Optimising resource utilisation through Load Balancing in Cloud

A major project report submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Dazzle (201414) & Tanya Gupta(201279)

Under the guidance & supervision of

Dr. Shubham Goel

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

Candidate's Declaration

We hereby declare that the work presented in this report entitled **'Optimising resource utilisation through Load Balancing in Cloud'** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Shubham Goel** (Assistant Professor, Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Students Name: Dazzle

Roll No.: 201414

Students Name: Tanya Gupta

Roll No.: 201279

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Supervisor Name: Dr. Shubham Goel

Designation: Assistant Professor

Department: Computer Science & Engineering and Information Technology

Dated: 15th May, 2024

Acknowledgement

Firstly, We express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to do the project work successfully.

We are really grateful and wish our profound indebtedness to Supervisor Dr. Shubham Goel (Assistant Professor), Department of CSE Jaypee University of Information Technology, Wakhnaghat.

His deep knowledge,endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to do this project.

We would like to express our heartiest gratitude to Dr. Shubham Goel (Assistant Professor), Department of CSE, for his kind help to finish my project.

We would also generously welcome each one of those individuals who have helped us straightforwardly or in a roundabout way in making this project a win. In this unique situation, We might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, We must acknowledge with due respect the constant support and patience of our parents.

Dazzle (201414) Tanya Gupta (201279)

List of Figures

AVERAGE TIME TAKEN VS ALGORITHM(S)

FIGURE 15 :

MAKE SPAN TIME VS ALGORITHM(S)

FIGURE 16 :

MINIMUM TIME TAKEN VS ALGORITHM(S)

FIGURE 17:

MAXIMUM TIME TAKEN VS ALGORITHM(S)

FIGURE 18 :

AVERAGE START TIME VS ALGORITHM(S)

FIGURE 19 :

LAST FINISH TIME VS ALGORITHM(S)

List of Tables

Table of Contents

CHAPTER-5 RESULTS AND EVALUATION

CHAPTER-6 CONCLUSIONS AND FUTURE SCOPE

ABSTRACT

Modern computing paradigms have been completely transformed by the widespread adoption of cloud computing, which provides scalability and flexibility in resource provisioning. Effective resource management is still a major issue in cloud environments, though. This project looks into how load balancing methods in the cloud can optimise resource allocation. The main goal is to create an intelligent load balancing solution in order to address issues associated with either excessive or insufficient use of resources. The first step of the project is a thorough review of the literature to determine any gaps or restrictions in the current load balancing mechanisms. The requirements analysis, design, architecture, data preparation, and implementation that follow are all part of the system development phase, which also addresses any obstacles that arise.

A comprehensive testing approach is utilised, succeeded by the display and explanation of outcomes, encompassing possible juxtapositions with well-established resolutions. A summary of the project's main conclusions, constraints, and contributions to the field is provided. It also describes potential directions for future research, imagining improvements and breakthroughs in resource optimization via adaptive load balancing techniques in cloud computing settings.

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

The exponential expansion and advancement of computing, especially in the area of cloud computing, have brought about a fundamental revolution in the way businesses use and adapt their computational resources in the quickly changing technological landscape of today. The foundation of this digital revolution is cloud computing, which provides an extensive range of online services including servers, storage, networking infrastructure, databases, and more. Access to this comprehensive package of services is provided by numerous well-known suppliers, including Google Cloud, Microsoft Azure, Amazon's AWS, and an ever-growing list of industry participants. Cloud computing is the key to revolutionising resource management paradigms and simplifying information accessibility inside the cloud ecosystem by enabling remote access to this wide range of applications and computational resources. Because of cloud computing's unmatched flexibility, scalability, and affordability, organisations may now overcome old barriers to optimise operations, boost productivity, and seize previously unattainable chances for innovation and expansion in the digital era.

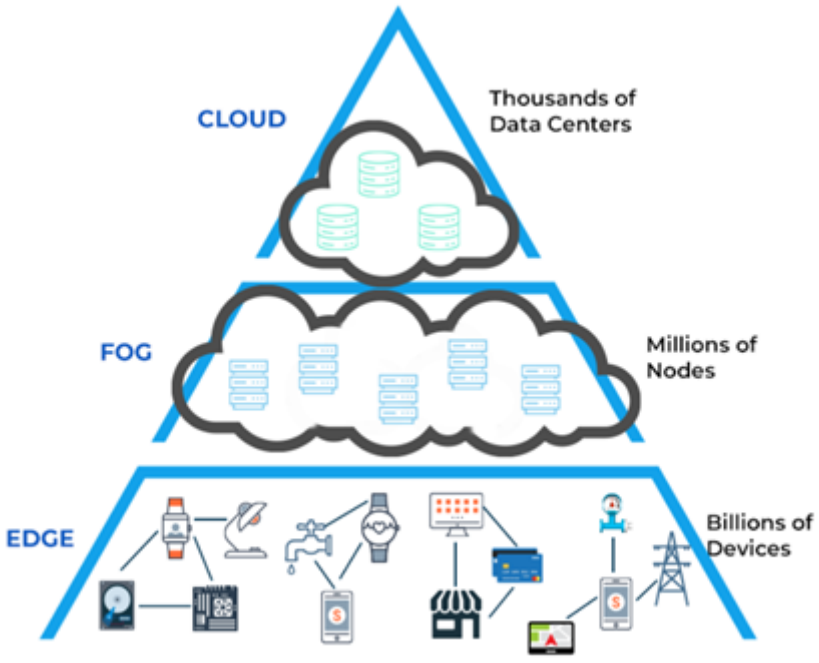

FIGURE 1: STRUCTURE OF CLOUD FOG SYSTEM

CLOUD SERVICE MODELS

Three main service models have become the cornerstones of digital innovation within the vast field of cloud computing: Infrastructure as a Service (IAAS), Platform as a Service (PAAS), and Software as a Service (SAAS). These service models were cleverly designed to meet and surpass the increasing demands of the global market for a wide variety of online services, including as servers, storage, networks, and databases. While PAAS provides application developers with extensive platforms to build, deploy, and maintain their applications, SAAS focuses directly on end users, providing smooth access to software applications. Conversely, IAAS meets the needs of network architects by providing the fundamental tools required to design and oversee intricate network infrastructures.

But as the digital world changes and the complexity of contemporary computer programmes grows, a ground-breaking addition to cloud computing's capabilities has surfaced: the fog computing paradigm. This innovative idea has received a lot of praise for efficiently bringing processing power to the edge of the computer network, which reduces latency and improves performance all around. By enhancing the cloud's conventional architecture, fog computing effectively decentralises computational resources and distributes them closer to the point of data generation and consumption, thereby addressing the increased complexity inherent in modern computing applications. By doing this, fog computing not only increases infrastructure efficiency but also opens up a world of possibilities for innovation and optimisation in the dynamic digital world.

CLOUD DEPLOYMENT MODELS

Now about deployment we are having a total 4 Cloud Deployment models: 1. Public Cloud, 2. Private Cloud, 3. Community Cloud, 4. Hybrid Cloud. They function as a virtual environment for cloud or other computing with an architecture depending on the amount of data. All these four deployment models are named and defined according to the whereabouts of the infrastructure of the environment. In meeting the needs of the different organisations these deployment models are essential. These models provide the flexible frameworks to all the companies so that they can customise their computing infrastructure in order to meet certain security requirements.

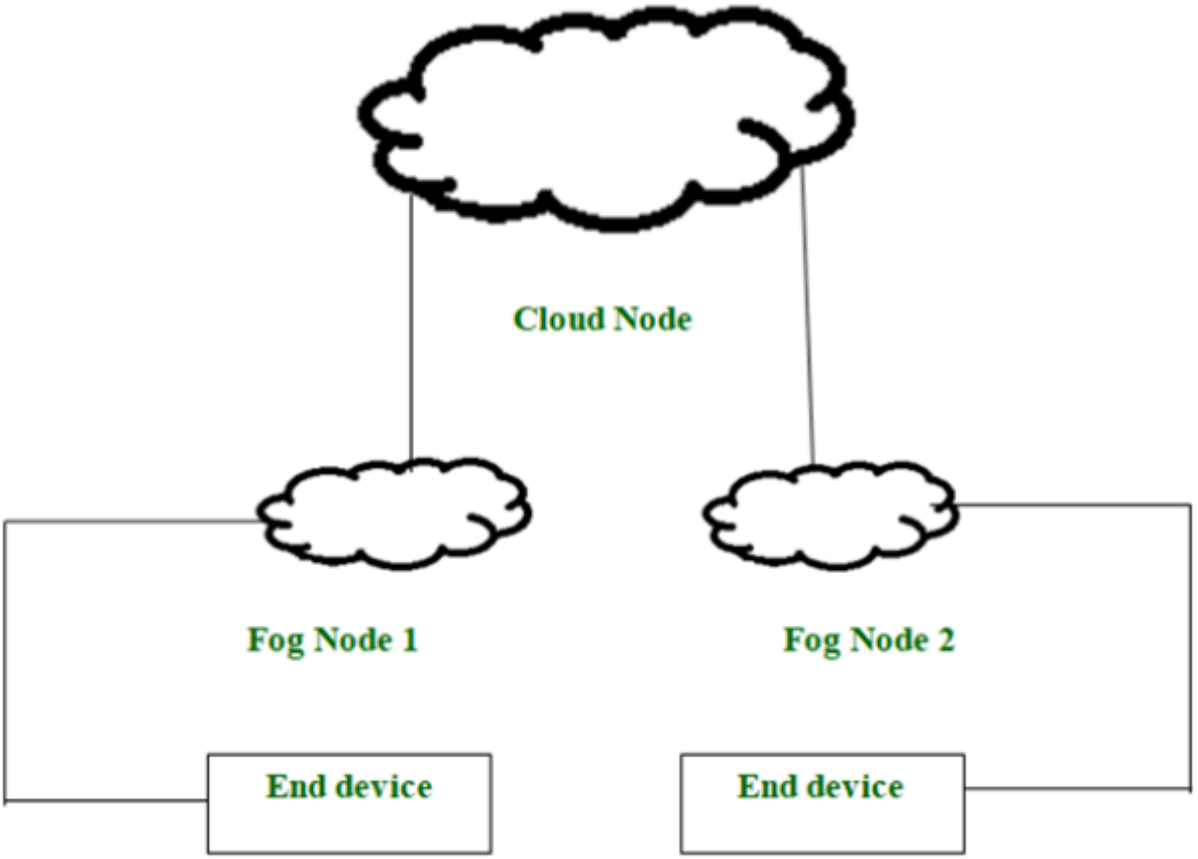

FIGURE 2: FOG COMPUTING ARCHITECTURE

TASK SCHEDULING

Task Scheduling is an important component for resource management. And for task scheduling we have two main algorithms Distributed and Centralised. Now the difference in both of them. As there is only one point of control present, the setup can be easier with the centralised scheduling type. But problems will reside in the event of system failure. On the other side, we have the second type that is distributed scheduling, It uses cooperative mapping between the schedulers. Distributed Scheduling provides increased scalability and also fault tolerance as these two are very important in the modern world.

Now we have seen two terms first being fog computing and second being the task scheduling. When these two are combined, that is fog computing being combined with task scheduling, then the allocation of computing resources among the various tasks is optimised which in turn maximises the resource utilisation and also improves the system efficiency.

VIRTUAL MACHINES

Now, VM's that are Virtual Machines are important components of the cloud computing environment and infrastructure. It is an isolated cloud computing environment which is created by resources(uses software to run programs and deploy apps) from a physical machine. That is why Virtual machines allow us to use more than one OS(Operating Systems) at same time on the same physical machine and hardware. It works by simulating the physical environment through the software. About control, Virtual Machines (VM's) are being controlled by a VMM that is a Virtual machine monitor or a hypervisor. In turn it provides high level segregation of the operating system and apps.

Virtual Machines (VMs) have become essential tools in today's ever-changing technological landscape, serving a wide range of businesses worldwide due to their exceptional flexibility, portability, availability, and resource optimisation capabilities. Virtual machines (VMs) have become essential tools that enable organisations to improve efficiency, encourage creativity, and streamline operations. This is especially true for industries that are deeply involved in cloud computing and software development, deployment, and testing.

"Optimising Resource Utilisation Through Load Balancing in the Cloud," our project, is well-positioned to meet the changing needs of this quickly growing technological environment. Considering the rapid and revolutionary developments of the last few years, our main goal is to explore the complex field of load balancing strategies in cloud computing settings. Through a thorough investigation, we hope to gain a deeper comprehension of the various strategies used to efficiently disperse workloads around cloud datacenters.

Furthermore, we are not content to stop at theoretical research—we are also dedicated to investigating real-world applications that utilise advanced load balancing techniques to optimise energy efficiency in cloud datacenters. The strategic placement of Virtual Machines in a way that maximises performance, minimises energy consumption, and makes the best use of the resources at hand is essential to this goal.

Our project aims to make a significant contribution to the field of cloud computing by means of thorough investigation, testing, and analysis. In the end, we hope to clear the path for improved resilience, sustainability, and efficiency in the digital infrastructure of the future.

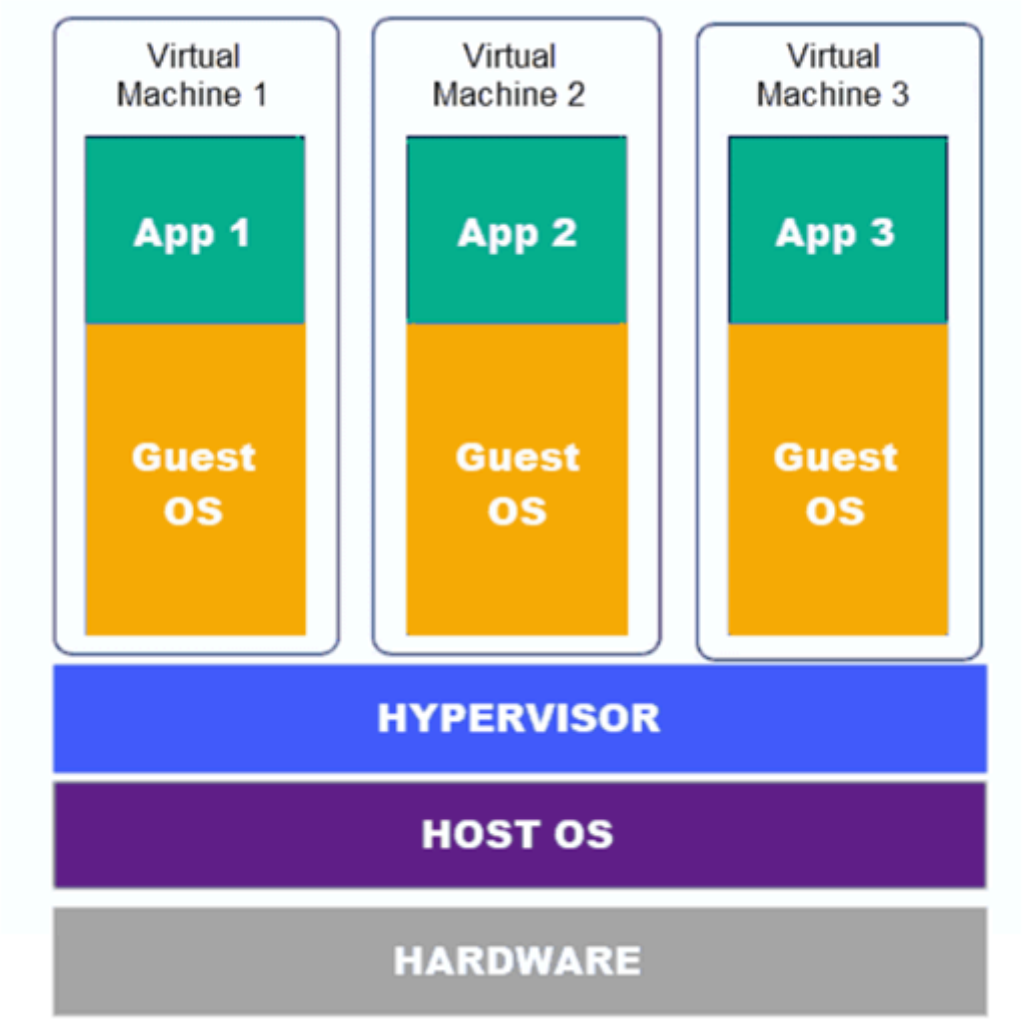

FIGURE 3: VIRTUAL MACHINE

JOB SCHEDULING ALGORITHMS

Job scheduling algorithms are very important for taking care of task execution in the computer systems for proper resource allocation and utilisation. These job scheduling algorithms maximise performance of the system and resource utilisation by telling the sequence in which tasks are executed.

ROUND ROBIN: It is a famous and popular scheduling algorithm which is straightforward and impartial. The tasks are given a defined sequence which they need to follow for their turn to come up in a repeating sequence. Hence, every process is assigned a fixed time slot in a sequence. A task is moved to the back place of the queue if its execution has not been able to be completed within the allotted time period. Tasks are to be executed for an already defined time quantum. As Round Robin allocates the same processor time to each task, therefore, Round robin scheduling guarantees that it's fair to all. Also it provides better performance and responsiveness.

FCFS: First Come First Serve is a very simple job scheduling algorithm. It assigns the task slot to processes according to their coming time as in the time they arrive. The First one to come is the first one to be served or handled. Jobs are sended into a queue and carried out one after another in that queue. The first task in the queue is executed at first followed by the other one in the queue and the task last in the queue is executed at last. All the other tasks in the queue wait for their turn to come up. FCFS is the easiest job scheduling algorithm to be set up and executed but it has a problem of convoy effect. It delays short tasks because long running tasks arrive first.

SHORTEST JOB FIRST(SJF): This technique reduces waiting time by executing the shortest time taking tasks first,and then it ranks tasks according to their burst or execution time.

PRIORITY SCHEDULING: In this tasks are given priorities through priority scheduling, which enables higher priority tasks to be completed first. Depending on the demands of the system, it can be either preemptive or non-preemptive.

MULTILEVEL QUEUE SCHEDULING: Sorts work according to priority by splitting the ready queue into several smaller queues, each with its own scheduling algorithm.

Efficient job scheduling algorithms are critical to meeting performance targets, maximising resource utilisation, and guaranteeing equitable resource distribution among diverse tasks in cloud computing environments. Typically, cloud data centres manage a variety of workloads with different resource requirements. Appropriate scheduling algorithms can minimise response times, increase system throughput, and boost overall system efficiency.

In order to distribute resources efficiently, load balancing techniques frequently incorporate job scheduling algorithms. The goal of load balancing is to split up workloads among several computer resources in order to maximise efficiency and prevent bottlenecks. Cloud-based systems' overall performance and efficiency can be greatly impacted by the load balancing framework's choice of suitable job scheduling algorithms.

Through our project "Optimising Resource Utilisation Through Load Balancing in Cloud," we want to make it possible to work on optimising resource utilisation and improving system performance in cloud data centres through the comprehension and application of effective job scheduling algorithms. These algorithms will be essential to the creation of energy-efficient load balancing systems designed for cloud environments with dynamic resource allocation.

1.2 PROBLEM STATEMENT

- Research will be conducted to explore various cloud load balancing techniques to understand their energy efficiency and workload optimization mechanisms.
- Task scheduling is an essential component of resource management in contexts where applications and cloud-fog networks experience optimization-related issues.
- The project aims to develop an energy-efficient load balancing solution that optimises virtual machine allocation, and adapts to dynamic resource needs in cloud data centres.

Challenges with Efficient Resource Allocation and Utilisation: Cloud data centres encounter difficulties with this. Due to varying workloads and multiple applications running at once, resources are frequently used inefficiently, which can result in either overuse or underuse of servers and virtual machines (VMs).

The Significance of Load Balancing: Load balancing plays a crucial role in allocating incoming network traffic or computational tasks among several servers or resources. An uneven workload distribution caused by inadequate load balancing can result in bottlenecks, decreased performance, and wasted energy.

Energy Efficiency Issues: One major issue is the high energy usage in cloud data centres. Aside from its negative effects on performance, inefficient resource allocation raises operating expenses and negatively affects the environment by causing needless power consumption.

Workloads in cloud environments are dynamic, changing over time in terms of intensity and resource requirements. This is known as dynamic workload adaptation. The inability of static resource allocation to adjust to these shifting demands frequently results in subpar resource utilisation and possible performance degradation.

Requirement for an Optimal Solution: An intelligent, flexible, and energy-efficient load balancing system is vital to meet these challenges. This solution should optimise virtual machine (VM) allocation based on workload characteristics, dynamically allocate resources, and guarantee minimal energy consumption without sacrificing performance.

This project's main goal is to investigate and assess different load balancing strategies and how they affect workload optimization and energy efficiency in cloud-fog networks. The goal of the research is to create and put into practice a novel load balancing solution that maximises energy consumption, adjusts to fluctuating demands, and intelligently distributes resources while improving overall system performance.

Project Outcome: The goal of the project is to provide a thorough understanding of load balancing mechanisms and how they affect performance optimization, energy efficiency, and resource utilisation. In the end, it aims to suggest and create a novel load balancing strategy that cloud data centres can actually use to greatly improve resource usage, energy efficiency, and overall system performance.

1.3 OBJECTIVES

- To understand the current situation and pinpoint opportunities for development, a thorough examination of the literature on task scheduling approaches for fog computing is essential. Through a thorough analysis of previous studies and research designs, this work seeks to provide light on the advantages, disadvantages, and general patterns influencing task scheduling in fog computing settings. By means of an extensive synthesis of the current literature, we strive to extract significant discoveries, approaches, and optimal procedures that guide our future research undertakings.

- The use of cloud and fog computing paradigms for task scheduling is then thoroughly evaluated in our work, with a special emphasis on important performance measures including reaction time, throughput, and computational cost. We want to clarify the advantages and disadvantages of various computer models by empirical evaluation in practical settings, which will help to guide resource allocation and strategic decision-making.

- Additionally, our work aims to perform a comparative study of effective task scheduling algorithms, focusing on how well they can assign jobs to computing resources according to workload dynamics, availability, and performance. We seek to find and characterise algorithms that demonstrate greater scalability, reliability, and adaptability in handling the various needs of fog computing settings by means of thorough benchmarking and performance studies.

- Using knowledge from the literature review and empirical evaluations, our work aims to shed light on important directions for improving fog computing task scheduling state-of-the-art. By combining theoretical understanding with real-world experience, we hope to provide new perspectives, approaches, and solutions that spur creativity and improve the effectiveness, scalability, and resilience of fog computing ecosystems.

1.4 SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

Effective Resource Usage: Cloud computing environments frequently face issues with either excessive or insufficient resource use. Ensuring that computational resources, such as servers and virtual machines, are efficiently allocated based on workload demands requires optimising resource utilisation through effective load balancing. This optimization may result in lower costs, enhanced system efficiency overall, and better performance.

Energy Efficiency and Cost Reduction: In cloud data centres, inefficient resource allocation leads to wasteful energy use. By concentrating on energy-efficient load balancing strategies, the project hopes to minimise power consumption and consequently lower operating costs and environmental impact. A cloud infrastructure that is more economical and sustainable can result from optimising resource usage.

Adaptability to Dynamic Workloads: Resource demands and workloads change over time in cloud environments. These dynamic circumstances are handled by an adaptive load balancing system, which cleverly reallocates resources in accordance with the shifting workload patterns. Because of its flexibility, resources are used as efficiently as possible, allowing for peak performance even in periods of workload fluctuations or spikes.

Technological Advancements in Cloud Computing: As cloud computing technologies continue to develop, creative resource management strategies are always needed. By introducing more intelligent and adaptable resource allocation mechanisms, the project's focus on creating an effective load balancing solution advances the infrastructure of cloud computing.

1.5 ORGANISATION OF PROJECT REPORT

The research on optimising resource utilisation through load balancing in cloud computing settings is extensively covered in this project report, which is divided into multiple chapters, each with a specific function. As the introduction, Chapter 1 gives a succinct summary of the importance and extent of the project. Subheadings in this chapter clarify the issue statement, objectives, importance, and motivation. They also establish the framework for the next few chapters and prepare the reader for a detailed examination of the subject.

In-depth analysis of relevant literature is provided in Chapter 2, which also provides insights into current research, technologies, and methodologies pertaining to cloud load balancing. Knowledge gaps are found in this review, which lays a strong basis for creative solutions that will advance the area.

The complexities of system development are covered in detail in Chapter 3, including requirements analysis, project design, architecture, data preparation, implementation details, and difficulties that may arise. This chapter lays the foundation for the project's later stages by diving into the technical nuances of converting abstract ideas into workable solutions.

In Chapter 4, the emphasis is shifted to the methods and strategies used for testing, with particular test cases and their outcomes being covered. This chapter provides insightful information on the developed solution's effectiveness and dependability in practical situations by carefully evaluating its functionality and performance.

In Chapter 5, the emphasis is on showcasing and evaluating the test findings, along with maybe drawing comparisons between the developed solution and other methods. The chapter seeks to clarify the advantages and disadvantages of the suggested solution by a thorough evaluation of these findings, providing insightful information for further modifications and advancements.

The project's results are finally summarised in Chapter 6, which also highlights its contributions and discoveries. This chapter also considers the project's shortcomings and provides a forward-looking viewpoint by talking about possible directions for future investigation. Chapter 6 ensures that the project's impact goes beyond its immediate focus by concluding the report with a look ahead, providing the foundation for future innovation and advancement in the cloud load balancing sector.

CHAPTER 2: LITERATURE SURVEY

2.1 OVERVIEW OF RELEVANT LITERATURE

S. no.	Paper Title	Year	Tools/ Techniques	Results	Limitations
1.	Overcome load balancing problem by weight based scheme on CLOUD	2023	Cloud computing, MATLAB, Improved Genetic algorithm	Faster response times, Faster task completion, Lower energy consumption.	Long execution time for the enhanced genetic algorithm
2.	A Load Balancing Algorithm for the Data Centres to Optimise Cloud Computing Applications	2021	Load balancing Algorithm Cloudsim Simulator	Achieved an average of 78% resource utilisation, More efficient task scheduling.	Resource utilisation improved only moderately compared to the existing algorithm.
3.	Load Balancing Using Cloudsim Simulator In Cloud Computing	2020	CloudSim, Cloud Reports, HTML Reports, Resource Utilisation Graphs	Produced resource utilisation graphs, including power consumption data.	Does not offer a comprehensive summary of research contributions
4.	Leveraging energy-efficient load balancing algorithms in fog computing	2020	Load balancing algorithms for fog computing iFogSim	Produced resource utilisation graphs, including power consumption data.	Does not offer a comprehensive summary of research contributions
5.	Energy Efficient Load Balancing in Cloud Data Center Using Clustering Technique	2019	JADE framework for agent-based simulation, K-means clustering for initial VM allocation	The proposed method reduces the number of migrations due to efficient initial VM allocation.	The paper does not provide detailed quantitative results for the energy savings.

6.	Task Scheduling in Fog Enabled Internet of Things for Smart Cities	2017	Genetic algorithms for task scheduling optimization. Simulation for performance evaluation.	The ADGTS algorithm improves task makespan in fog-enabled environments compared to Min-Min.	Lack of real-world implementation.

Energy consumption in cloud data centres is the subject of a paper titled "Energy Efficient Load Balancing in Cloud Data Center Using Clustering Technique". The authors suggest an agent-based model that reduces response time and energy consumption by using k-means clustering for initial virtual machine allocation. They perform server consolidation to lower the number of active servers and energy consumption, and they use a novel clustering agent to enhance load balancing. As user requests increase, the study shows a reduction in virtual machine migrations and an improvement in response time. Future directions for research include investigating intelligent or self-organised agents for enhanced energy efficiency, live migration heuristics, and dynamic threshold adjustment.

Due to growing energy costs and environmental concerns, the paper "A survey on techniques to achieve energy efficiency in cloud computing" examines the urgent need to reduce energy consumption in cloud computing. It examines a number of energy-saving technologies, such as scheduling techniques, virtualization, hardware optimization, and clustering. Energy-efficient hardware, energy-aware scheduling, consolidation, and energy conservation in server clusters are the four primary categories into which the paper divides energy-saving strategies. In order to achieve significant energy savings, it emphasises how important it is to implement specific plug-ins and energy-control centres for large-scale hardware and software networks. The creation of such solutions to improve energy efficiency in cloud computing environments may be the focus of future research.

"Load Balancing Using Cloudsim Simulator In Cloud Computing" is a paper that emphasises the importance of cloud computing in terms of making resources accessible and how widely it is used in businesses, mainly due to virtualization. It highlights how cloud computing uses a lot of energy from server machines and how resource allocation is necessary. CloudSim and Cloud reports help with simulation and modelling of cloud environments. With CloudSim providing a useful modelling and simulation solution, the paper tackles the problem of effectively allocating resources such as Virtual Machines, CPU, and Memory to minimise energy consumption in the dynamically expandable cloud computing environment. One useful tool for evaluating power consumption,

resource usage, and infrastructure costs in customised cloud environments is mentioned: cloud reports.

A load-balancing algorithm for cloud computing application optimization in Infrastructure as a Service (IaaS) models is introduced in the paper "A Load Balancing Algorithm for the Data Centers to Optimise Cloud Computing Applications." It takes into account VM priority and QoS parameters, and places an emphasis on efficiently allocating tasks to meet Service Level Agreements (SLAs). According to the results, resource utilisation has improved by 78% when compared to current techniques, indicating its effectiveness in dynamic cloud environments. In order to improve performance, future work will refine cloud resource optimization and take into account extra SLA parameters.

Fog computing is examined in the paper "Leveraging Energy-Efficient Load Balancing Algorithms in Fog Computing" as a potential remedy for latency and connectivity problems in cloud and smart device environments. It talks about different fog computing load balancing algorithms, like Source IP Hash and Round Robin, and stresses the significance of energy-efficient load balancing methods. The study offers research and developers in the field a useful resource by shedding light on load balancing techniques and their applicability in fog environments. As the need for effective traffic management grows, future directions point to the potential for these algorithms in emerging technologies, online gaming, video streaming, and applications for social causes.

In cloud computing environments, load balancing presents a challenge that is addressed in the paper "Overcome Load Balancing Problem by Weight-Based Scheme on Cloud". It talks about how cloud computing is dynamic and emphasises how load balancing reduces efficiency. The study suggests using an enhanced genetic algorithm to improve virtual machine migration and handle load balancing issues. The modified genetic algorithm produces better results in terms of reaction time, completion time, energy consumption, cost, and number of migrations. The results show that the modified genetic algorithm reduces execution time while maintaining reliability and speed. Further research could entail comparing this approach to other migration strategies, integrating it with them, and optimising the results using a hybrid meta-heuristic algorithm.

2.2 KEY GAPS IN THE LITERATURE

Long Execution Times for the Enhanced Genetic Algorithm: This constraint suggests that, despite having the potential to be successful in resource allocation optimization, the suggested enhanced genetic algorithm has long execution times. Long execution times may make it less useful in high-demand or real-time cloud environments, which would reduce its viability for managing dynamic workloads.

Moderate Resource Usage Improvement Compared to Current Algorithms: The project's suggested solution may show only modest increases in resource usage in comparison to current or established load balancing algorithms. This limitation implies that even if the solution offers improvements, they might not be significant enough to surpass the state-of-the-art methods that are currently in use.

A thorough synopsis of the research contributions is necessary to highlight the project's originality and distinctive worth. The project may not effectively convey the importance or originality of its suggested load balancing solution if it is unable to succinctly describe its contributions.

In order to verify the efficacy and significance of the suggested solution, quantitative results are essential. The project's ability to persuasively illustrate the improvements in energy efficiency could be limited if comprehensive quantitative data about the energy savings made possible by the load balancing approach are lacking.

Validating the proposed solution's practical applicability and effectiveness requires a real-world implementation. Lack of such implementation could undermine the project's credibility and relevance by restricting its ability to show how the load balancing technique would function in a real cloud data centre environment.

There are several approaches that could be used to address these limitations:

- Optimising Algorithm Efficiency: To make the algorithm more viable for real-time or high-demand cloud environments, try to improve its execution time without sacrificing its efficacy.
- Improving Resource Usage: Optimise the suggested approach to greatly increase resource usage, with the goal of achieving a greater performance increase than current algorithms.
- Detailed Synopsis of Contributions: Express the project's unique contributions and innovations in the research field in a clear and concise manner.

- Quantitative Information to Reduce Energy Use: Perform comprehensive quantitative analyses to produce comprehensive and convincing data illustrating the energy savings attained by using the suggested load balancing strategy.
- Real-World Implementation: To confirm the efficacy and viability of the suggested solution, think about running tests or simulations in actual cloud environments.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 REQUIREMENTS AND ANALYSIS

In the development of an effective load balancing solution for cloud data centres, a thorough understanding of requirements and a detailed analysis of the system are essential. This section outlines the key steps involved in defining the requirements and conducting a comprehensive analysis.

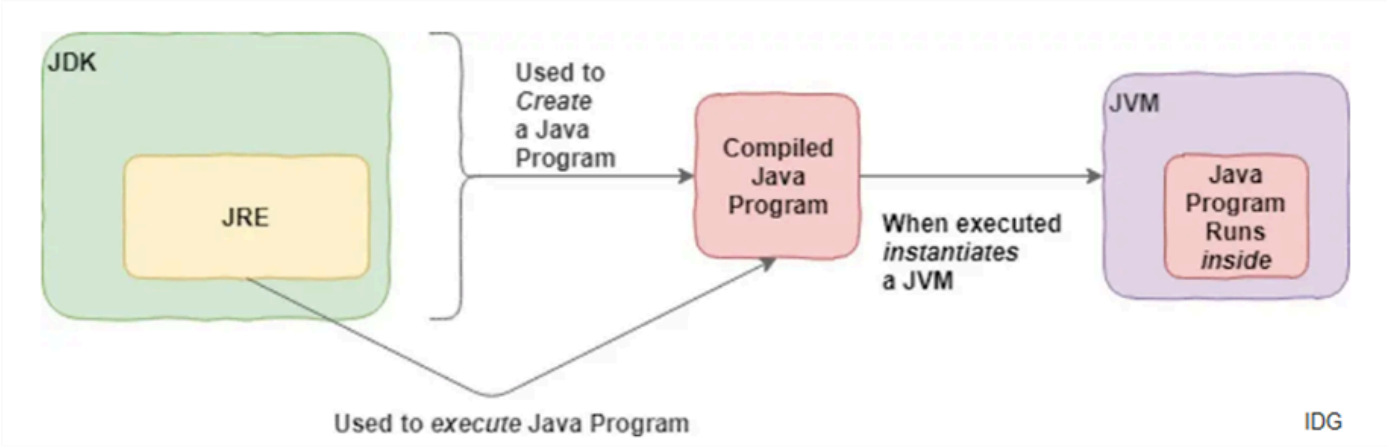

FIGURE 4: JAVA PROGRAM EXECUTION

SYSTEM REQUIREMENTS

To commence the development phase of our project, it is imperative to delineate the system requirements comprehensively.
This includes:

FUNCTIONAL REQUIREMENTS:

Load Balancing Algorithm Implementation: Develop and implement the chosen load balancing algorithm (e.g., Round Robin, Weighted Round Robin) to optimise resource utilisation.

Task Scheduling Module: Create a robust task scheduling module to allocate tasks to computing resources based on availability, performance, and workload.

Virtual Machine Management: Implement a system for efficient virtual machine allocation, considering dynamic resource needs in cloud data centres.

Monitoring and Reporting: Incorporate monitoring tools to track system performance metrics and generate comprehensive reports.

NON-FUNCTIONAL REQUIREMENTS:

Performance: Ensure the load balancing system enhances overall system performance and minimises response times.

Scalability: Design the system to scale seamlessly with an increasing number of tasks and computing resources.

Fault Tolerance: Implement mechanisms to handle system failures gracefully and ensure continuous operation.

Usability: Develop an intuitive user interface for system administrators to monitor and manage the load balancing process.

3.1.2 SYSTEM ANALYSIS

TASK SCHEDULING ANALYSIS:

Algorithm Evaluation: Analyse the chosen task scheduling algorithm's effectiveness in terms of response time, throughput, and cost of computation.

Comparison with Alternatives: Evaluate the selected algorithm against alternative scheduling algorithms to justify its suitability.

LOAD BALANCING ANALYSIS:

Algorithm Performance: Assess the performance of the implemented load balancing algorithm under varying workloads.

Impact on Energy Efficiency: Investigate how the load balancing solution influences energy consumption in cloud data centres.

VIRTUAL MACHINE MANAGEMENT ANALYSIS:

Optimization Techniques: Study and implement optimization techniques for virtual machine allocation to achieve energy efficiency and resource optimization.

Adaptability to Dynamic Resource Needs: Analyse the system's ability to adapt to changing resource requirements in real-time.

3.2 PROJECT DESIGN AND ARCHITECTURE

SYSTEM ARCHITECTURE CLOUD FOG COMPUTING INTEGRATION

The architectural structure of the system is envisioned to seamlessly combine both cloud and fog computing features, exploiting the capabilities of each paradigm. Fog computing, functioning at the network edge, will be crucial in reducing data transmission latency and increasing overall system efficiency.

Load Balancing Module: Load Balancing Algorithm Implementation: At the heart of the system architecture lies the implementation of the chosen load balancing algorithm, precisely developed to distribute jobs efficiently across the varied computing resources at the system's disposal.

Dynamic Resource Allocation: A critical component within the architecture involves the implementation of a dynamic resource allocation module. This module will dynamically change virtual machine allocation in response to the real-time fluctuations in resource needs.

TASK SCHEDULING MODULE:

Algorithm Integration: The task scheduling module easily incorporates the selected scheduling technique, meticulously evaluating critical parameters such as response time, throughput, and cost of calculation.

Real-time Monitoring: A crucial element of the architecture is the integration of monitoring tools to track and evaluate task execution times, offering vital insights into the system's performance.

VIRTUAL MACHINE MANAGEMENT:

Optimization strategies: The design comprises the deployment of optimization strategies aimed at refining the allocation of virtual machines. These strategies prioritise energy efficiency and resource optimization.
Adaptive Allocation: An important element is the implementation of adaptive allocation techniques, permitting the system to intelligently distribute virtual machines based on the increasing resource requirements.

DATABASE DESIGN SYSTEM DATABASE:

Task Information: The system's database structure involves the storage of complete task-related information, including task kinds, resource needs, and historical execution times.

Virtual Machine Data: Another vital part requires keeping a robust database of virtual machines, recording specifications, current utilisation, and historical data.

TOOLS AND TECHNOLOGIES

The development and execution of the project leverages a collection of carefully chosen tools, technologies, and programming languages to provide robustness, efficiency, and seamless integration. The selection of these components aligns with the project's aims and requirements.

PROGRAMMING LANGUAGE: JAVA

Platform Independence: Java's "write once, run anywhere" approach ensures platform independence, a critical component in a diversified computing environment.

Rich Ecosystem: The broad Java ecosystem provides a multitude of libraries and frameworks, facilitating speedy development and debugging.

Community Support: A broad and active Java community ensures accessibility to materials and support during the development process.

SIMULATION FRAMEWORK: CLOUDSIM

Cloud Environment Modeling: CloudSim is specifically built for modeling and simulating cloud computing environments, enabling for the testing and validation of the created load balancing and task scheduling modules.

Customization: The flexibility and customization possibilities given by CloudSim enable the simulation of varied cloud settings, providing comprehensive testing.

Community Adoption: Being widely embraced in academics and industry, CloudSim fits with best practices and community standards.

SCALABILITY AND FAULT TOLERANCE

Scalability: The architectural plan promotes scalability, enabling the system to extend horizontally to accommodate a burgeoning number of jobs and virtual machines effortlessly.

Fault Tolerance: Recognizing the possibility of system failures, the architecture integrates redundancy and failover solutions. These techniques are meant to handle unexpected failures without affecting the continued operation of the system.

The thorough architectural design creates a solid platform for the succeeding phases of development and implementation. It strives to design a system that is not only strong and adaptive but also corresponds closely with the specified requirements and analysis, ensuring a successful realisation of the project goals.

3.3 IMPLEMENTATION

The implementation phase entails translating the design specifications into functional code, ensuring that the system adheres to the defined architecture. This section provides a summary of the primary components implemented and the approaches followed during this phase.

LOAD BALANCING ALGORITHM IMPLEMENTATION

DYNAMIC WORKLOAD ADAPTATION

To address the dynamic nature of workloads, the implemented load balancing algorithm incorporates:

Dynamic Thresholds: The algorithm dynamically modifies thresholds for job allocation depending on real-time monitoring of system performance.

Predictive Analysis: Utilises predictive analysis to foresee potential workload fluctuations, enabling proactive load balancing decisions.

ENERGY EFFICIENCY OPTIMIZATION

Efforts have been devoted towards achieving energy efficiency without compromising performance:

Power-aware Scheduling: The load balancing method includes power-aware scheduling strategies, considering the energy consumption characteristics of each computer resource.

Adaptive Resource Allocation: Virtual machines are dynamically allocated, focusing on energy-efficient servers during peak and off-peak times.

VIRTUAL MACHINE MANAGEMENT IMPLEMENTATION

LOAD BALANCING ALGORITHMS
The virtual machine management module is further enhanced to incorporate allocation to brokers and cloudlet formations, increasing the overall system architecture:

FCFS-BASED LOAD BALANCING

Implementation: FCFS-based load balancing is extended to assign resources to brokers depending on their arrival order. Incoming brokers are served in the order of their initiation, providing fairness in resource distribution.

Objective: FCFS-based load balancing for brokers seeks to offer equitable access to resources for all incoming brokers, retaining a first-come-first-serve approach.

ROUND ROBIN LOAD BALANCING

Implementation: Round Robin load balancing is expanded to allocate resources to brokers and build cloudlets in a cyclic way. Each broker and cloudlet formation request is executed in a cyclical fashion.

Objective: Round Robin load balancing for brokers and cloudlets prevents any single broker or cloudlet from monopolising resources, guaranteeing a balanced distribution.

SHORTEST JOB FIRST LOAD BALANCING

Implementation: SJF-based load balancing is upgraded to assign resources to brokers and build cloudlets depending on the projected execution times of their jobs. Tasks with shorter execution times are preferred.

Objective: SJF-based load balancing for brokers and cloudlets attempts to optimise resource allocation by prioritising jobs with shorter execution times, boosting overall system efficiency.

ADAPTIVE ALLOCATION TO BROKERS

The virtual machine management module's adaptive allocation technique is enhanced to allow dynamic allocation to brokers:

Dynamic Broker Allocation: The system dynamically assigns resources to brokers based on real-time monitoring of their resource demands. This ensures that brokers receive resources appropriate to their workload.

CLOUDLET FORMATIONS

The module includes techniques for the production and control of cloudlets, aligned with load balancing objectives:

Dynamic Cloudlet Formation: The system dynamically forms cloudlets based on the incoming workload. Cloudlets are produced and managed to efficiently process activities.

REAL-TIME MONITORING

The virtual machine management module's real-time monitoring capabilities are enlarged to capture metrics unique to the extended load balancing algorithms, broker allocation, and cloudlet formations:

FCFS Load Balancing Metrics: Monitors waiting times and resource allocation order for incoming brokers under FCFS-based load balancing.

Round Robin Load Balancing Metrics: Tracks the distribution of resource allocations for brokers and cloudlet formations in a circular fashion, measuring the fairness of distribution.

SJF Load Balancing Metrics: Captures data on task burst times, resource allocation order for brokers, and cloudlet forms to measure the efficiency of SJF-based load balancing.

The combination of these elements ensures that the system is ready to handle not just various workloads but also varying resource requirements of brokers and cloudlet formations, encouraging fairness and efficiency in the allocation process. Rigorous testing and monitoring continue to validate the reliability and effectiveness of these linked components.

3.4 KEY CHALLENGES

DYNAMIC WORKLOAD ADAPTATION CHALLENGE:
The changing nature of workloads in cloud settings creates a substantial difficulty. Fluctuations in intensity and resource requirements required a system that can change in real-time to ensure maximum resource use.

ADDRESSING THE CHALLENGE:

Algorithm Flexibility: The load balancing algorithm utilised in the system is designed to be versatile, capable of dynamically responding to fluctuating workloads.

Real-time Monitoring: Continuous real-time monitoring of task execution and resource utilisation feeds the system's decision-making process.

Energy Efficiency Optimization Challenge: Achieving energy efficiency without compromising system performance is a tricky balancing act. The issue lies in devising systems that optimise resource utilisation while minimising energy consumption.

ADDRESSING THE CHALLENGE:

Optimization Techniques: The virtual machine management module features optimization techniques specifically focused at minimising energy usage without losing performance.

Adaptive Allocation: The system's adaptive allocation mechanism dynamically modifies virtual machine allocation to coincide with energy-efficient practices.

Algorithm Performance Challenge: The performance of load balancing and job scheduling algorithms can vary based on the specifics of the workload. Ensuring consistent and high-performance outcomes is a problem.

ADDRESSING THE CHALLENGE:

Algorithm examination: Rigorous examination and testing of the developed algorithms under varied workload circumstances are undertaken to find strengths and flaws.

Iterative refining: The algorithms undergo iterative refining based on the outcomes of testing to boost overall performance.

CHAPTER 4: TESTING

4.1 TESTING STRATEGY

The testing process is critical to confirm the functionality, performance, and reliability of the proposed load-balancing system for cloud data centres. The testing plan involves a thorough approach to assure the robustness of the system.

TYPES OF TESTING

The testing technique comprises several types of testing, including:

Functional Testing: Validates that each component of the system performs as intended. This includes evaluating the load balancing algorithm, task scheduling module, and virtual machine management system.

Performance Testing: Assesses the system's responsiveness, throughput, and resource utilisation under varying workloads. This covers stress testing, load testing, and scalability testing.

Usability Testing: Evaluates the user interface's intuitiveness and efficacy in handling the load balancing process.

Security Testing: Ensures that the system is resilient to security threats and vulnerabilities, ensuring the integrity and confidentiality of data.

Compatibility Testing: Verifies the system's compatibility with multiple cloud settings and provides seamless integration with different computing resources.

TESTING ENVIRONMENTS

- To imitate real-world circumstances, testing will be undertaken in varied environments:
- Development Environment: Initial testing in a controlled context to discover and solve any basic flaws.

Staging Environment: Comprehensive testing in an environment that matches the production setup to identify potential difficulties before deployment.

Production Environment: Final testing to guarantee the load balancing system functions properly in the live cloud data centre environment.

4.2 TEST CASES AND OUTCOMES

Testing is conducted using a series of written test cases that cover all areas of the system. Each test case is built to analyse certain features and circumstances, ensuring a thorough review.

FUNCTIONAL TESTING

LOAD BALANCING ALGORITHM IMPLEMENTATION:

Test Case 1: Validate that the chosen load balancing algorithm distributes tasks equitably across computing resources.
Test Case 2: Confirm that the algorithm reacts to changing workloads and adjusts job allocation dynamically.

TASK SCHEDULING MODULE:

Test Case 3: Ensure the task scheduling module allocates jobs based on availability, performance, and workload.
Test Case 4: Verify that the module incorporates real-time changes in resource needs during task allocation.

VIRTUAL MACHINE MANAGEMENT:

Test Case 5: Validate the efficient allocation of virtual machines considering dynamic resource needs.
Test Case 6: Confirm the implementation of monitoring tools for tracking system performance indicators.

PERFORMANCE TESTING

LOAD BALANCING ANALYSIS:

Test Case 7: Assess the load balancing algorithm's performance under varied workloads.
Test Case 8: Measure the impact of the load balancing system on energy usage in cloud data centers.

VIRTUAL MACHINE MANAGEMENT ANALYSIS:

Test Case 9: Evaluate the adaptability of the system to dynamic resource requirements.

DATABASE DESIGN:

Test Case 10: Confirm the accuracy and speed of retrieving task-related information from the system database.

USABILITY TESTING

Test Case 11: Evaluate the intuitiveness of the user interface for system administrators in monitoring and managing the load balancing process.

TESTING OUTCOMES

The outcomes of the testing phase will be reported, outlining any concerns identified and their resolutions. A complete test report will be prepared, providing insights into the system's performance, dependability, and conformance to the set standards.

CHAPTER 5: RESULTS AND EVALUATIONS

5.1 RESULTS

This study seeks to assess how well scheduling algorithms perform in a fog computing setup. The algorithms will be tested in a heterogeneous fog environment, and the system's performance will be evaluated based on the make span time of the scheduling algorithm.

The following are the parameter constraints we took for the performance analysis of our model:

- AVERAGE TIME TAKEN - This is the average time taken by one process to complete.
- START TIME- When a process is assigned to the processor to be processed.
- MAKE SPAN TIME - The amount of time that elapses between the start and finish of a sequence of operations in a group of machines is referred to as the make span.
- FINISH TIME- The completion time is the point at which the process execution ends.
- MINIMUM TIME TAKEN- This is the Minimum time taken by a process in a Algorithm among all process
- MAX TIME TAKE- This is the Maximum time taken by a process in a Algorithm among all process
- AVERAGE START TIME OF PROCESS FOR VM- This is the average start time of the process in an algorithm
- LAST FINISH TIME FOR A VM – Last finish time meaning the last process that entered its leaving time

In conclusion, the evaluation of scheduling algorithms in a fog computing setting becomes a critical undertaking to guarantee efficient work distribution and best use of available resources. The model's performance was assessed using a number of important parameter constraints, including make span time, average time taken, start time, finish time, minimum and maximum time taken, average process start time for virtual machines (VMs), and the last finish time for a VM. We determined the effectiveness and efficiency of these algorithms through extensive testing in a heterogeneous fog environment, revealing subtle insights into their performance dynamics.

Notably, our results demonstrated the strong influence of starting population sizes on the algorithms' performance, emphasising the role of this parameter in determining the algorithms' operational efficiency. This research adds to the body of knowledge in this field and provides useful insights for future approaches to resource allocation and task management by clarifying the complexities of scheduling algorithms within the fog computing paradigm. Thus, our research lays the groundwork for improving the resilience and effectiveness of fog computing ecosystems and opens the door for the creation of creative solutions that maximise performance and improve operational effectiveness in an ever-evolving technological environment.

FCFS SCHEDULER RESULTS

The FCFS scheduler performance analysis was based on the task ID's start time, end time, and make span time

TABLE 1: FCFS SCHEDULER RESULTS

Process ID	STATUS	Data Center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	2	3197.36	0.2	3197.56
1	SUCCESS	3	6	3323.73	0.2	3323.93
2	SUCCESS	2	2	4115.78	3197.56	7313.34
3	SUCCESS	3	6	470.38	3323.93	3794.3
4	SUCCESS	2	3	2138.51	0.2	2138.71
5	SUCCESS	2	4	1953.32	0.2	1953.52
6	SUCCESS	2	4	1554.43	1953.52	3507.95
7	SUCCESS	2	3	2598.04	2138.71	4736.75
8	SUCCESS	2	3	1870.24	4736.75	6606.99
9	SUCCESS	2	3	730.9	6606.99	7337.88
10	SUCCESS	2	5	2097.9	0.2	2098.1
11	SUCCESS	3	6	2310.03	3794.3	6104.34
12	SUCCESS	3	6	1340.24	6104.34	7444.58
13	SUCCESS	2	4	1840.01	3507.95	5347.96

14	SUCCESS	2	5	1283.63	2098.1	3381.73
15	SUCCESS	2	4	1139	5347.96	6486.96
16	SUCCESS	2	3	2663.89	7337.88	10001.77
17	SUCCESS	2	4	981.83	6486.96	7468.79
18	SUCCESS	2	3	2136.1	10001.77	12137.88
19	SUCCESS	2	2	3386.39	7313.34	10699.74
20	SUCCESS	2	4	1922.57	7468.79	9391.36
21	SUCCESS	2	4	3041.01	9391.36	12432.37
22	SUCCESS	2	3	1486.72	12137.88	13624.6
23	SUCCESS	2	2	1132.2	10699.74	11831.93
24	SUCCESS	2	2	3196.17	11831.93	15028.1
25	SUCCESS	2	3	1865.75	13624.6	15490.35
26	SUCCESS	2	2	2892.62	15028.1	17920.72
27	SUCCESS	3	6	907.71	7444.58	8352.29
28	SUCCESS	2	2	4254.46	17920.72	22175.18
29	SUCCESS	2	3	2491.56	15490.35	17981.91

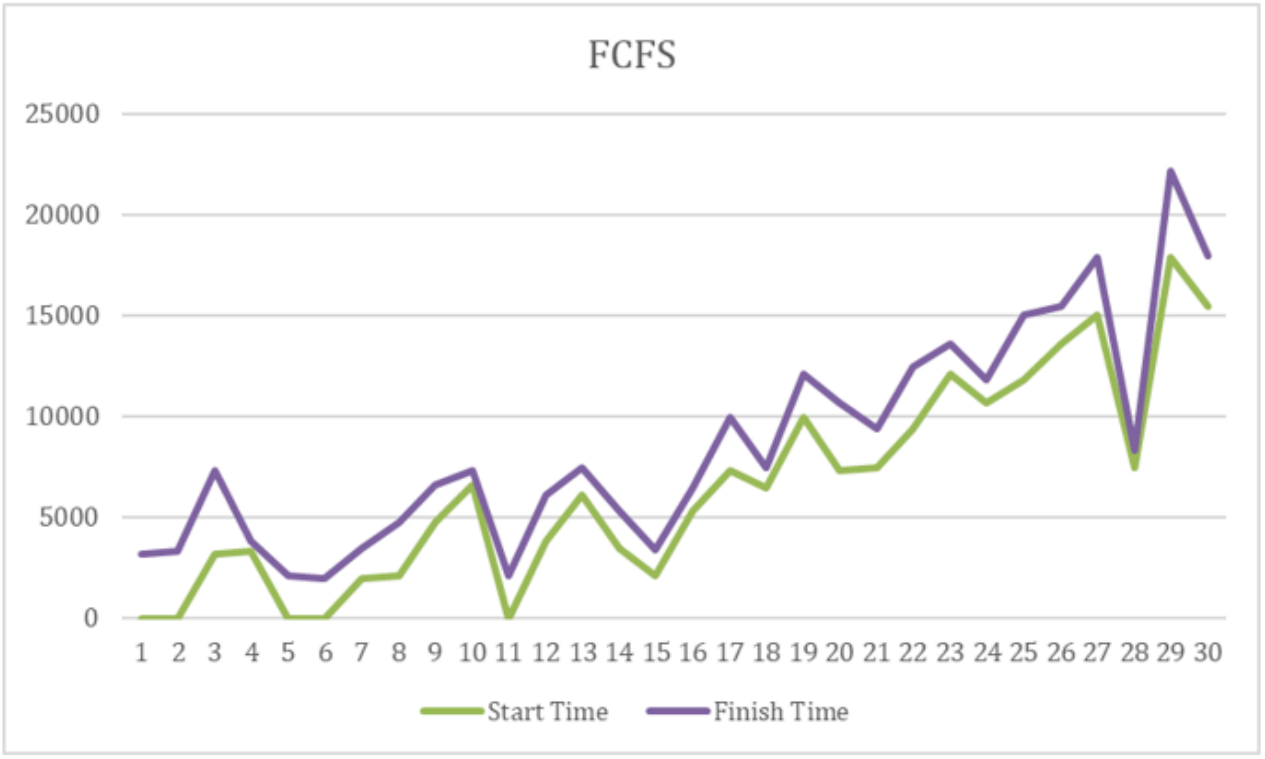

FIGURE 5: PROCESS ID vs START TIME & FINISH TIME

The above plot illustrates the start time and end time by each Process to accomplish the task assigned to distinct VMs in FCFS order.

```
Starting FCFS Scheduler...
Reading the Matrices...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Datacenter_2 is starting...
Datacenter_3 is starting...
Datacenter_4 is starting...
Broker_0 is starting...
Entities started.
0.0: Broker_0: Cloud Resource List received with 5 resource(s)
0.0: Broker_0: Trying to Create VM #2 in Datacenter_0
0.0: Broker_0: Trying to Create VM #3 in Datacenter_0
0.0: Broker_0: Trying to Create VM #4 in Datacenter_0
0.0: Broker_0: Trying to Create VM #5 in Datacenter_0
0.0: Broker_0: Trying to Create VM #6 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #6 to Host #0 failed by RAM
0.1: Broker_0: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker_0: VM #3 has been created in Datacenter #2, Host #0
0.1: Broker_0: VM #4 has been created in Datacenter #2, Host #0
0.1: Broker_0: VM #5 has been created in Datacenter #2, Host #0
0.1: Broker_0: Creation of VM #6 failed in Datacenter #2
0.1: Broker_0: Trying to Create VM #6 in Datacenter_1
0.2: Broker_0: VM #6 has been created in Datacenter #3, Host #0
0.2: Broker_0: Sending cloudlet 0 to VM #2
0.2: Broker_0: Sending cloudlet 1 to VM #4
0.2: Broker_0: Sending cloudlet 2 to VM #4
0.2: Broker_0: Sending cloudlet 3 to VM #3
0.2: Broker_0: Sending cloudlet 4 to VM #4
0.2: Broker_0: Sending cloudlet 5 to VM #6
0.2: Broker_0: Sending cloudlet 6 to VM #4
0.2: Broker_0: Sending cloudlet 7 to VM #3
0.2: Broker_0: Sending cloudlet 8 to VM #2
0.2: Broker_0: Sending cloudlet 9 to VM #3
0.2: Broker_0: Sending cloudlet 10 to VM #2
```

```
<terminated> FCFS_Scheduler [Java Application] C:\Users\samee\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_
5909.324: Broker_0: Cloudlet 18 received
6027.436: Broker_0: Cloudlet 4 received
7932.567999999999: Broker_0: Cloudlet 24 received
8364.616: Broker_0: Cloudlet 10 received
8488.424: Broker_0: Cloudlet 21 received
8762.424: Broker_0: Cloudlet 6 received
9004.856: Broker_0: Cloudlet 22 received
10826.460000000001: Broker_0: Cloudlet 26 received
11207.02: Broker_0: Cloudlet 11 received
11655.528: Broker_0: Cloudlet 15 received
12453.708: Broker_0: Cloudlet 28 received
12961.668: Broker_0: Cloudlet 29 received
13289.351999999999: Broker_0: Cloudlet 17 received
14123.903999999999: Broker_0: Cloudlet 13 received
15933.667999999998: Broker_0: Cloudlet 25 received
17334.904: Broker_0: Cloudlet 16 received
17545.316: Broker_0: Cloudlet 27 received
20501.316: Broker_0: Cloudlet 20 received
20501.316: Broker_0: All Cloudlets executed. Finishing...
20501.316: Broker_0: Destroying VM #2
20501.316: Broker_0: Destroying VM #3
20501.316: Broker_0: Destroying VM #4
20501.316: Broker_0: Destroying VM #5
20501.316: Broker_0: Destroying VM #6
Broker_0 is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Datacenter_2 is shutting down...
Datacenter_3 is shutting down...
Datacenter_4 is shutting down...
Broker_0 is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
```

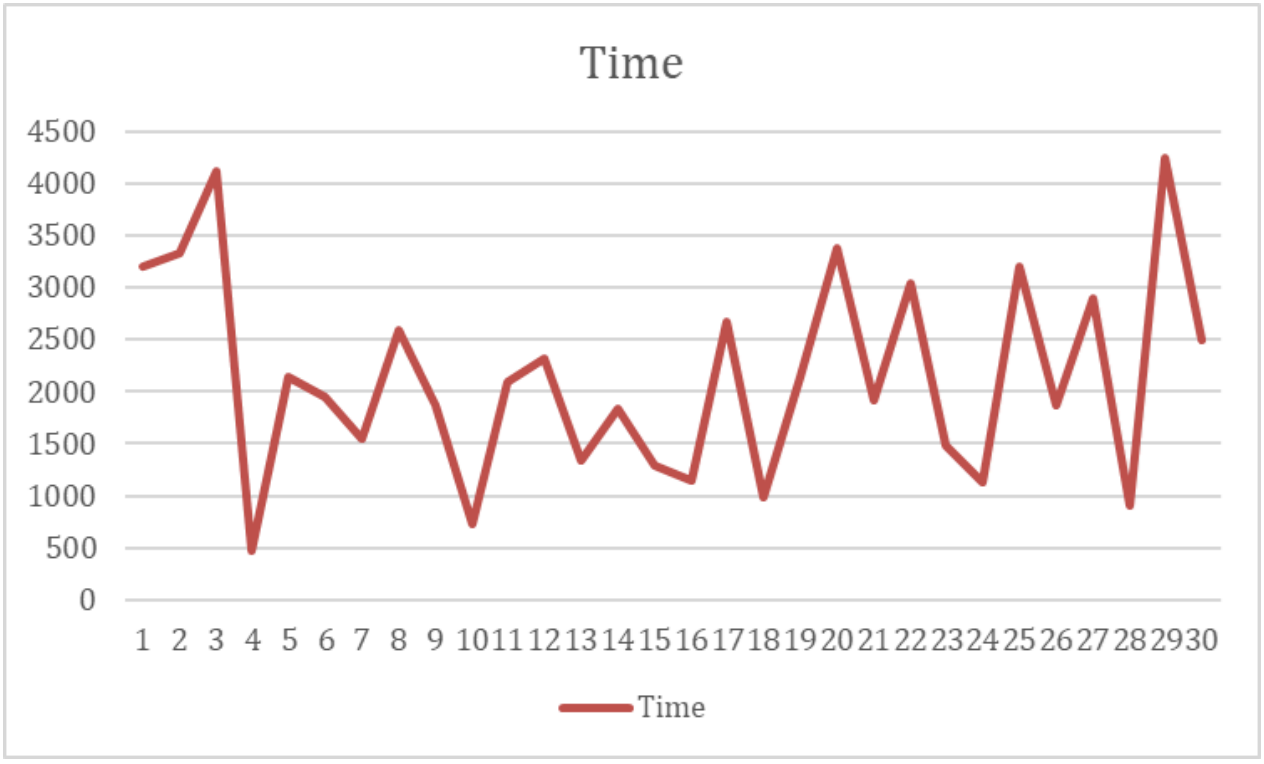

FIGURE 6: PROCESS ID VS TOTAL TIME TAKEN FOR EACH PROCESS

The above plot depicts the total time taken by each process to complete the task assigned to different VMs. The time taken to completely execute the task is plotted against the Process ID.

PSO SCHEDULER RESULTS

The PSO scheduler performance analysis was based on the Process IDs start time, end time, and make span time.

TABLE 2: RESULTS OF 30 PROCESSES FOR PSO ALGORITHM

Process ID	STATUS	Data Center ID	VM ID	Time	Start Time	Finish Time
2	SUCCESS	4	3	1228.44	0.1	1228.54
3	SUCCESS	5	5	1635.36	0.1	1635.46
1	SUCCESS	3	3	1917.84	0.1	1917.94
5	SUCCESS	6	6	1953.32	0.1	1953.42
9	SUCCESS	4	4	730.9	1228.54	1959.44
0	SUCCESS	2	2	2118.3	0.1	2118.4
6	SUCCESS	6	6	1554.43	1953.42	3507.85
11	SUCCESS	4	4	1903.16	1959.44	3862.6
7	SUCCESS	3	3	2171.8	1917.94	4089.74
8	SUCCESS	2	2	2132.76	2118.4	4251.15
10	SUCCESS	6	6	887.46	3507.85	4395.31
4	SUCCESS	5	5	3360.25	1635.46	4995.7

16	SUCCESS	3	3	1076.5	4089.74	5166.24
17	SUCCESS	3	3	309.69	5166.24	5475.92
15	SUCCESS	6	6	1139	4395.31	5534.32
12	SUCCESS	4	4	1998.46	3862.6	5861.06
13	SUCCESS	2	2	1819.08	4251.15	6070.23
21	SUCCESS	3	3	1039.38	5475.92	6515.31
19	SUCCESS	6	6	1097.02	5534.32	6631.33
22	SUCCESS	2	2	881.52	6070.23	6951.74
25	SUCCESS	6	6	959.83	6631.33	7591.16
14	SUCCESS	5	5	2767.51	4995.7	7763.22
27	SUCCESS	2	2	907.71	6951.74	7859.46
18	SUCCESS	4	4	2136.1	5861.06	7997.16
26	SUCCESS	6	6	619.57	7591.16	8210.73
20	SUCCESS	5	5	684.59	7763.22	8447.8
24	SUCCESS	3	3	2270.12	6515.31	8785.43
23	SUCCESS	4	4	1121.43	7997.16	9118.59
29	SUCCESS	5	5	1560.22	8447.8	10008.02
28	SUCCESS	6	6	1902.29	8210.73	10113.02

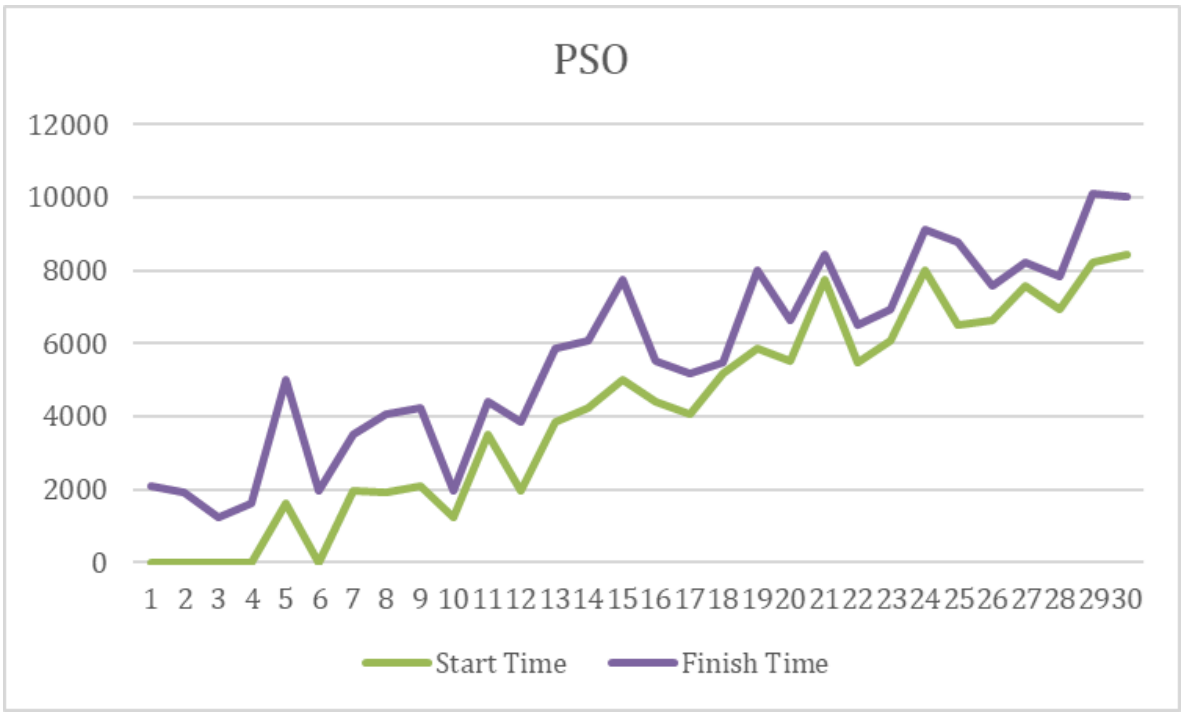

FIGURE 7 : PROCESS ID VS START TIME AND FINISH TIME

The above plot depicts the start time and end time by each Process to complete the task assigned to different VMs in PSO order.

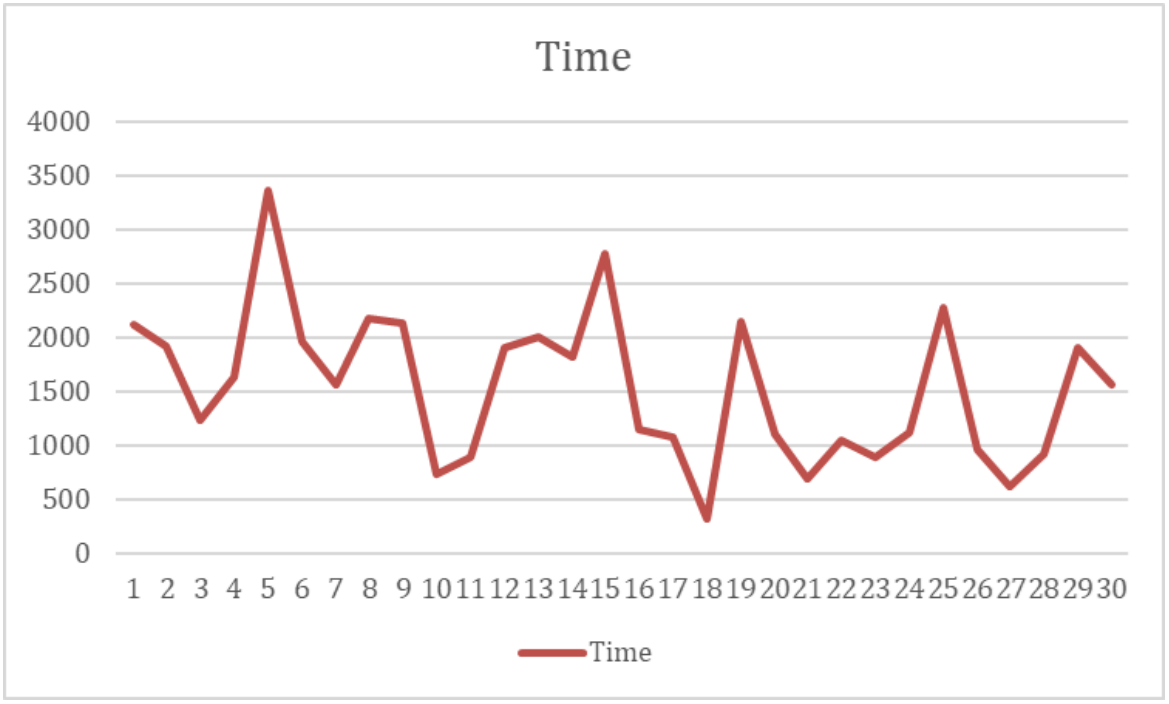

FIGURE 8 : PROCESS ID VS TOTAL TIME TAKEN FOR EACH PROCESS

The above plot depicts the total time taken by each process to complete the task assigned to different VMs. The time taken to completely execute the task is plotted against the Process ID.

ROUND ROBIN SCHEDULER RESULTS

The ROUND ROBIN scheduler performance analysis was based on the Process IDs start time, end time, and make span time.

TABLE 3: RESULTS OF 30 PROCESSES FOR RR ALGORITHM

Cloudlet ID	STATUS	Data Center ID	VM ID	Time	Start Time	Finish Time
3	SUCCESS	6	6	470.38	0.1	470.48
1	SUCCESS	5	5	1917.84	0.1	1917.94
0	SUCCESS	4	4	1960.44	0.1	1960.54
4	SUCCESS	3	3	2138.51	0.1	2138.61
5	SUCCESS	6	6	2776.87	470.48	3247.34
2	SUCCESS	2	2	4115.78	0.1	4115.88
9	SUCCESS	5	5	2348.68	1917.94	4266.62
7	SUCCESS	6	6	1744.94	3247.34	4992.29

6	SUCCESS	3	3	3169.31	2138.61	5307.92
8	SUCCESS	4	4	3367.98	1960.54	5328.52
10	SUCCESS	5	5	2097.9	4266.62	6364.52
15	SUCCESS	2	2	2764.45	4115.88	6880.34
13	SUCCESS	4	4	1840.01	5328.52	7168.53
12	SUCCESS	3	3	1998.46	5307.92	7306.38
18	SUCCESS	6	6	3388.43	4992.29	8380.72
22	SUCCESS	6	6	881.52	8380.72	9262.23
11	SUCCESS	5	5	3382.87	6364.52	9747.38
19	SUCCESS	2	2	3386.39	6880.34	10266.73
24	SUCCESS	3	3	3391.54	7306.38	10697.92
14	SUCCESS	4	4	3918.22	7168.53	11086.75
27	SUCCESS	5	5	1841.3	9747.38	11588.68
21	SUCCESS	2	2	2229.21	10266.73	12495.94
26	SUCCESS	3	3	1962.92	10697.92	12660.84
16	SUCCESS	4	4	2122.48	11086.75	13209.23
29	SUCCESS	2	2	1560.22	12495.94	14056.16
17	SUCCESS	4	4	981.83	13209.23	14191.06
28	SUCCESS	5	5	2676.32	11588.68	14265
20	SUCCESS	4	4	1922.57	14191.06	16113.63
23	SUCCESS	4	4	1823.3	16113.63	17936.93
25	SUCCESS	4	4	959.83	17936.93	18896.76

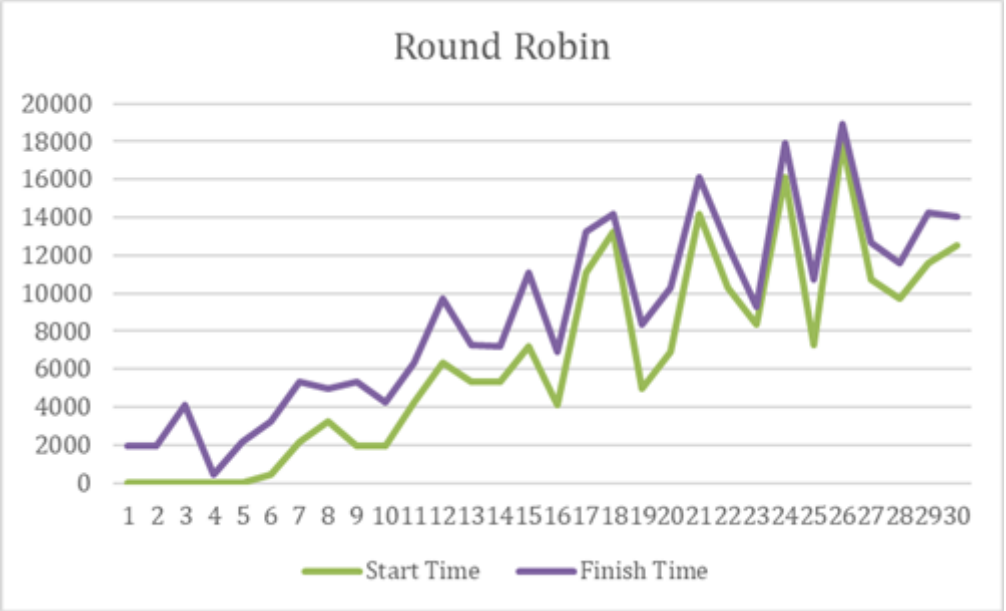

FIGURE 9 : PROCESS ID VS START TIME AND FINISH TIME

The above plot depicts the start time and end time by each Process to complete the task assigned to different VMs in RR order.

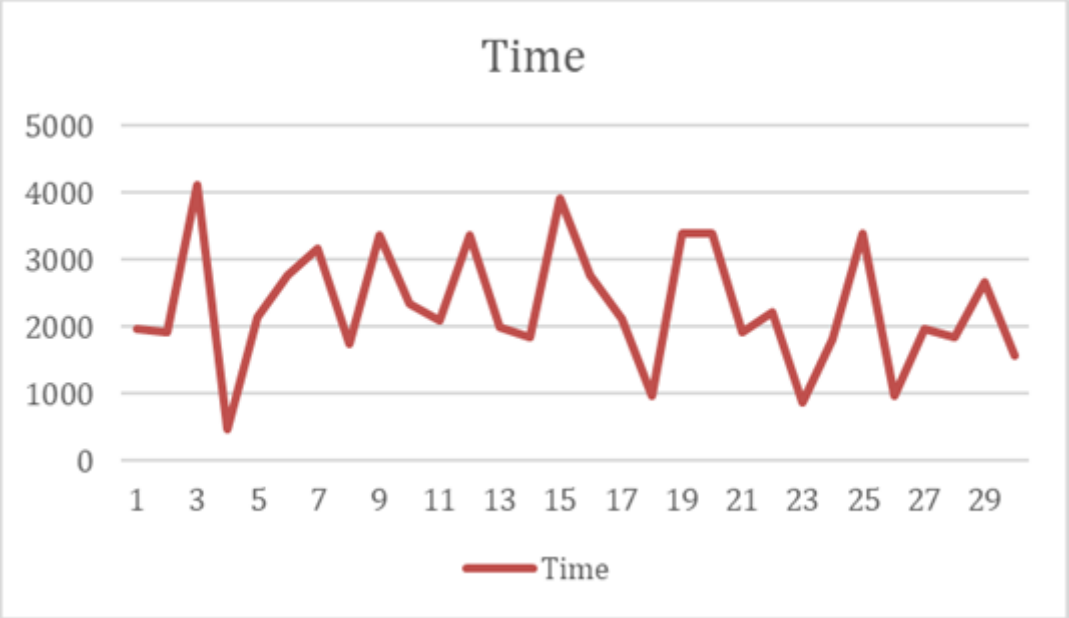

FIGURE 10 : PROCESS ID VS TOTAL TIME TAKEN FOR EACH PROCESS

The above plot depicts the total time taken by each process to complete the task assigned to different VMs. The time taken to completely execute the task is plotted against the Process ID.

5.4 SHORTEST JOB FIRST SCHEDULER RESULTS

The Shortest Job First scheduler performance analysis was based on the task ID's start time, end time, and make span time.

TABLE 4: RESULTS OF 30 PROCESSES FOR SJF ALGORITHM

Cloudlet ID	STATUS	Data Center ID	VM ID	Time	Start Time	Finish Time
6	SUCCESS	2	2	1838.26	0.1	1838.36
12	SUCCESS	3	3	1998.46	0.1	1998.56
0	SUCCESS	5	5	2782.9	0.1	2783
13	SUCCESS	2	2	1034.6	1838.36	2872.97
1	SUCCESS	6	6	3323.73	0.1	3323.83
3	SUCCESS	4	4	4257.71	0.1	4257.81
9	SUCCESS	5	5	2348.68	2783	5131.68
17	SUCCESS	2	2	2633.88	2872.97	5506.85
2	SUCCESS	6	6	2517.74	3323.83	5841.57
14	SUCCESS	3	3	4185.74	1998.56	6184.3
23	SUCCESS	5	5	1702.99	5131.68	6834.67
7	SUCCESS	4	4	2877.3	4257.81	7135.11
18	SUCCESS	2	2	2318.34	5506.85	7825.19
4	SUCCESS	6	6	2008.1	5841.57	7849.67
16	SUCCESS	3	3	2663.89	6184.3	8848.18
26	SUCCESS	5	5	2458.26	6834.67	9292.93
11	SUCCESS	4	4	2960.48	7135.11	10095.59
22	SUCCESS	3	3	1486.72	8848.18	10334.91
5	SUCCESS	6	6	2776.87	7849.67	10626.54
29	SUCCESS	5	5	1801.84	9292.93	11094.77
19	SUCCESS	2	2	3386.39	7825.19	11211.58
15	SUCCESS	4	4	1139	10095.59	11234.59
20	SUCCESS	2	2	684.59	11211.58	11896.17
8	SUCCESS	6	6	2132.76	10626.54	12759.3
10	SUCCESS	6	6	301.56	12759.3	13060.85
21	SUCCESS	4	4	3041.01	11234.59	14275.6

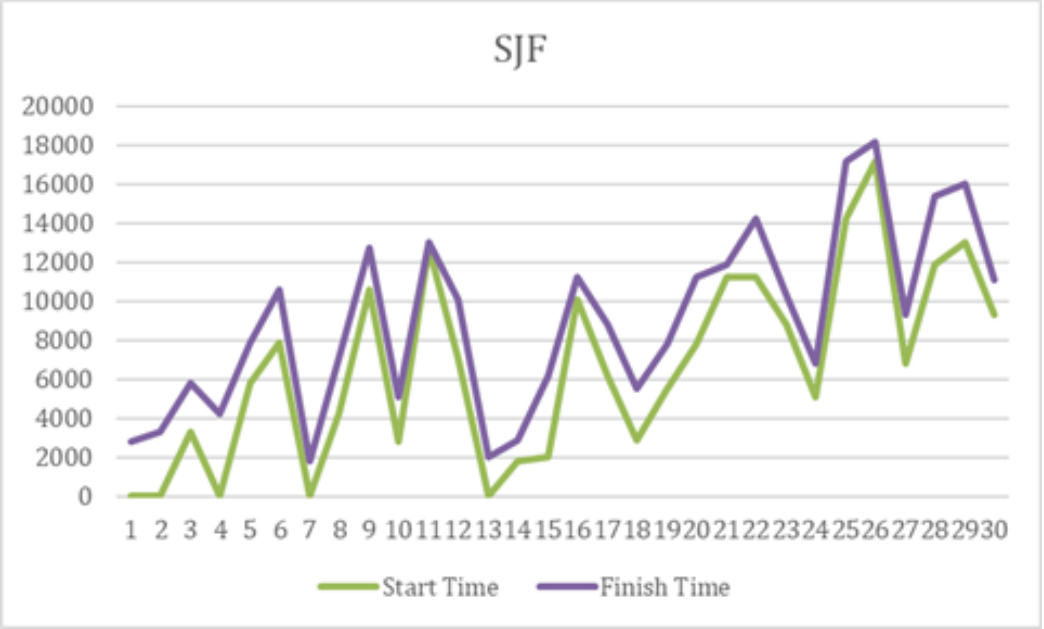

FIGURE 11 : PROCESS ID VS START TIME AND FINISH TIME

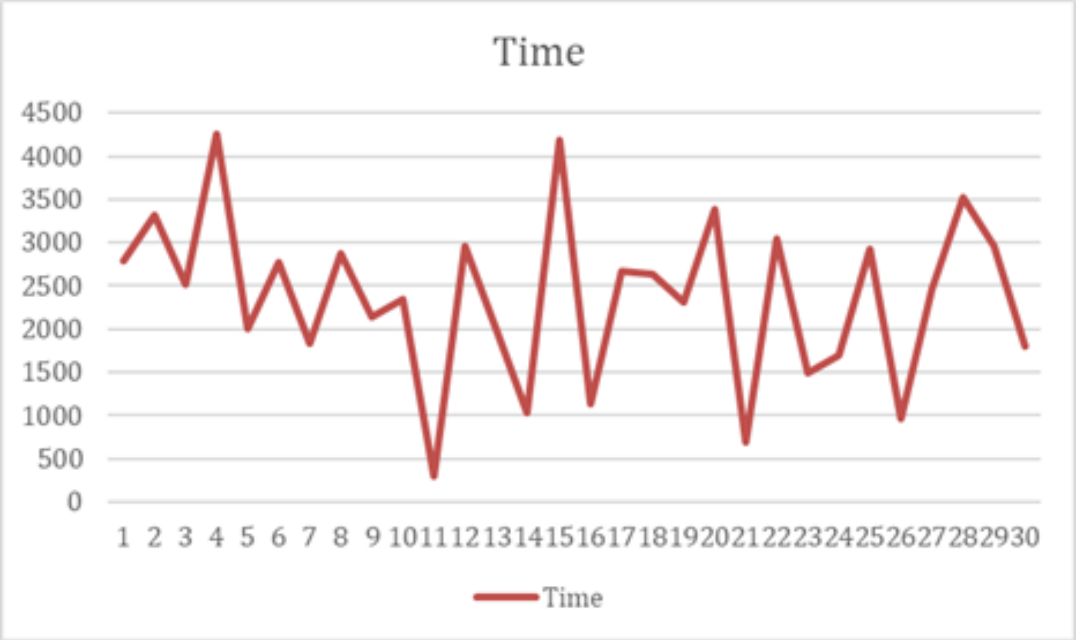

FIGURE 12 : PROCESS ID VS TOTAL TIME TAKEN FOR EACH PROCESS

5.5 PERFORMANCE ANALYSIS

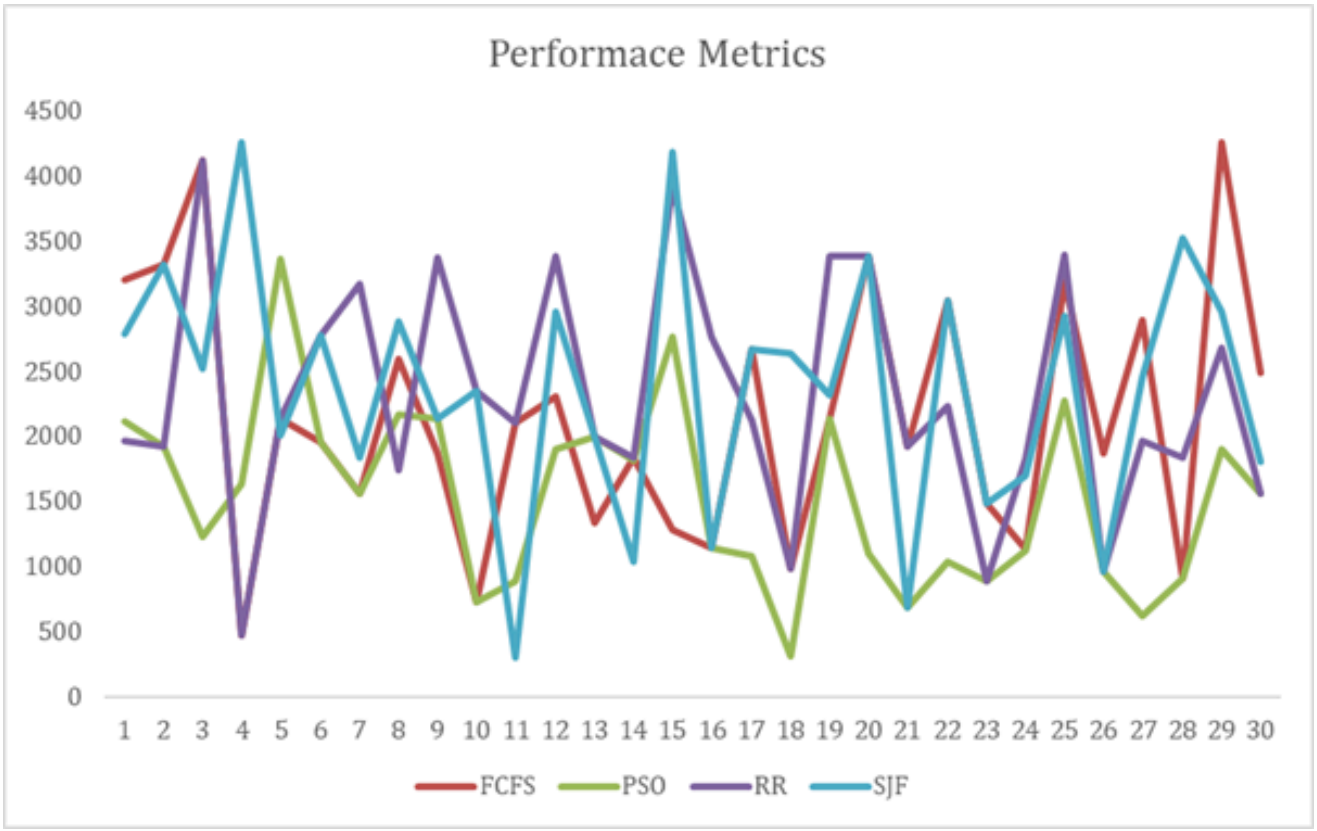

FIGURE 13 : Process ID vs Total Time Taken for each Process in all algorithm

The above graph and Results depicts the Total time Taken Variation among all the algorithm for different algorithms. As we can see the results we can compare and tell which algorithm is performing well when there is multi load on the VMs. To analyse it more let's look into more parameters

AVERAGE TIME TAKE

Comparison between the average Time Taken by all Algorithm(s). It is calculated by taking the average of Time taken from results of each Algorithm(s) .

TABLE 5: AVERAGE TIME TAKEN BY ALGORITHM(S)

Algorithm	Average Time Taken
FCFS	2144.082667
PSO	1529.468
RR	2304.683333
SJF	2367.481

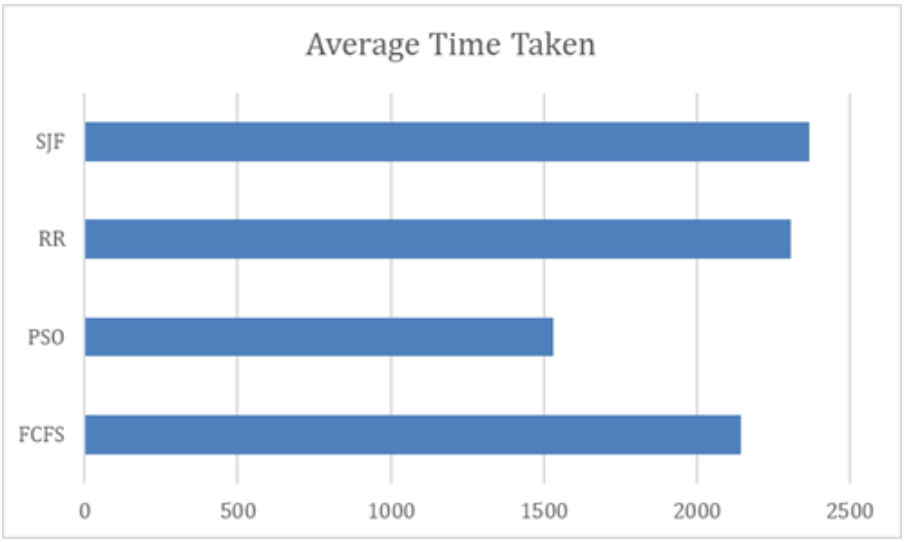

FIGURE 14 : AVERAGE TIME TAKEN VS ALGORITHM(S)

MAKE SPAN TIME

Make span is defined as the completion time of the last job to leave the system

TABLE 6: MAKE SPAN TIME OF ALGORITHM(S)

Algorithm	Make Span Time
FCFS	5225.6833
PSO	2521.23502
RR	4703.163581
SJF	4220.378558

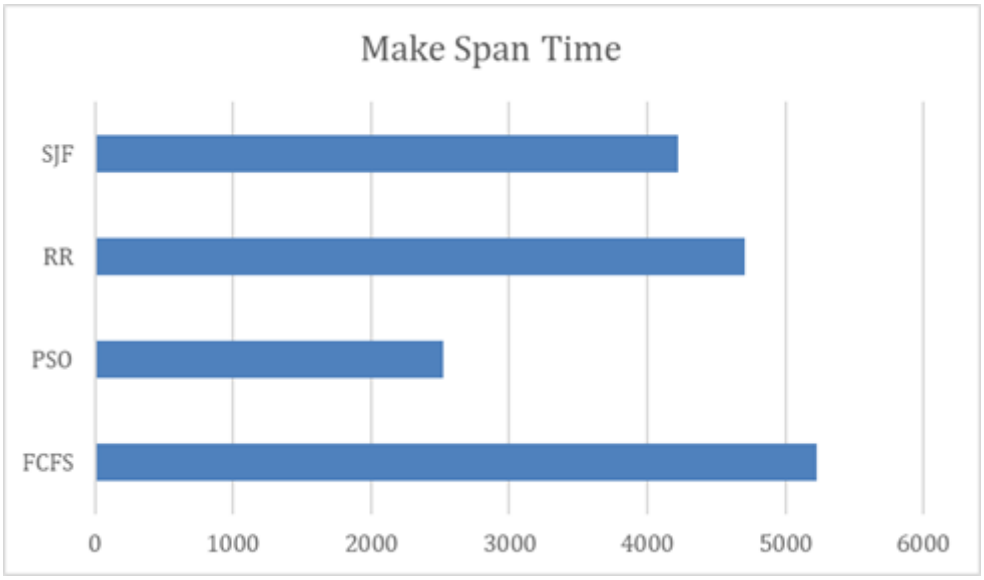

FIGURE 15 : Make Span Time vs Algorithm(s)

MINIMUM TIME TAKEN

TABLE 7: MINIMUM TIME TAKEN BY ALGORITHM(S) FOR A PROCESS

Algorithm	Min Time Taken
FCFS	470.38
PSO	309.69
RR	470.38
SJF	301.56

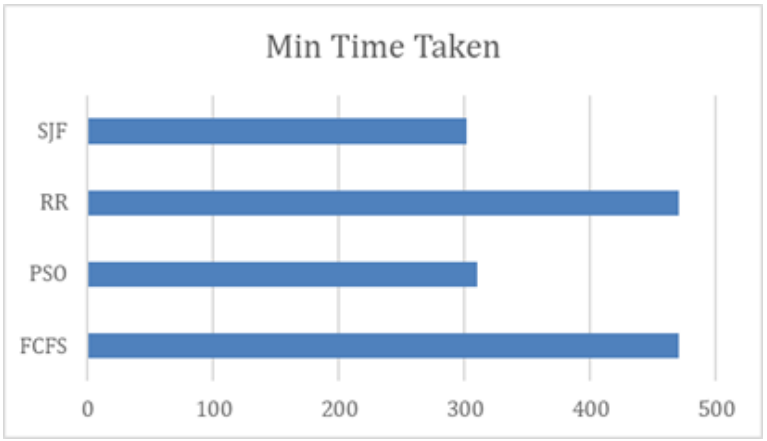

FIGURE 16 : MINIMUM TIME TAKEN VS ALGORITHM(S)

MAXIMUM TIME TAKEN

TABLE 8: MAXIMUM TIME TAKEN BY ALGORITHM(S) FOR A PROCESS

Algorithm	Max Time Taken
FCFS	4254.46
PSO	3360.25
RR	4115.78
SJF	4257.71

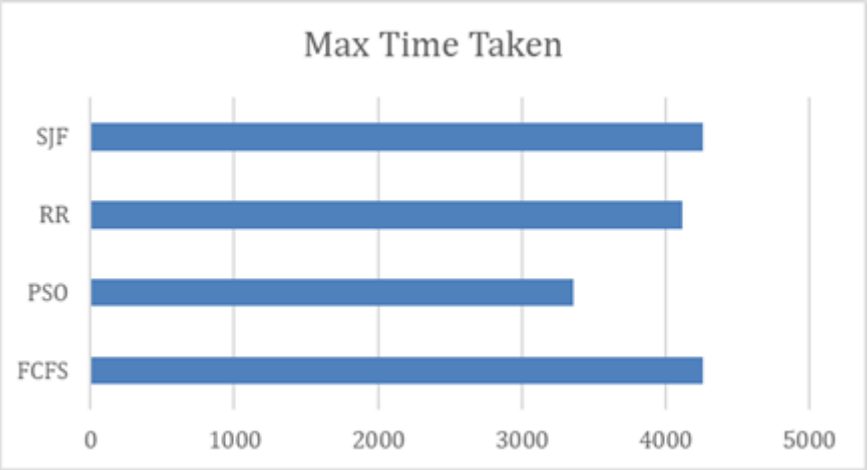

FIGURE 17: MAXIMUM TIME TAKEN VS ALGORITHM(S)

AVERAGE START TIME OF PROCESS FOR VM

TABLE 9: AVERAGE START TIME OF A PROCESS IN AN ALGORITHM(S)

Algorithm	Average Start Time
FCFS	6499.637
PSO	4137.742333
RR	6572.712667
SJF	6662.856667

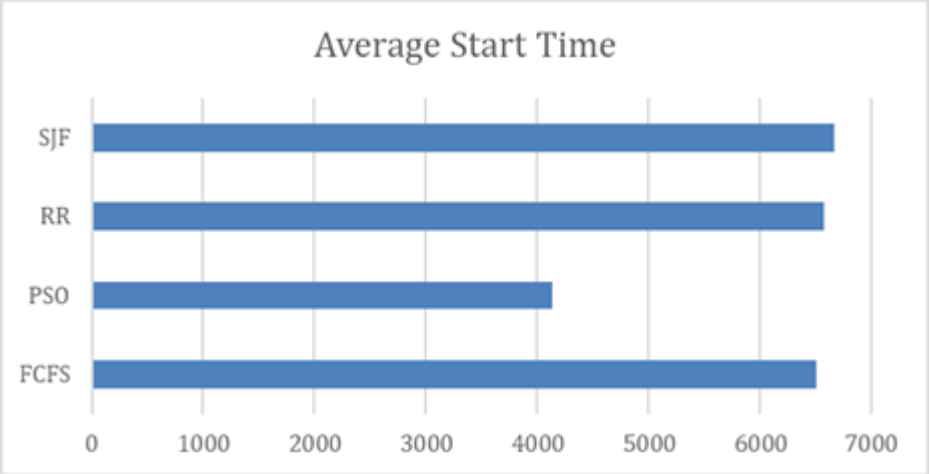

FIGURE 18 : Average Start Time vs Algorithm(s)

LAST FINISH TIME FOR A VM

TABLE 10: LAST FINISH TIME OF A PROCESS IN AN ALGORITHM(S)

Algorithm	**Last Finish Time of Process**
FCFS	22175.18
PSO	10113.02
RR	18896.76
SJF	18160.13

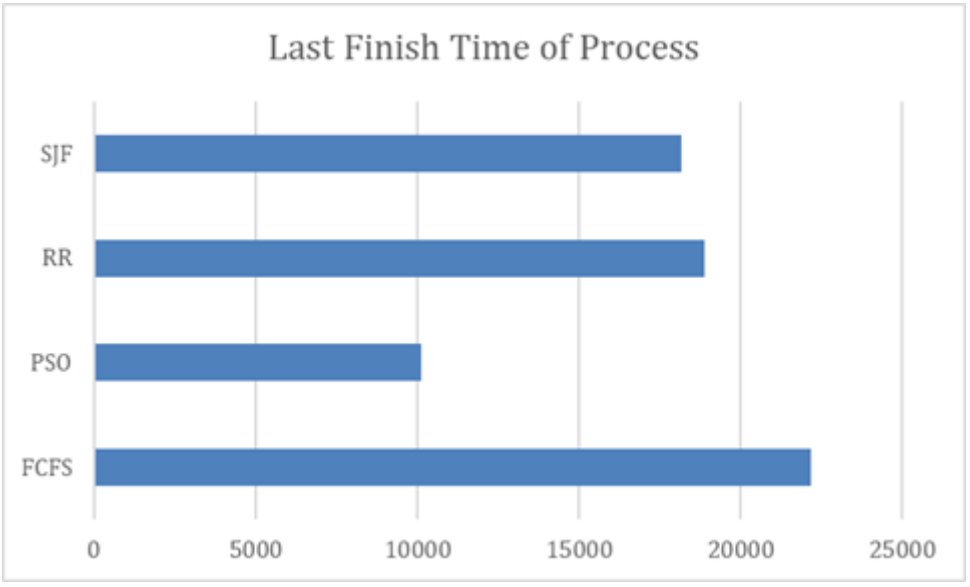

FIGURE 19 : LAST FINISH TIME VS ALGORITHM(S)

CHAPTER 6: CONCLUSIONS & FUTURE SCOPE

6.1 CONCLUSIONS

The report explores the complexities involved in developing a meta-heuristic scheduling strategy that is carefully crafted to tackle the complex issues that arise when coordinating the scheduling of IoT application tasks in the ever-changing fog environment. The importance of having a skilled scheduling algorithm in this complex ecosystem, where resources interact with cloud-based resources in a way that easily meets the diverse needs of contemporary life, cannot be emphasised. By means of a systematic assessment procedure carried out in a static setting and comprehensive comparison with a range of alternative algorithms, the empirical data unequivocally confirms the superiority of the Particle Swarm Optimisation (PSO) technique on a multitude of crucial metrics, such as speed, dependability, and flexibility. As a result, the PSO algorithm becomes not just the leader but the clear leader when it comes to scheduling IoT application tasks in fog conditions, establishing a new benchmark for effectiveness and efficiency in this emerging industry. Therefore, this study not only makes a substantial theoretical contribution to the field of fog computing, but it also provides useful insights that have the potential to drastically alter the deployment and management of Internet of Things applications in the near future.

6.2 FUTURE SCOPE

- The current implementation of the algorithm utilises VMs, but it is suggested that switching to containers would lead to better results. Containers offer more tangible benefits, such as faster boot times, compared to VMs.

- The set of rules's main downside is its reliance on static optimization techniques.

- The proposed algorithm may be more advantageous by using dynamic optimization strategies, which might make it extra suitable for business programs and improve the nice service (QoS) it offers.

- The suggestion is to enhance the PSO algorithm used for the scheduling optimization by transforming it into a weighted PSO, which is an advanced version of the standard PSO algorithm**.**

REFERENCES:

1) N. K. Uparosiya and A. Kumar, "Overcome Load Balancing Problem by Weight Based Scheme on Cloud," Journal of Data Acquisition and Processing, vol. 38, no. 2, pp. 393, 2023.
2) D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah and M. A. Alzain, "A Load Balancing Algorithm for the Data Centres to Optimize Cloud Computing Applications," in IEEE Access, vol. 9, pp. 41731-41744, 2021, doi: 10.1109/ACCESS.2021.3065308.
3) J. P. Mishra and S. R. Panda, "Load Balancing Using Cloudsim Simulator in Cloud Computing," International Journal of Scientific & Technology, vol. 9, no. 01, 2020.
4) S. P. Singh, et al., "Leveraging Energy-Efficient Load Balancing Algorithms in Fog Computing," Concurrency and Computation: Practice and Experience, vol. 34, no. 13, 2022
5) T.S. Nikoui, A. Balador, A. M. Rahmani and Z. Bakhshi, "Cost-awaretaskscheduling in fog-cloud environment", 2020
6) N. Thilagavathi, et al., "Energy Efficient Load Balancing in Cloud Data Center Using Clustering Technique," International Journal of Intelligent Information Technologies (IJIIT), vol. 15, no. 1, 2019, pp. 84-100.
7) S. Singh, A. Swaroop, A. Kumar and Anamika, "A survey on techniques to achive energy efficiency in cloud computing," 2016 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2016, pp. 1281-1285, doi: 10.1109/CCAA.2016.7813915.
8) K. Benchikh and L. Louail, "Task scheduling approaches for fog computing," 202130th Wireless and Optical Communications Conference (WOCC), 2021
9) Q.Liu, Y. Wei, S. Leng and Y. Chen, "Task scheduling in fog enabled Internet ofThings for smart cities," 2017 IEEE 17th International Conference on CommunicationTechnology (ICCT), 2017
10) K. De Donno, K. Tange and N. Dragoni, "Foundations and evolution of moderncomputing paradigms: Cloud iot edge and fog", IEEE Access, vol. 7, pp. 150 936-150948, 2019.
11) L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang and Y. Pan, "Stochastic Load Balancing for Virtual Resource Management in Datacenters," in IEEE Transactions on Cloud Computing, vol. 8, no. 2, pp. 459-472, 1 April-June 2020, doi: 10.1109/TCC.2016.2525984.
12) H. Shen and L. Chen, "A Resource Usage Intensity Aware Load Balancing Method for Virtual Machine Migration in Cloud Datacenters," in IEEE Transactions on Cloud Computing, vol. 8, no. 1, pp. 17-31, 1 Jan.-March 2020, doi: 10.1109/TCC.2017.2737628.
13) D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah and M. A. Alzain, "A Load Balancing Algorithm for the Data Centres to Optimize Cloud Computing Applications," in IEEE Access, vol. 9,

pp. 41731-41744, 2021, doi: 10.1109/ACCESS.2021.3065308.

14) M. Z. Nayyer et al., "LBRO: Load Balancing for Resource Optimization in Edge Computing," in IEEE Access, vol. 10, pp. 97439-97449, 2022, doi: 10.1109/ACCESS.2022.3205741.

15) M. Junaid et al., "Modeling an Optimized Approach for Load Balancing in Cloud," in IEEE Access, vol. 8, pp. 173208-173226, 2020, doi: 10.1109/ACCESS.2020.3024113.

16) S. Geng, D. Wu, P. Wang and X. Cai, "Many-Objective Cloud Task Scheduling," in IEEE Access, vol. 8, pp. 79079-79088, 2020, doi: 10.1109/ACCESS.2020.2990500.

17) B. Kruekaew and W. Kimpan, "Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm With Reinforcement Learning," in IEEE Access, vol. 10, pp. 17803-17818, 2022, doi: 10.1109/ACCESS.2022.3149955.

18) A. Razaque, N. R. Vennapusa, N. Soni, G. S. Janapati and k. R. Vangala, "Task scheduling in Cloud computing," 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 2016, pp. 1-5, doi: 10.1109/LISAT.2016.7494149.

19) P. Zhang and M. Zhou, "Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy," in IEEE Transactions on Automation Science and Engineering, vol. 15, no. 2, pp. 772-783, April 2018, doi: 10.1109/TASE.2017.2693688.

20) S. Selvarani and G. S. Sadhasivam, "Improved cost-based algorithm for task scheduling in cloud computing," 2010 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 2010, pp. 1-5, doi: 10.1109/ICCIC.2010.5705847.

21) A. Keivani and J. -R. Tapamo, "Task Scheduling in Cloud Computing: A Review," 2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), Winterton, South Africa, 2019, pp. 1-6, doi: 10.1109/ICABCD.2019.8851045.

22) Y. Li, M. Chen, W. Dai and M. Qiu, "Energy Optimization With Dynamic Task Scheduling Mobile Cloud Computing," in IEEE Systems Journal, vol. 11, no. 1, pp. 96-105, March 2017, doi: 10.1109/JSYST.2015.2442994.

23) S. Mittal and A. Katal, "An Optimized Task Scheduling Algorithm in Cloud Computing," 2016 IEEE 6th International Conference on Advanced Computing (IACC), Bhimavaram, India, 2016, pp. 197-202, doi: 10.1109/IACC.2016.45.

24) A. Keivani, F. Ghayoor and J. -R. Tapamo, "A review of recent methods of task scheduling in cloud computing," 2018 19th IEEE Mediterranean Electrotechnical Conference (MELECON), Marrakech, Morocco, 2018, pp. 104-109, doi: 10.1109/MELCON.2018.8379076.

25) S. Yin, P. Ke and L. Tao, "An improved genetic algorithm for task scheduling in cloud computing," 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), Wuhan, China, 2018, pp. 526-530, doi: 10.1109/ICIEA.2018.8397773.

201279

ORIGINALITY REPORT

12%	9%	5%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ir.juit.ac.in:8080 Internet Source	4%
2	Submitted to Jaypee University of Information Technology Student Paper	2%
3	link.springer.com Internet Source	<1%
4	dokumen.pub Internet Source	<1%
5	www.ivoryresearch.com Internet Source	<1%
6	medium.com Internet Source	<1%
7	5dok.net Internet Source	<1%
8	www.ir.juit.ac.in:8080 Internet Source	<1%
9	Yogita Yashveer Raghav, Vaibahv Vyas. "chapter 10 Load Balancing Using Swarm	<1%

	Intelligence in Cloud Environment for Sustainable Development", IGI Global, 2023 Publication	
10	"Recent Trends in Electronics and Communication", Springer Science and Business Media LLC, 2022 Publication	<1 %
11	Yogesh Lohumi, Durgaprasad Gangodkar, Prakash Srivastava, Mohammad Zubair Khan, Abdulrahman Alahmadia, Ahmed H. Alahmadia. "Load Balancing in cloud Environment: A State of-the-Art Review", IEEE Access, 2023 Publication	<1 %
12	standards.iteh.ai Internet Source	<1 %
13	Henry Chima Ukwuoma, Gabriel Arome, Aderonke Thompson, Boniface Kayode Alese. "Post-quantum cryptography-driven security framework for cloud computing", Open Computer Science, 2022 Publication	<1 %
14	Submitted to University of East London Student Paper	<1 %
15	Submitted to University of Florida Student Paper	<1 %
	isteonline.in	

16	Internet Source	<1 %
17	C Gagné, W L Price, M Gravel. "Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence-dependent setup times", Journal of the Operational Research Society, 2017 Publication	<1 %
18	Sushama Annaso Shirke, Naveenkumar Jayakumar, Suhas Patil. "Design and performance analysis of modern computational storage devices: A systematic review", Expert Systems with Applications, 2024 Publication	<1 %
19	idr.l2.nitk.ac.in Internet Source	<1 %
20	patents.google.com Internet Source	<1 %
21	www.igi-global.com Internet Source	<1 %
22	"Intelligent Computing and Optimization", Springer Science and Business Media LLC, 2023 Publication	<1 %

23	Yellamma Pachipala, Kavya Sri Sureddy, A.B.S. Sriya Kaitepalli, Nagalakshmi Pagadala, Sai Satwik Nalabothu, Mihir Iniganti. "Optimizing Task Scheduling in Cloud Computing: An Enhanced Shortest Job First Algorithm", Procedia Computer Science, 2024 Publication	<1 %
24	eudl.eu Internet Source	<1 %
25	ojs.sgsci.org Internet Source	<1 %
26	Muhammad Sohaib Ajmal, Zeshan Iqbal, Muhammad Basit Umair, Muhammad Shahzad Arif. "Flexible Genetic Algorithm Operators for Task Scheduling in Cloud Datacenters", 2020 14th International Conference on Open Source Systems and Technologies (ICOSST), 2020 Publication	<1 %
27	Pranav R Hegde, Saileshwar Karthik, Shashank Udyavar Madan, Sumukh Raju Bhat, Sudarshan Tsb. "QLEN: A Load Distribution Algorithm to Improve QoS Factors Among Fog Devices", 2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3), 2023 Publication	<1 %

28	beei.org Internet Source	<1%
29	export.arxiv.org Internet Source	<1%
30	www.alhenshiri.net Internet Source	<1%
31	www.ijisae.org Internet Source	<1%
32	"Advances on Broadband and Wireless Computing, Communication and Applications", Springer Science and Business Media LLC, 2019 Publication	<1%
33	Darakhshan Syed, Ghulam Muhammad Shaikh, Hani Alshahrani, Mohammed Hamdi, Mohammad Alsulami, Asadullah Shaikh, Syed Rizwan. "A Comparative Analysis of Metaheuristic Techniques for High Availability Systems (September 2023)", IEEE Access, 2024 Publication	<1%
34	Garima Verma. "Hybrid Optimization Model for Secure Task Scheduling in Cloud: Combining Seagull and Black Widow Optimization", Cybernetics and Systems, 2022 Publication	<1%

35	Md. Owais Qurani, Ravinder Singh. "Load Balancing for Virtual Resources Management in Data Center", 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2018 Publication	<1 %
36	Nupur Jangu, Zahid Raza. "Chapter 5 JAYA-Based Task Scheduling Algorithm in Fog-Cloud Environment", Springer Science and Business Media LLC, 2023 Publication	<1 %
37	Zulfiqar Ali Khan, Izzatdin Abdul Aziz, Nurul Aida Bt Osman, Israr Ullah. "A Review on Task Scheduling Techniques in Cloud and Fog Computing: Taxonomy, Tools, Open Issues, Challenges, and Future Directions", IEEE Access, 2023 Publication	<1 %
38	dyuthi.cusat.ac.in Internet Source	<1 %
39	ijisae.org Internet Source	<1 %
40	iris.unige.it Internet Source	<1 %
41	web.archive.org Internet Source	<1 %

42	www.publishingindia.com Internet Source	<1 %
43	www.sci-hub.st Internet Source	<1 %
44	www.techscience.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 8 words
Exclude bibliography	On		