fog computing system to efficiently allocate resources and distribute tasks based on the specific computational requirements of different applications and workloads. The Cloud Data Center (CDC) exhibits a processing capacity in the range of 60000 MIPS respectively.

Moreover, the experiments included three models to explore different architectural scenarios: the Homogeneous Architecture, the Fixed Heterogeneity (FHM) Architecture, and the Mixed Heterogeneity (MHM) Architecture. In the Homogeneous Architecture, all fog nodes had identical processing capacities within a specific range, resulting in a uniform distribution of computational resources. The Fixed Heterogeneity (FHM) Architecture introduced a level of heterogeneity by assigning a certain percentage of jobs to fog nodes with higher processing capacities, while the remaining jobs were allocated to fog nodes with lower capacities. The Mixed Heterogeneity (MHM) Architecture randomly assigned different processing capacities to fog nodes within the specified ranges, creating a diverse and dynamic computational environment.

All simulations in this study were implemented using MATLAB platform. The experimental setup was executed on a desktop computer equipped with an Intel(R) Core(TM) i7-6700 CPU running at 3.40GHz (boost frequency up to 3.41GHz), 8 GB of RAM, and operating on the Windows 10 operating system. To ensure statistical robustness and accuracy, each experiment was repeated five times with a different seed value for random number generation. This process of repeating the experiments with varying seed values helps account for any inherent randomness in the simulations. The average values obtained from these repeated runs were then used to report

the final results. The reference paper [1] served as a basis for the implementation, and modifications were made to incorporate the specific changes required for this study. By considering these parameters and architectural scenarios, the experimental study aimed to evaluate the system's behavior and performance under different workload, communication, and resource allocation conditions.

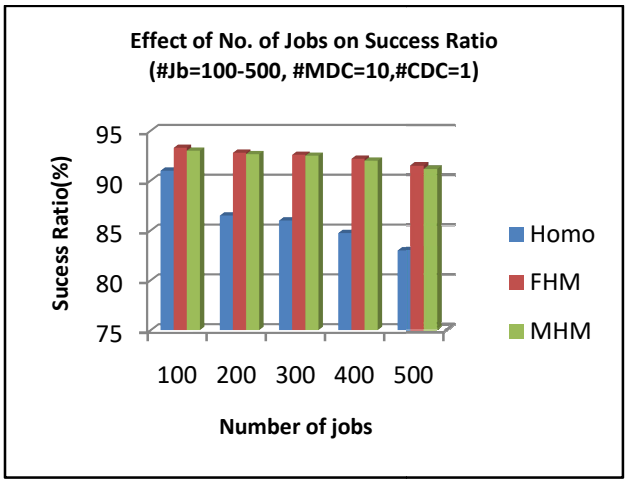
6. Results and Discussion

This section presents the observations derived from the analysis of various aspects of fog computing systems. The study focuses on investigating the impact of multiple variables, such as the number of jobs, deadline ranges, heterogeneity levels, and the number of Mobile Data Centers (MDCs), on the performance and resource utilization of these systems. By examining metrics like the success ratio, rejection ratio, and resource utilization across different architectures and scenarios, valuable insights are obtained. The results shed light on the operational effectiveness of fog computing systems and emphasize the advantages of incorporating heterogeneity into these systems.

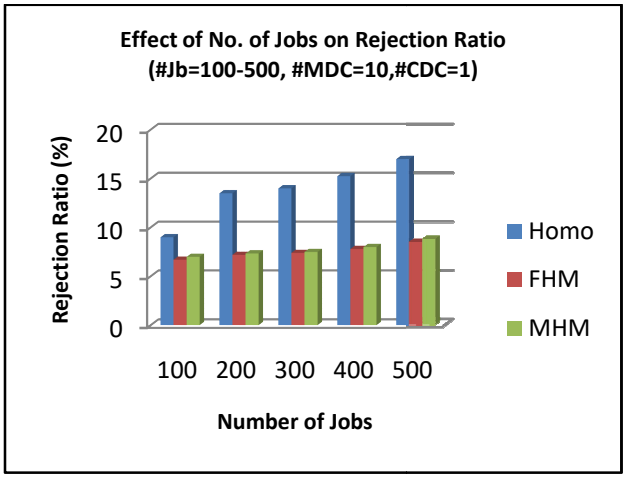
The experimental work includes 400 jobs, 10 MDCs, and 1CDC. The analysis takes into account a moderate deadline range of 10% to 40% of the job's execution time. Homogeneous Architecture ensures uniform processing capacities among all MDCs. The Fixed Heterogeneity Model (FHM) includes 30% fast MDCs and the rest as low-speed MDCs. In the Mixed Heterogeneity Model (MHM), different processing capacities are randomly distributed among MDCs, resulting in a constantly changing and diverse computational environment. This architectural model enables researchers to assess how resource heterogeneity impacts system performance.

A. Job Scaling

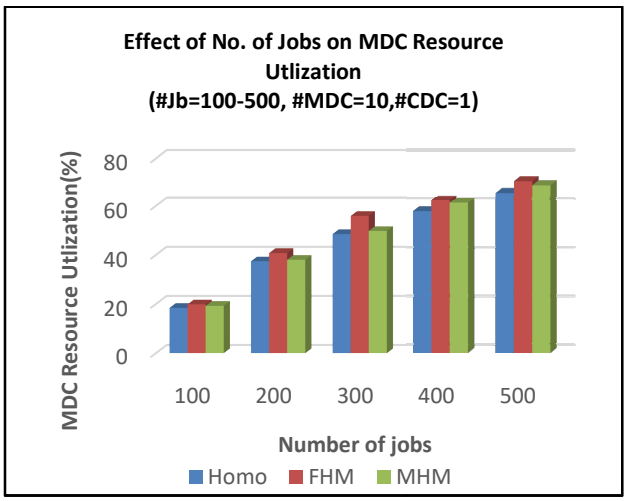
The results in "Fig.1" primarily revolve around varying the workload volume while maintaining a moderate deadline range of 10% to 40% of the processing time for each job. The configuration involves 10 MDCs, 1CDC at heterogeneity level 1.Examining the success ratio, it is evident that the Homogeneous model initially achieves a relatively high success ratio of 91% for 100 jobs. However, as the number of jobs increases, the success ratio gradually declines. In contrast, both the Fixed Heterogeneity Model (FHM) and Mixed Heterogeneity Model (MHM) consistently exhibit higher success ratios. The FHM Architecture experiences a slight decline from 93.3% to 91.5% with increasing job numbers, while the MHM Architecture follows a similar trend, starting at 93% and slightly decreasing.



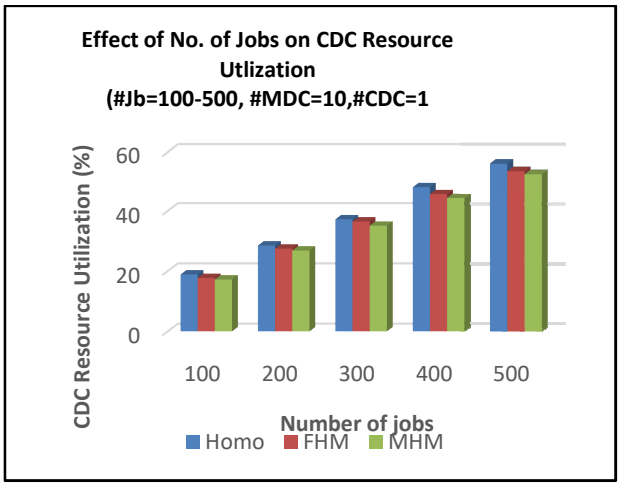
(a)



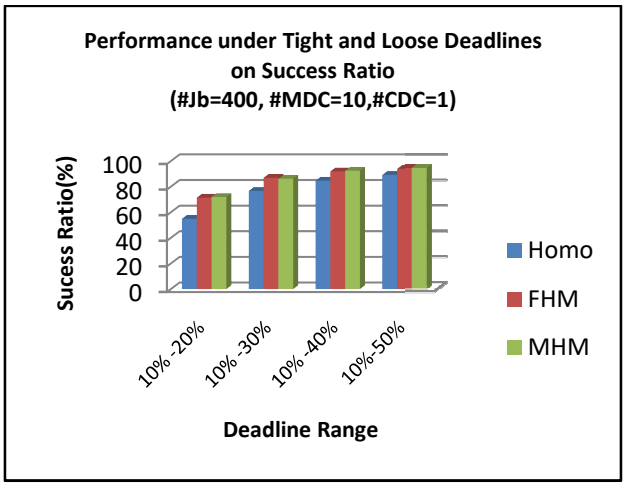
(b)



(c)

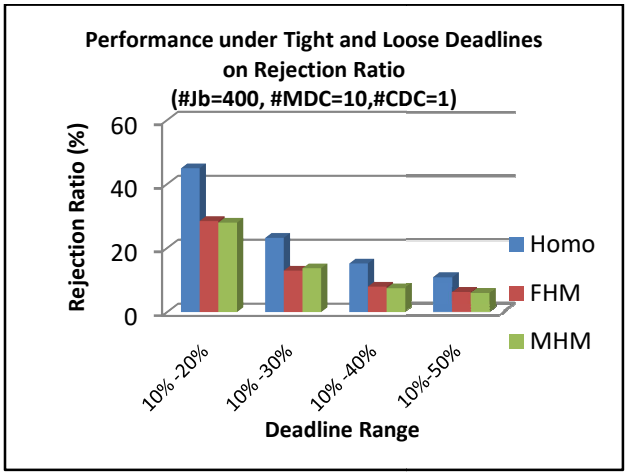


(d)

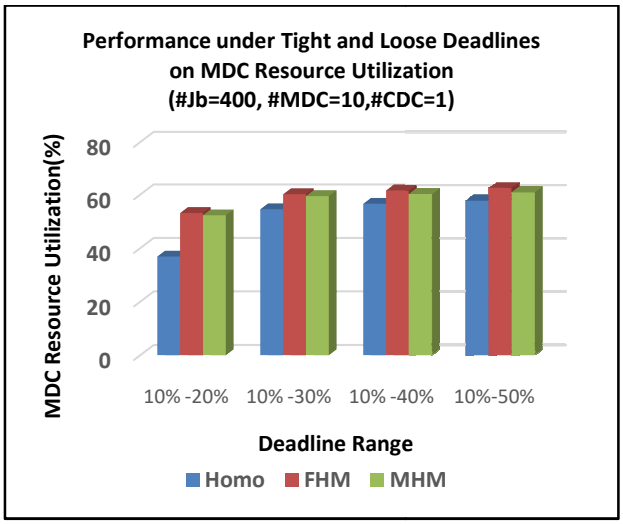


(a)

Figure 1. Impact of Increasing Numbers of Jobs on (a)Success Ratio, (b)Rejection Ratio, (c) MDC Resource Utilization and (d) CDC Resource Utilization in Homogeneous and Heterogeneous Fog Computing Systems



(b)

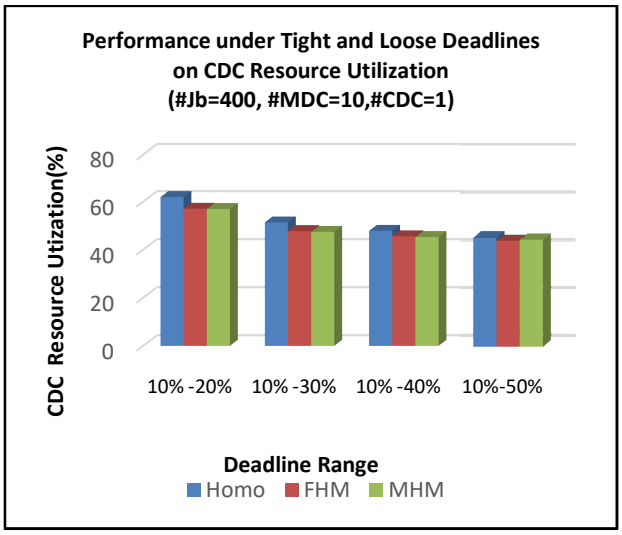


(c)

Analyzing the rejection ratio, the Homogeneous Model stands out with the highest rejection ratio, indicating a larger number of job rejections. Conversely, the FHM and MHM architectures demonstrate lower rejection ratios, suggesting more efficient job allocation and scheduling approach. Both MDC and CDC resource utilization increase as the number of jobs increases across all architectures. In the Homogeneous Architecture, resource utilization steadily rises for both MDCs and CDCs, reaching levels of up to 65.77% and 55.85%, respectively. Similarly, the FHM and MHM architectures shows increase in resource utilization in both MDCs and CDCs, with the FHM Architecture showcasing the highest utilization rates among the three architectures. The FHM and MHM architectures consistently outperform the Homogeneous Architecture in terms of success ratio, job rejection ratio and resource utilization when number of jobs is increased indicating a need for efficient resource allocation and management.

B. Varied Deadline Constraints

"Fig. 2" presents significant findings regarding the performance attributes of fog computing systems under different deadline constraints. The experiment, using a dataset with 400 jobs, 10 MDCs, 1 CDC, and heterogeneity level 1, investigates the effects of wider (10% to 50%) and narrower (10% to 20%) deadline ranges on the system's performance metrics.



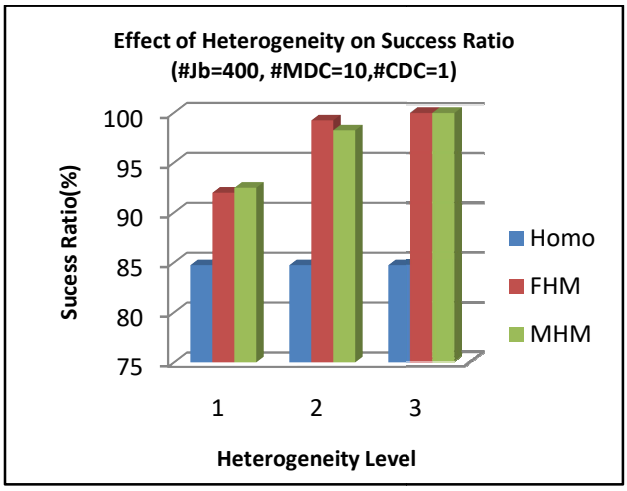
(d)

Figure 2. Performance under Tight and Loose Deadlines on (a)Success Ratio, (b)Rejection Ratio, (c) MDC Resource Utilization and (d) CDC Resource Utilization in Homogeneous and Heterogeneous Fog Computing Systems

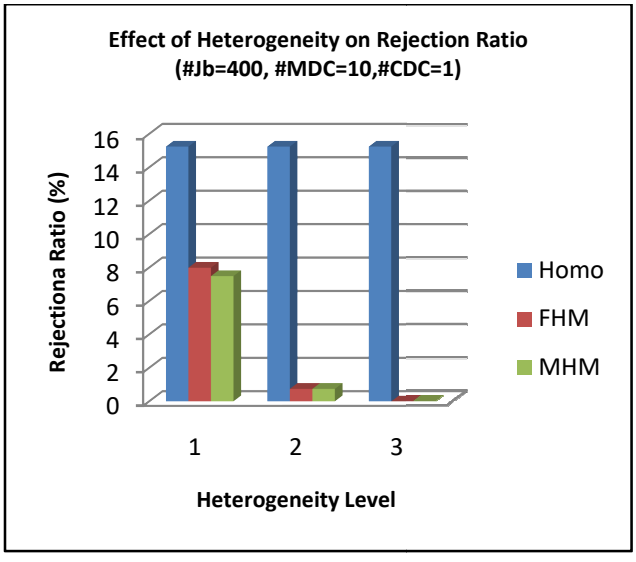
The empirical analysis in "Fig. 2" demonstrates that extending the deadline range to 10% to 50% yields substantial performance improvements across all architectural models when compared to a narrower range of 10% to 20%. Specifically, the Homogeneous Architecture exhibits significantly higher success ratios within the broader deadline range (84.75% to 89.25%), while the narrower range results in a diminished success ratio of 55%. Similarly, the Fixed Heterogeneity (FHM) Architecture and Mixed Heterogeneity (MHM) Architecture showcase enhanced success ratios within the wider deadline range (FHM: 92% to 94%, MHM: 92.5% to 94.25%).Furthermore, a broader deadline range is associated with reduced rejections across all architectural models. The Homogeneous Architecture experiences a substantial decrease in the rejection ratio (from 15.25% to 10.75%) within the wider deadline range, while the narrower range leads to a higher rejection ratio of 45%. Similarly, the FHM and MHM architectures maintain lower rejection ratios within the broader deadline range (FHM: 8% to 6%, MHM: 7.5% to 5.75%).Moreover, both MDCs and CDCs demonstrate improved resource utilization within the broader deadline range. The Homogeneous Architecture achieves higher resource utilization levels (MDCs: up to 57.98%, CDCs: up to 47.93%) within the wider deadline range, whereas the narrower range results in reduced utilization rates. Similarly, the FHM and MHM architectures exhibit enhanced resource utilization in both MDCs and CDCs when subject to the wider deadline range. In conclusion, the adoption of a broader deadline range (10% to 50%) optimizes performance metrics in fog computing systems, leading to higher success ratios, lower rejection ratios, and improved resource utilization. This facilitates more efficient job allocation and ultimately enhances the overall system performance compared to a narrower deadline range.

C. *Impact of Heterogeneity*

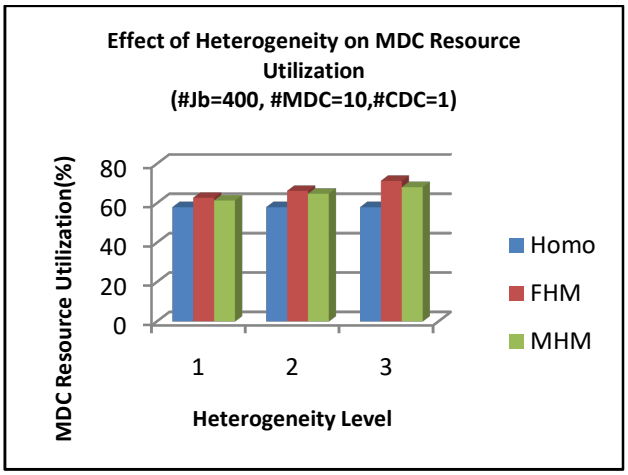
The gathered data provides significant results regarding the impact of heterogeneity on the success ratio, rejection ratio, and resource utilization in fog computing systems with varying MDC capacities. The analysis encompasses three MDC capacity ranges: Heterogeneity Level 1(3000-4000 MIPS), Level 2 (2500-4500 MIPS), and Level 3 (2000-5000MIPS). Within the Homogeneous Architecture, the MDC capacity is determined by the average value within each range.



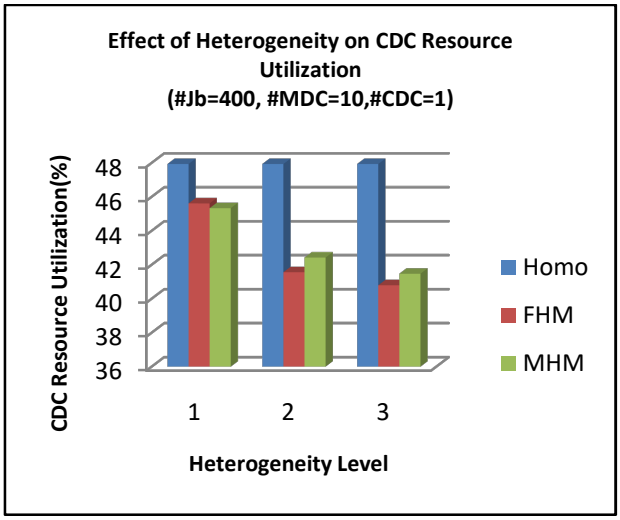
(a)



(b)



(c)



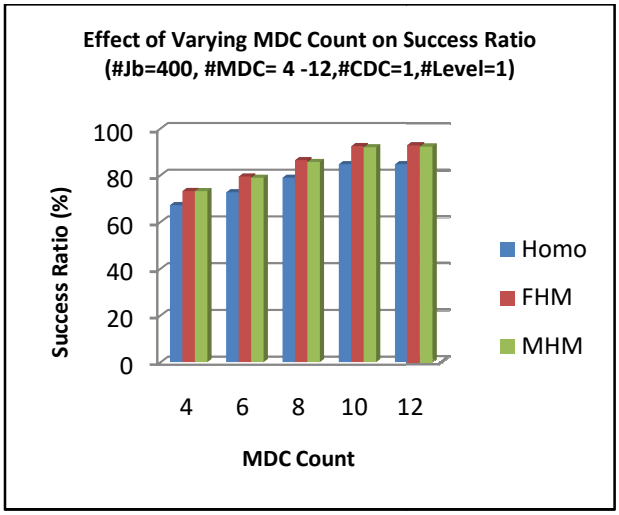
(d)

Figure 3. Impact of Heterogeneity on (a)Success Ratio, (b)Rejection Ratio, (c) MDC Resource Utilization and (d) CDC Resource Utilization in Homogeneous and Heterogeneous Fog Computing Systems

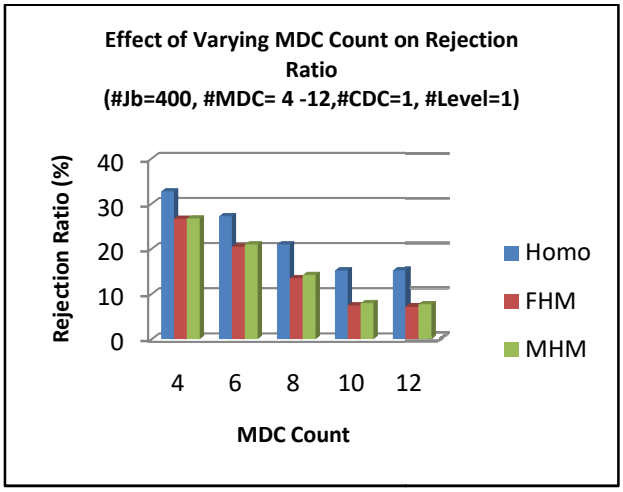
As depicted in "Fig. 3", the success ratio varies significantly across different architectural models and heterogeneity levels in fog computing systems. The Homogeneous Model maintains a consistent success ratio of 84.75% across all capacity ranges, while FHM and MHM achieve notably higher success ratios, even reaching 100% in certain capacity ranges. Comparatively, the Homogeneous Model exhibits a constant rejection ratio of 15.25%, while both FHM and MHM achieve an impressively low rejection ratio of 0%. Moreover, the introduction of heterogeneity in fog computing systems alleviates the stress on the CDC as the major workload is effectively handled by the MDCs. The utilization of heterogeneous architectures allows for more balanced resource distribution, resulting in reduced resource utilization in the CDC and enhanced performance across the fog computing system. Overall, incorporating heterogeneity in fog computing systems yields higher success ratios, reduced rejection ratios, and improved resource utilization, particularly within MDCs. These findings underscore the significance of considering heterogeneous architectures and optimizing heterogeneity levels to enhance system performance and resource utilization in fog computing environments.

D. MDC Count Impact on Fog Computing (Heterogeneity Level 1)

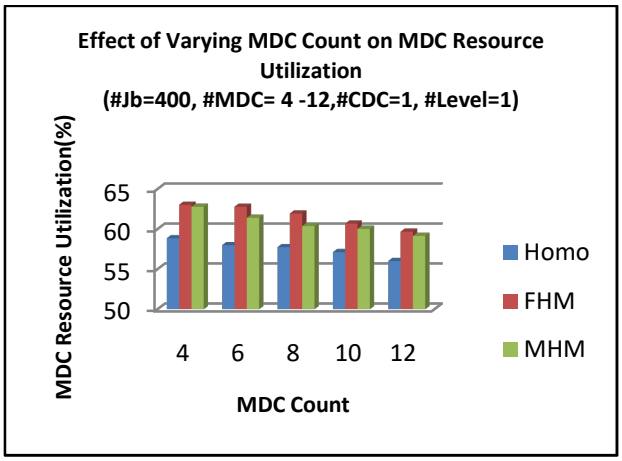
The experiments conducted using a dataset comprising 400 jobs evaluates the system's performance metrics with varying numbers of MDCs under a moderate deadline range (10% to 40% of job execution time) at heterogeneity level 1.



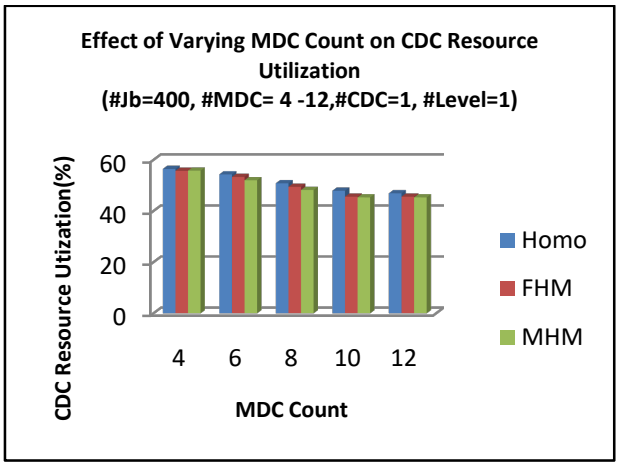
(a)



(b)



(c)



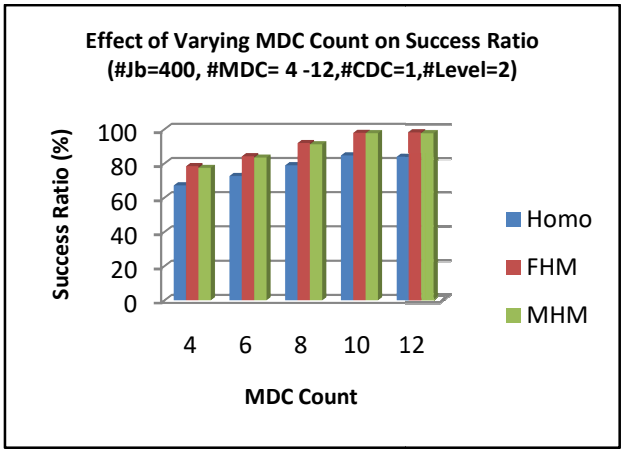
(d)

Figure 4. Impact of Varying MDC Count on (a)Success Ratio, (b) Rejection Ratio, (c) MDC Resource Utilization and (d) CDC Resource Utilization in Homogeneous and Heterogeneous Fog Computing Systems at Heterogeneity Level 1

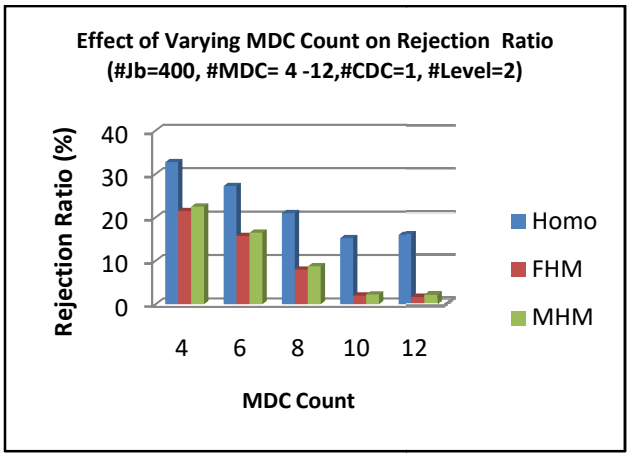
As depicted in "Fig. 4", the Homogeneous Architecture displays a consistent success ratio ranging from 67.25% to 84.75% as the number of MDCs increases. In contrast, both FHM and MHM exhibit increasing success ratios with higher MDC counts, achieving up to 93% for FHM and 92.5% for MHM. This demonstrates that incorporating heterogeneity in the form of additional MDCs improves the overall success ratio of fog computing systems. The rejection ratio in the Homogeneous Architecture decreases from 32.75% to 15.25% as the number of MDCs increases. In contrast, FHM and MHM exhibit significantly lower rejection ratios, ranging from 26.7% to 7% for FHM and 26.75% to 7.5% for MHM in the highest MDC count. The Homogeneous Architecture experiences a slight decrease in MDC resource utilization, ranging from 58.84% to 56%, with increasing MDCs. Similarly, FHM and MHM show declining trends in MDC resource utilization, ranging from 63.01% to 59.66% for FHM and 62.77% to 59.13% for MHM. This implies that the introduction of more MDCs results in a slightly lower MDC resource utilization. In case of CDC resource utilization, the Homogeneous Architecture exhibits a gradual decrease from 56.43% to 46.93% with an increasing number of MDCs. Similarly, both FHM and MHM display decreasing resource utilization in the CDCs, ranging from 55.71% to 45.61% for FHM and 55.79% to 45.33% for MHM. This suggests that as the number of MDCs increases, the load on the CDCs decreases, leading to reduced CDC resource utilization. Introducing heterogeneity through additional MDCs results in improved success ratios, reduced rejection ratios, and a more balanced utilization across the system.

E. MDC Count Impact on Fog Computing (Heterogeneity resource Level 2)

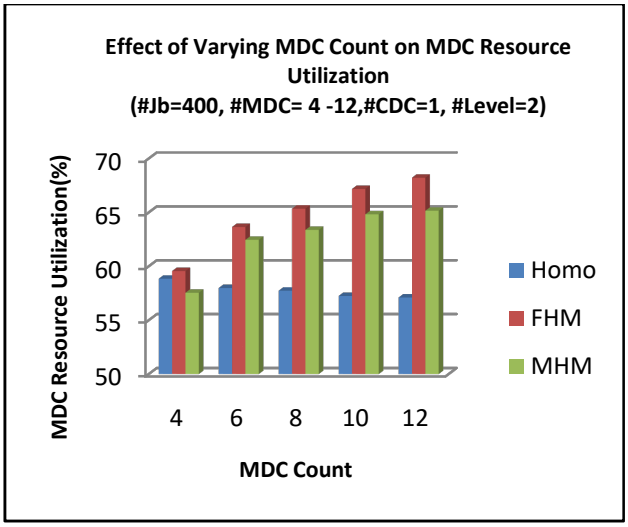
In the first scenario("Fig. 4")with heterogeneity level 1 both the Homogeneous and Heterogeneous architectures showed improvements in success ratio, reduced task violations, and slightly decreased resource utilization in MDCs and the CDC as the number of MDCs increased.



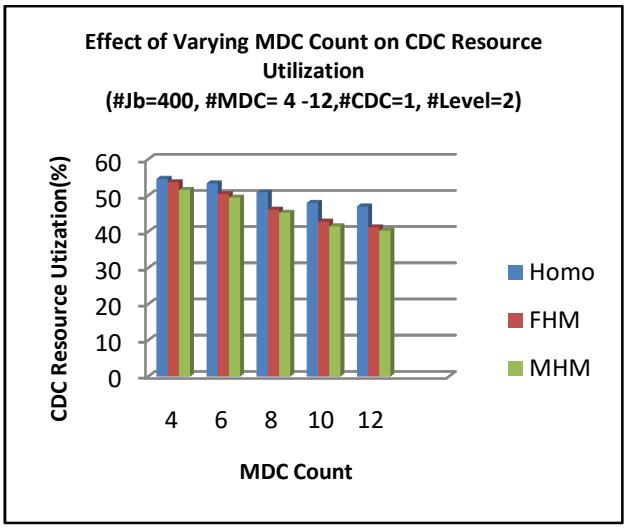
(a)



(b)



(c)



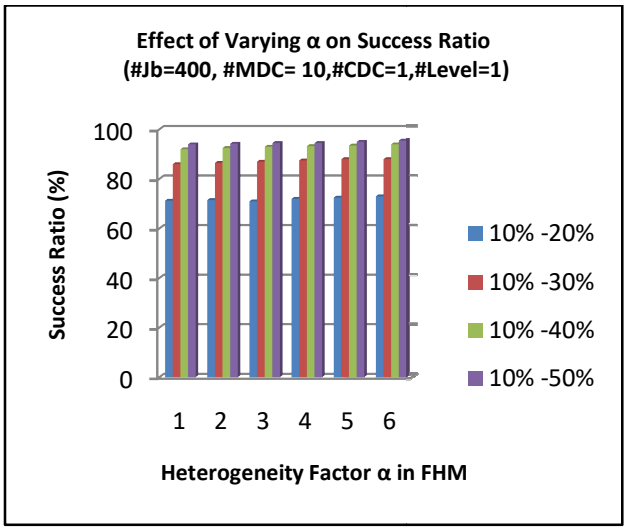
(d)

Figure 5. Impact of Varying MDC Count on (a) Success Ratio, (b)Rejection Ratio, (c) MDC Resource Utilization and (d) CDC Resource Utilization in Homogeneous and Heterogeneous Fog Computing Systems at Heterogeneity Level 2

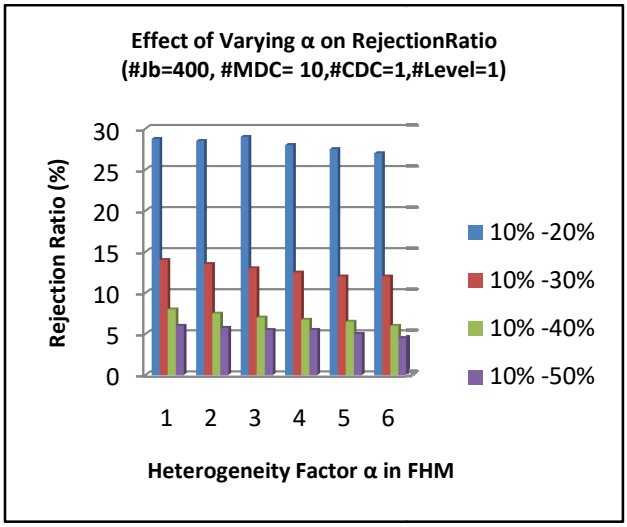
In the second scenario ("Fig. 5"), where MDC heterogeneity level 2 was introduced, a notable impact on the system's performance has been observed. Increased heterogeneity led to higher success ratios, lower rejection ratios, and improved resource utilization compared to the previous scenario. The research findings conclusively showcase the benefits of integrating heterogeneity into fog computing systems. Level 3 of heterogeneity, with MDCs having processing capacities from 2000 to 5000 MIPS, demonstrates exceptional performance with a 100% success ratio and 0% rejection ratio, even with a low count of MDCs. These MDCs efficiently handle diverse workloads, ensuring timely completion of all tasks. Also the resource utilization is optimized. Overall, this heterogeneous resource utilization maximizes system efficiency and performance, making it ideal for fog computing systems with dynamic workload variations.

F. Influence of α on Fog Computing (Heterogeneity Level 1)

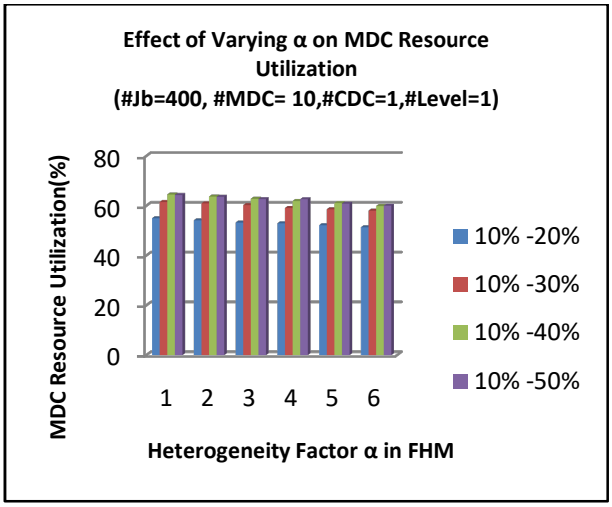
As depicted in "Fig. 6", the data analysis focuses on the impact of varying α values (ranging from 1 to 6) on fog computing systems. Heterogeneity factor α represents the percentage of fast Mobile Data Centers (MDCs) in case of FHM. The study examines the effect of α value on success ratio, rejection ratio, and resource utilization under both loose and tight deadline scenarios. The analysis considers 400 jobs, 10 MDCs, 1 CDC and a heterogeneity level 1.



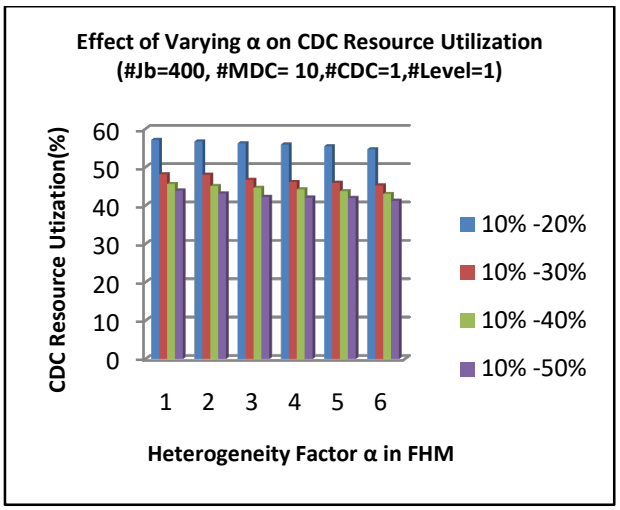
(a)



(b)



(c)



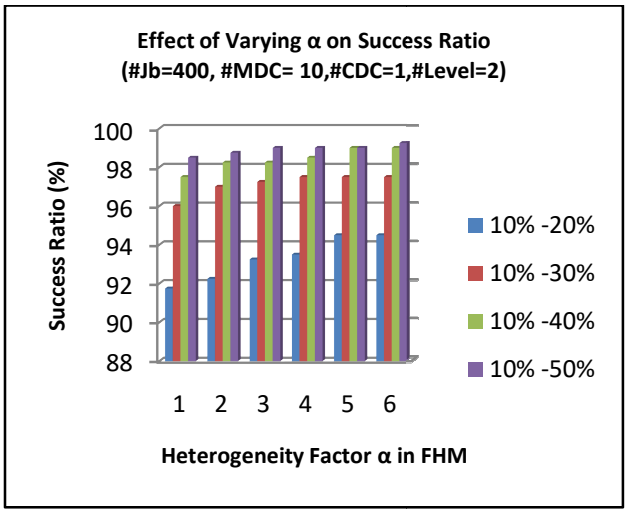
(d)

Figure 6. α influence on (a)Success Ratio, (b) Rejection Ratio, (c) MDC Resource Utilization and (d) CDC Resource Utilization in Homogeneous and Heterogeneous Fog Computing Systems at Heterogeneity Level 1

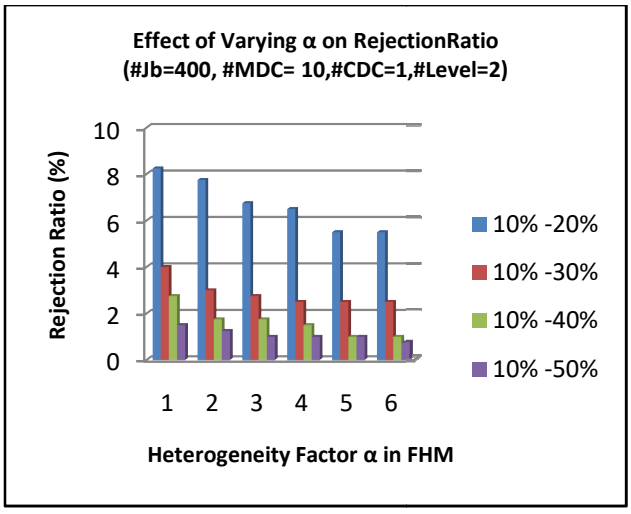
The results indicate that increasing α result in higher success ratios, irrespective of the deadline scenario. This suggests that a higher proportion of α leads to improved task success rates. Moreover, higher α value corresponds to lower rejection ratios in both loose and tight deadline scenarios. This implies that a greater availability of fast MDCs facilitates efficient task allocation and scheduling, reducing the number of rejected tasks. Generally, higher α value contributes to improved resource utilization across different deadline scenarios. These insights contribute to the ongoing efforts in enhancing the overall performance and resource efficiency of fog computing systems.

G. *Influence of α on Fog Computing (Heterogeneity Level 2)*

In the context of heterogeneity level 1(“Fig. 6”), the analysis considering both loose and tight deadlines yields intriguing results. When compared with heterogeneity level 2 as depicted in “Fig. 7”, heterogeneity level 2 consistently outperforms the 3000-4000 MIPS heterogeneity case for both loose and tight deadlines. The higher success ratios in the heterogeneity level 2 scenario indicate enhanced task completion rates.



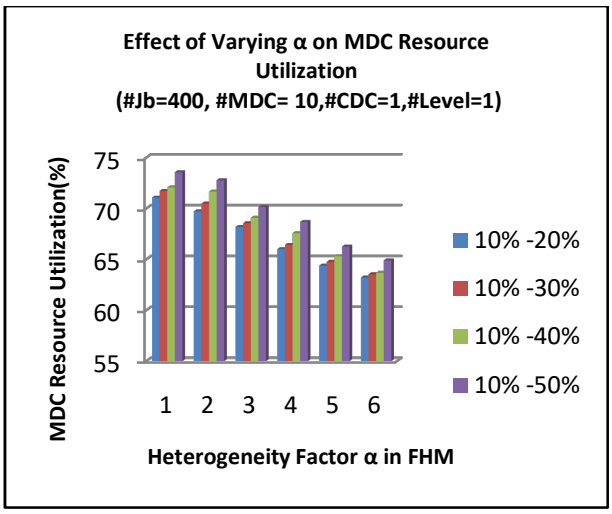
(a)



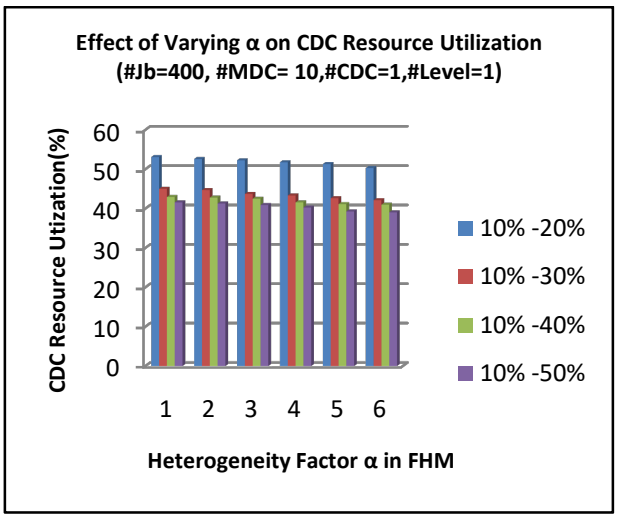
(b)

Upon analyzing the rejection ratios in both heterogeneity level 1 and 2 scenarios, a decreasing trend as the α values increase, indicating improved task allocation strategies. However, heterogeneity level 2

consistently exhibits lower rejection ratios compared to heterogeneity level 1 for both loose and tight deadlines.



(c)



(d)

Figure 7. α influence on (a)Success Ratio, (b) Rejection Ratio, (c) MDC Resource Utilization and (d) CDC Resource Utilization in Homogeneous and Heterogeneous Fog Computing Systems at Heterogeneity Level 2

In terms of resource utilization, the heterogeneity level 2 scenario with 2500-4500 MIPS demonstrates superior resource utilization in both MDCs and CDC compared to the heterogeneity level 1 scenario with 3000-4000 MIPS. This suggests more efficient utilization of computational resources, leading to higher overall system performance. In conclusion, the analysis considering both loose and tight deadlines highlights that the heterogeneity level 2 scenario (as depicted in Figure 6.7) outperforms the heterogeneity level 1 scenario (as shown in Figure 6.6) in terms of success ratio, rejection ratio, and resource utilization.

7. Conclusions and Future Scope

Fog computing plays a crucial role in the IoT era, bringing computing resources closer to IoT devices for reduced latency. Our research thoroughly explores fog computing system performance, focusing on heterogeneity impact. The study examines varied deadline constraints, Mobile Data Center (MDC) count, and heterogeneity integration. The Fixed Heterogeneity Model (FHM) and Mixed Heterogeneity Model (MHM) show practical applications in smart cities, healthcare, and smart manufacturing. FHM aids in efficient traffic management, MHM dynamically allocates processing capacities for healthcare tasks, and both contribute to predictive maintenance in manufacturing. These models outperform the Homogeneous Architecture, showcasing higher success ratio, reduced rejection ratio, and optimized resource utilization. The study emphasizes the importance of heterogeneity and resource management in fog computing, offering insights for real-world applications. Future research avenues include exploring the integration of edge computing technologies and artificial intelligence to improve decision-making and resource management in fog computing environments for increased efficiency and performance.

References

[1] A. Singh, N. Auluck, O. Rana, A. Jones, and S. Nepal, "Scheduling Real-Time Security Aware tasks in Fog Networks," Journal of LATEX Class, vol. 14, no. 8, 2015. DOI: 10.1109/TSC.2019.2914649.

[2] A. Toor, S. ul Islam, N. Sohail, A. Akhunzada, J. Boudjadar, H. A. Khattak, I. U. Din, and J. J. P. C. Rodrigues, "Energy and Performance Aware Fog Computing: A Case of DVFS and Green Renewable Energy," Future Generation Computer Systems, vol. 101, pp. 1112-1121, 2019. DOI: 10.1016/j.future.2019.07.010.

[3] C. Zhang, S. Zhao, N. Xiong, and Y. Liu, "Experimental Study on IoT Task Offloading in Fog Computing," Journal of Network and Computer Applications, vol. 151,2020 102515.

[4] Fog computing and internet of things: Extend the cloud to where the things are. Cisco White Paper, 2015. www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing overview.pdf

[5] G. L. Stavrinides, and H. D. Karatza, "A hybrid approach to scheduling real-time IoT workflows in fog and cloud environments," Multimedia Tools and Applications, vol. 78, no. 17, pp. 24639-24655, 2019.

[6] H. U. Atiq, Z. Ahmad, S. K. uzZaman, M. A. Khan, A. A. Shaikh, and A. Al-Rasheed, "Reliable Resource Allocation and Management for IoT Transportation Using Fog Computing," Electronics, vol. 12, no. 6, pp. 1452, 2023. Available: https://doi.org/10.3390/electronics12061452.

[7] H. Zhou, J. Hu, Y. Li, and L. Zhang, "Fog-to-Cloud Offloading with Joint Optimization of Energy and Latency in Mobile Edge Computing," IEEE Transactions on Mobile Computing, vol. 20, no. 1, pp. 208-222, 2021.

[8] J. A. Stankovic, M. Spuri, K. Ramamritham, G.C. Buttazzo, "Deadline scheduling for real-time systems: EDF and related algorithms," Springer Science and Business Media, vol. 460, 2012.

[9] J. Xu, X. Sun, R. Zhang, H. Liang, and Q. Duan, "Fog-cloud task scheduling of energy consumption optimization with deadline consideration," International Journal of Internet Manufacturing Services, vol.7, no.4, pp. 375–392, 2020.

[10] J. Zhang, C. Ma, and J. Liu, "Fog Computing for Internet of Things: A Survey," Journal of Network and Computer Applications, vol. 168, 102765, 2020.

[11] L. Feng, Y. Wang, X. Liu, and Y. Zhou, "Artificial Intelligence Techniques for Resource Management in Fog Computing: A Comprehensive Survey," IEEE Communications Surveys and Tutorials, vol.22, no. 4, pp. 2422-2451, 2020.

[12] M. Kumar, S.C. Sharma, A. Goel, and S.P. Singh, "A comprehensive survey for scheduling techniques in cloud computing," Journal of Network and Computer Applications, vol. 143, pp. 1-33, 2019.

[13] M. Li, J. Yan, and A. V. Vasilakos, "Experimental Study on Energy Efficiency of Computation Offloading Algorithms in Fog Computing," Journal of Parallel and Distributed Computing, vol. 128, pp.1-9, 2019.

[14] M. Rizwan, A. Akram, U. Ghani, and N. Javaid, "Experimental Evaluation of Machine Learning Algorithms for Resource Management in Fog Computing," Future Generation Computer Systems, vol. 105, pp. 577-588, 2020.

[15] N. Kaur, S. Bansal, and R.K. Bansal, "Task scheduling and energy conservation techniques for multiprocessor computing systems," International Journal of Networks and Systems, vol.2, no. 2, pp. 5-8, 2015.

[16] N. Kaur, S. Bansal, and R.K. Bansal, "Towards energy efficient scheduling with DVFS for precedence constrained tasks on heterogeneous cluster system," 2nd IEEE International conference on Recent Advances in Engineering and Computational Sciences, pp. 1-6, 2015.

[17] N. Kaur, S. Bansal, and R.K. Bansal, "Energy efficient duplication-based scheduling for precedence constrained tasks on heterogeneous computing cluster," Multiagent and Grid Systems, vol. 12, no. 3, pp. 239-252, 2016.

[18] N. Kaur, S. Bansal, and R.K. Bansal, "Duplication-controlled static energy-efficient scheduling on multiprocessor computing system," Concurrency and Computation: Practice and Experience, vol. 29, no. 12, pp. e4124, 2017.

[19] N. Kaur, S. Bansal, and R.K. Bansal, "Survey on energy efficient scheduling techniques on cloud computing," Multiagent and Grid Systems, vol. 17, no. 4, pp. 351-366, 2021.

[20] N. Sehgal, S. Bansal, and R.K. Bansal, "Task scheduling in fog computing environment: An overview," International Journal of Engineering Technology and Management Sciences, vol. 7, no.1, pp. 47-54, 2023.

[21] R. K. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," Future Generation Computer Systems, vol. 25, no. 6, pp. 599-616, 2009. DOI: 10.1016/j.future.2008.12.001.

[22] R. Omaa, S. Nakamuraa, D. Duolikuna, T. Enokido, and M. Takizawac, "A Tree-Based Model of Energy-Efficient Fog Computing Systems in IoT," Complex, Intelligent, and Software Intensive Systems, vol. 12, pp. 14-26, 2018. DOI: 10.1016/j.iot.2018.08.003.

[23] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach," Journal of Network and Computer Applications, vol.201, 103333, 2022.

[24] S. Bansal, P. Kumar, and K. Singh, "An improved duplication strategy for scheduling precedence constrained graphs in multiprocessor systems," IEEE Transactions on Parallel and Distributed Systems, vol.14, no. 6, pp. 533-544, 2003.

[25] S. Bansal, P. Kumar, and K. Singh, "Dealing with heterogeneity through limited duplication for scheduling precedence constrained task graphs," Journal of Parallel and Distributed Computing, vol.65, no. 4, pp.479-491, 2005.

[26] S. Bansal, R.K. Bansal, and K. Arora, "Energy-cognizant scheduling for preference-oriented fixed-priority real-time tasks," Journal of Systems Architecture, vol.108, 101743, 2020.

[27] W. Jia, Y. Chen, H. Jiang and M. Li, "Fog-Based Data Storage and Processing for IoT: A Survey," IEEE Internet of Things Journal, vol.7, no. 12, pp. 12182-12201, 2020.

[28] W. Kang, H. Jin, C. Li, Q. Wang, and X. Li., "An Energy-Efficient Resource Allocation Approach in Fog Computing for Industrial Internet of Things," IEEE Internet of Things Journal, vol.7, no.2, pp.1270-1280, 2020.

[29] Y. Wang, F. Li, Y. Liu, and D. Zhao, "A Trust Management Framework for Fog Computing in Industrial Internet of Things," IEEE Transactions on Industrial Informatics, vol. 18, no.1, pp. 227-236, 2022.

[30] Y. Wang, Y. Zhou, L. Wang, and H. Zheng, "Secure Data Aggregation in Fog Computing: A Survey," IEEE Transactions on Industrial Informatics, vol.17, no.3, pp. 2017-2031, 2021.

[31] Y. Wu, J. Li, H. Ma, and W. Tian, "Experimental Study on QoS-Aware Task Offloading in Fog Computing," IEEE Transactions on Industrial Informatics, vol.17, no.3, pp. 2043-2053, 2021.

[32] Y. Xu, C. Xu, X. Lin, and Y. Li, "Dynamic Task Offloading and Resource Allocation for IoT Applications in Mobile Edge Computing," IEEE Transactions on Cloud Computing, vol.8, no. 4, pp. 1260-1273, 2020.

[33] Y. Zhang, S. Wen, Z. Wu, and L. Gao "Experimental Study of Energy Efficiency Optimization for Fog Computing," IEEE Internet of Things Journal, vol.9, no.2, pp. 1387-1397, 2022.

[34] Yousefpour, G. Ishigaki, J. P Jue and Y. Zhang, "Fog computing: Towards minimizing delays of cloud service provisioning," IEEE Transactions on Services Computing, vol.10, no.5, pp. 817-830, 2017.

[35] Z. Shi, Z. Zhou, and Q. Zhu, "An Energy-Efficient Resource Allocation Scheme for Fog Computing in Internet of Things," IEEE Transactions on Industrial Informatics, vol.15, no.6, pp. 3568-3576, 2019.

[36] Z. Zhang, C. Wang, Y. Chen, and Z. Xiao "Fog Computing: Vision, Architecture, and Challenges," IEEE Internet of Things Journal, vol.6, no. 5, pp. 7912-7924, 2019.

[37] Z. Deng, Z. Yan, H. Huang and H. Shen, "Energy-Aware Task Scheduling on Heterogeneous Computing Systems with Time Constraint," IEEE Access, vol.8, pp. 23936-23950, 2020.

AUTHOR'S BIOGRAPHY

Nikita Sehgal from India received both her Bachelors and Masters in Electronics & Communication Engineering from I.K.G. Punjab Technical University, Jalandhar and Punjabi University, Patiala, respectively. She is pursuing Ph.D. in ECE from Maharaja Ranjt Singh Punjab Technical University, Bathinda. Presently, she is working as Assistant Professor at Giani Zail Singh Campus College of Engineering & Technology, Maharaja Ranjit Singh Punjab Technical University, Bathinda, India since 2014. Her current areas of interest include fog computing, scheduling. She is available at nikis762@gmail.com.

Savina Bansal from India earned her Ph.D. from IIT Roorkee, Masters in Computers from TIET, Patiala, Bachelors in Engg. from PEC, Chandigarh (1988) and Bachelors in Science from Punjab University (1984). She is a Professor at Giani Zail Singh Campus College of Engineering & Technology, Maharaja Ranjit Singh Punjab Technical University, Bathinda, since 2005. Earlier Dean (R&D), Dean (Academics) she is now Dean (Planning & Development) at Maharaja Ranjit Singh Punjab Technical University, Bathinda, India also. Her current areas of interest include energy-efficient and fault-tolerant scheduling on parallel & distributed computing systems, wireless sensor networks, image processing and wireless communication systems. She has more than 100^{+} publications at the International level. She is available at savina.bansal@gmail.com.

Rakesh Kumar Bansal from India earned his Ph.D. from Punjabi University, Patiala and Masters in Computers and Bachelors in Engg. (1986) both from TIET, Patiala. He is a Professor at Giani Zail Singh Campus College of Engineering & Technology, Maharaja Ranjit Singh Punjab Technical University, Bathinda, India since 2009. His areas of interest include real time fault-tolerant multiprocessor scheduling, distributed computing and wireless sensor networks. He has 65^{+}publications at the International level.. He is available at drrakeshkbansal@ gmail.com.