University of Cabuyao
(PAMANTASAN NG CABUYAO)

COLLEGE OF COMPUTING STUDIES

ENHANCED PSO AND ACO FOR CLOUD LOAD BALANCING: A COMPARATIVE STUDY

A Thesis Requirement

Submitted to the Faculty of

The College of Computing Studies

UNIVERSITY OF CABUYAO

City of Cabuyao, Laguna

In Partial Fulfillment

Of the Requirements for the Degree

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

By:

Laranga, John Danmel C.

Reyes, Kier Christian F.

Violanta, Jan Alfred G.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER I
THE PROBLEM AND ITS SETTING

Introduction

As organizations increasingly migrate from traditional on-premises infrastructures to cloud-based environments, managing dynamic and unpredictable workloads has become a critical challenge. Effective load balancing is one of the main issues in cloud computing since it guarantees the best work distribution across virtual machines (VMs), directly influencing system reaction time, resource use, energy efficiency, degree of imbalance, and makespan.

Heuristic-based solutions including particle swarm optimization (PSO) and ant colony optimization (ACO) have attracted interest due to their adaptability in order to solve the constraints of conventional methodologies. These algorithms dynamically allocate workloads based on multiple performance parameters, demonstrating promising improvements in performance metrics such as response time, resource utilization, energy efficiency, degree of imbalance, and Makespan [1], [2], [3].

Despite their potential, studies comparing these algorithms—especially in enhanced and adaptive forms—are limited in simulation environments that mirror real-world workload behavior. This study proposes a comparative analysis between Enhanced PSO, which employs Adaptive Velocity Clamping [1], and Enhanced ACO, which integrates Pheromone Update [4], to evaluate their effectiveness in cloud load balancing.

A custom simulation system is developed using CloudSim, a widely used cloud simulation toolkit, to model realistic workload scenarios. These workloads are derived from a preprocessed subset of the Google Cluster Dataset, ensuring authenticity and practical relevance. The system collects data based on five critical performance metrics: response time, resource utilization, energy efficiency, degree of imbalance, and Makespan. For a comprehensive evaluation, the results are analyzed and visualized using MATLAB.

The study focuses on enhanced versions of two widely recognized dynamic algorithms to offer empirically grounded insights that aid in the selection of efficient scheduling strategies in cloud environments. The findings serve both academic and practical interests, guiding cloud infrastructure design toward more adaptive, sustainable, and performance-optimized solutions.

Statement of the Problem

Efficient load balancing in cloud computing remains a critical challenge due to fluctuating workloads and constrained computational resources [1]. While heuristic algorithms like Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) [2]. To address this, enhanced variants—Enhanced PSO (EPSO) with adaptive velocity clamping and Enhanced ACO (EACO) with adaptive pheromone evaporation [4]—have been developed to improve task scheduling by optimizing response time, resource utilization, energy efficiency, degree of imbalance, and Makespan. However, direct comparative studies of these enhanced algorithms in real cloud environments are limited. This study aims to fill that gap by using CloudSim to simulate diverse workloads from the Google Cluster Traces and evaluate the performance of EPSO and EACO. The goal is to identify which enhanced algorithm delivers superior results under varying conditions, contributing to more efficient, scalable, and sustainable cloud load-balancing strategies.

Specific Statement of the Problem

Concerning the literature reviewed, this study seeks to answer the following questions:

1. How can a CloudSim-based simulation framework be designed to evaluate load-balancing strategies by

    1.1. configuring virtual machines and task scheduling parameters;

    1.2. implementing EPSO and EACO within the simulation; and

    1.3. integrating Google cluster traces as workload inputs for realistic testing?

2. How can PSO load-balancing algorithm be developed and enhanced in CloudSim by applying

    2.1. adaptive velocity clamping; and

    2.2. nonlinear inertia weight reduction?

3. How can ACO load-balancing algorithm be developed and enhanced in CloudSim by applying

    3.1. adaptive pheromone evaporation; and

    3.2. heuristic load-based reinforcement?

4. Is there a significant difference between EPSO and EACO in terms of

    4.1. response time;

    4.2. resource utilization;

4.3. energy efficiency;

4.4. degree of imbalance; and

4.5. makespan?

5. What are the assessment of end users regarding the proposed system in terms of

5.1. functional suitability;

5.2. interaction capability;

5.3. performance efficiency; and

5.4. learnability?

6. What are the assessment of IT Experts regarding the proposed system in terms of

6.1. functional suitability;

6.2. interaction capability;

6.3. accuracy of results; and

6.4. scalability?

Hypotheses

The researchers formulated the following hypotheses to validate the efficacy of EPSO and EACO.

H_o: There is no significant difference between EPSO and EACO regarding response time, resource utilization, energy efficiency, degree of imbalance, and Makespan.

H_1: There is a significant difference between EPSO and EACO regarding response time, resource utilization, energy efficiency, degree of imbalance, and Makespan.

Scope and Limitation

The study focuses on developing a simulation tool that analyzes task scheduling algorithms, specifically Enhanced Particle Swarm Optimization (EPSO) and Enhanced Ant Colony Optimization (EACO), across varying workloads. It aims to support cloud specialists and IT professionals based in the southeastern part of Metro Manila and Cabuyao, where the study will be conducted. The development of the system follows the SCRUM methodology, enabling progressive enhancement of the CloudSim-based simulation through continuous testing and refinement with the aid of sprints. To effectively manage the project, a time-boxed method will be employed to ensure that tasks are completed within fixed periods, maintaining consistent development pacing and allowing for timely deliverables. Key activities include preprocessing Google Cluster Data, implementing dynamic task scheduling algorithms, constructing essential simulation components, and evaluating system performance per iteration. The simulation tool allows users to input actual workloads or choose from preprocessed datasets, configure cloud infrastructure parameters, and select scheduling algorithms for analyzing workload distribution. It provides graphical visualizations generated via MATLAB scripting, focusing on response time, resource utilization, energy efficiency, degree of imbalance, and

Makespan. These outputs are intended for cloud professionals, academic researchers, and decision-makers seeking comparative performance insights between EPSO and EACO, aiding in informed decisions for optimizing cloud deployment.

However, the study may face limitations: CloudSim's inability to emulate physical network infrastructure and real-time latency affects generalizability. The network modeling in CloudSim supports message-passing and basic topologies; however, it may not accurately simulate the dynamic nature of real-world cloud network traffic, such as congestion, variable latency, and protocol overheads - which might impact the generalizability of the study [5]. This adds to the reason researchers did not include traffic models and network simulation in the study and to ensure our focus on algorithm performance remains intact. Additionally, while realistic, The Google Cluster Dataset may not capture all workload variations; focusing solely on stakeholders from southeastern Metro Manila and Cabuyao may limit diverse perspectives. Future research may benefit from incorporating live cloud environments, more heterogeneous datasets, expanded energy measurement, and broader expert consultation for improved practical relevance.

Significance of the Study

This research holds significant value in the field of cloud computing as it addresses the limited comparative analysis between enhanced load-balancing algorithms, specifically Enhanced Particle Swarm Optimization (EPSO) and Enhanced Ant Colony Optimization (EACO). While many studies focus on traditional versus enhanced models, few have directly compared two advanced heuristic algorithms under identical conditions. In order to help stakeholders select the most effective dynamic task scheduling strategies, this study aims to bridge the gap by offering new insights into how such algorithms function in actual cloud systems.

By leveraging a simulation tool built on CloudSim and using real-world data from the Google Cluster Dataset, the research evaluates both algorithms based on critical performance metrics: response time, resource utilization, energy efficiency, degree of imbalance, and Makespan. The study ensures the analysis captures both end-user experience and infrastructure-level effectiveness, offering a holistic understanding of algorithmic impact in cloud operations.

Cloud Specialists benefit from actionable data to guide the selection of efficient load-balancing solutions. Through EPSO and EACO, they gain options that can reduce delays during peak usage, enhance resource use, and minimize power consumption—all while maintaining SLA compliance.

IT Experts and Infrastructure Managers can apply the study's findings to improve virtual machine placement and reduce computational waste, resulting in more cost-effective and responsive cloud environments.

Academic and Research Communities gain a foundation for future investigations into heuristic-based and hybrid algorithms. This work promotes the exploration of integrating machine learning with enhanced optimization models, offering a new layer of intelligence in load-balancing.

Ultimately, this study enriches the understanding of load-balancing strategies by directly comparing two advanced algorithms, offering guidance to cloud professionals and contributing to the development of scalable, efficient, and sustainable cloud infrastructures.

CHAPTER II
REVIEW OF LITERATURE AND STUDIES

Introduction

This chapter outlines the study's conceptual and theoretical foundations, including relevant literature, related studies, and the conceptual framework. It discusses the fundamental principles guiding the development of the proposed application, reviews existing research and methodologies, and identifies key theories that support the study. It also presents the synthesis of findings, highlights research gaps, and defines essential terms used throughout the research.

Conceptual Literature

In cloud computing, efficient load-balancing is critical for optimizing performance metrics such as resource utilization, response time, energy efficiency, degree of imbalance, and Makespan. This study focuses on two heuristic-based, adaptive algorithms: Enhanced Particle Swarm Optimization (EPSO) and Enhanced Ant Colony Optimization (EACO). EPSO is employed for energy-aware task scheduling, optimizing task-to-VM assignments without VM migration framework by leveraging adaptive velocity clamping inspired by PSO-SAVL. EACO, on the other hand, utilizes pheromone-based task allocation to dynamically distribute workloads, ensuring efficient resource use under variable conditions. Both algorithms aim to enhance cloud performance through adaptive, heuristic-driven approaches tailored to dynamic environments.

This iteration-based approach, inspired by adaptive strategies in PSO-based studies, including PSO-SAVL by Li et al. [6], balances exploration and exploitation, improving task scheduling convergence. An energy model from Zhang and Li is adopted as a contextual reference to measure energy efficiency - one of the metrics in our study to be evaluated [7]. EPSO assigns tasks to VMs using PSO's velocity and position update formulas, targeting optimal performance across the specified metrics. EACO simulates pheromone-based task allocation in parallel to achieve robust workload distribution under dynamic loads [4].

The simulation tool must incorporate essential features: input for uploading preprocessed Google Cluster Data or using users' format-aligned workload traces, configuration for cloud parameters (such as virtual machines (VMs) and data centers), and selection of the Enhanced Ant Colony Optimization (EACO) and Enhanced Particle Swarm Optimization (EPSO) algorithms. Performance analysis will be conducted using MATLAB graphs to analyze key metrics: energy efficiency, resource utilization, and response time [8]. Development must consider CloudSim's constraints, such as its reduced network modeling affecting live migration accuracy and the possible bias of the Google Cluster Data, which may not cover all workload patterns [9]. Although network energy might be excluded, the energy-aware design must prioritize server power minimization, which calls for future development [7]. Auxiliary libraries researchers find beneficial throughout development include MATLAB, a computational tool for numerical computations, particularly linear algebra (matrices). Initially meant to give interactive access to the Linpack and Eispack libraries, MATLAB has since evolved into a comprehensive tool for communication, engineering, research, programming, and visualization. MATLAB's modern algorithms, excellent data management capabilities, and strong programming tools make it invaluable for plotting simulation results, particularly in cloud computing simulations for comparing load-balancing algorithms [4].

CloudSim will be the system's core for simulating cloud environments, considering that algorithms like Enhanced ACO and EPSO will be scheduled and simulated. CloudSim is an economical tool for evaluating and analyzing the functioning of cloud components under various conditions and workloads, aiding in product evaluation and issue prioritization [9]. Python's extensive standard libraries, which facilitate the creation of machine learning models, online services, data mining, and classification tasks, make it a popular choice for data science and analytics. Python's libraries will also be essential for data preprocessing, which transforms datasets before they are sent into CloudSim for cloud simulation execution [10]. For the simulation system interface, ReactJS—an open-source technology for building UIs specifically for single-page applications—will be used. ReactJS allows the creation of large-scale web applications that can dynamically consume data and update in real time without refreshing the page [11]. Its flexible design makes it ideal for visualizing the performance metrics of EACO and EPSO algorithms.

User-centric design ensures that the simulation tool is tailored to the needs of cloud specialists and IT experts. The React.js frontend will offer intuitive visualizations, such as interactive graphs displaying energy efficiency, resource utilization, and response time metrics. It will also provide user-friendly controls for configuring simulations, allowing users to interpret the results effectively. Testing will validate the performance of the Enhanced Ant Colony Optimization (ACO) and Enhanced Particle Swarm Optimization (EPSO) algorithms under varied workloads, focusing on energy efficiency, resource utilization, and response time. Additionally, user feedback will be collected to enhance the tool's usability and practical applicability in real-world scenarios [9].

Research Literature

Review of Related Literature

The study of Zhang and Li [7] proposes an Enhanced PSO (EPSO) algorithm integrated with virtual machine (VM) live migration to optimize resource allocation in cloud-based IoT systems. IoT environments, with their vast array of connected devices, require efficient scheduling to minimize energy use and execution delays. By combining PSO's swarm intelligence with VM migration, the system dynamically balanced workloads, preventing server overloads and underloads. This approach, implemented in CloudSim, reduces energy consumption by 10% compared to particle crowding and over 8% compared to standard PSO scheduling. Execution time is also reduced by 18% versus particle swarm scheduling and 8% versus PSO. These improvements highlight how enhanced PSO can optimize resource efficiency in dynamic, energy-sensitive IoT clouds, making it an ideal candidate for load-balancing in such systems.

In addition, other methods, such as the hybrid load balancer 'ELBABCE bHC,' blend artificial bee colonies (ABC) with an enhanced b-Hill climbing algorithm to address key performance factors such as response time, processing cost, and resource utilization. By combining ABC's global search capabilities with the local optimization of the β-Hill climbing algorithm, this hybrid approach ensures load balance across servers, resulting in improved performance. In CloudAnalyst testing, 'ELBABCE bHC' outperforms traditional algorithms like throttled load-balancing and active monitoring load-balancing, proving to be a strong solution for enhancing cloud performance [12].

Load-balancing is a critical challenge in cloud computing due to its complex nature, affecting Quality of Service (QoS), performance, and compliance with service-level agreements (SLA) [13]. As an NP-hard problem, load-balancing requires efficient solutions to map workloads to resources. Metaheuristic approaches, such as PSO, have been widely adopted to address these challenges. PSO, in particular, has been shown to improve key performance indicators such as Makespan, response time, and resource utilization. The comparative analysis of metaheuristic methods highlights PSO's effectiveness in achieving better performance metrics compared to other load-balancing strategies, reinforcing its suitability for dynamic cloud load-balancing solutions.

In their study, Tirmazi et al. analyzed Google's Borg cluster management system using a new dataset from eight clusters in May 2019. Their study explores the evolution of workload scheduling since 2011, highlighting key shifts such as adopting alloc sets and job dependencies and focusing on best-effort batch jobs. They find that the top 1% of jobs consume over 99% of resources, emphasizing the need for adaptive scheduling. The study compares the 2019 trace with the previous one, showcasing workload and resource utilization changes. This research lays the groundwork for improved load-balancing algorithms and offers valuable insights for scheduler design in cloud computing [14].

The study by Zhang and Li [7], which explored the integration of virtual machine (VM) migration with Particle Swarm Optimization (PSO) to optimize energy consumption in cloud computing environments, serves as a key inspiration for our research. Their approach highlighted the effectiveness of combining VM migration with PSO to achieve dynamic workload balancing and energy savings. Building on this foundation, our work adapts and extends these concepts by incorporating advanced strategies into Enhanced PSO (EPSO) and Adaptive Ant Colony Optimization (AACO). Unlike Zhang and Li's reliance on VM migration, our study introduces adaptive mechanisms—such as velocity clamping in EPSO and pheromone-guided task allocation in AACO—to manage task scheduling without VM migration dynamically.

Review of Related Studies

Many studies in cloud load-balancing utilize simulation platforms such as CloudSim and CloudAnalyst to evaluate the performance of scheduling algorithms. These tools enable a controlled environment for assessing metrics such as response time, CPU and memory utilization, and energy consumption. However, Kshama S. B. and S. R. [15] went one step further by utilizing Amazon Web Services (AWS) to integrate simulation with real-world deployment. Their dual-method methodology validated the simulation results against real-world settings, including system configurations and network delay. While this method offers a more robust evaluation, the present study is limited to simulation due to restricted access to cloud platforms and time constraints.

While useful for experimentation, Zhang and Li highlighted that simulation tools cannot fully replicate the physical complexity of cloud systems, particularly regarding live migration latency and hardware-level power consumption. Despite these drawbacks, simulation is a proper and generally recognized technique for initial performance analysis [7].

In the study, *"A Systematic Analysis of Load Balance in the Cloud,"* published in the *International Journal of Computer Sciences and Engineering*, Vijaykumar, and Chandre explored the efficacy of load-balancing algorithms in cloud computing, a topic directly pertinent to research on heuristic-based load-balancing techniques [16]. The authors conducted a systematic review of peer-reviewed studies from 2010 to 2018, examining how algorithms enhance key performance metrics such as resource utilization, response time, throughput, and energy efficiency. Their methodology involved categorizing algorithms into static and dynamic types, with a detailed analysis revealing that dynamic, heuristic-based approaches—such as variants of Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO)—outperformed static methods. Specifically, these algorithms improved resource utilization by up to 20% and reduced response times by 15–25% under varying workloads. The study also underscored the promise of hybrid algorithms and machine learning-based predictive models to enhance adaptability in load-balancing. The study provides a complete background for assessing dynamic approaches. Our study of metaheuristic-based algorithms such as Enhanced PSO (EPSO) and Enhanced ACO (EACO) is quite essential for job scheduling in cloud environments. Moreover, it points to possible future directions of research, such as combining predictive models with scheduling systems to control workload variations actively, thereby enhancing the breadth and focus of our work.

Satyanarayana Nimmala et al. proposed a novel task-scheduling algorithm for cloud computing, focusing on optimizing performance under dynamic workloads [17]. The research aimed to reduce Makespan and improve energy efficiency, achieving a 15% reduction in Makespan and a 10% improvement in energy efficiency compared to traditional heuristic methods. The study employed a simulation-based research design, using CloudSim to model a cloud environment with 100 VMs and varying task arrival rates. Procedures involved initializing the simulation with synthetic workloads modeled after real-world traces, such as the Google Cluster Data, and evaluating the algorithm's performance over 1000 iterations. Data gathering relied on simulation logs, capturing metrics like response time, resource utilization, energy efficiency, degree of imbalance, and Makespan, with no primary data collection instruments. The population consisted of simulated tasks with characteristics based on cloud workload patterns, sampled purposely to cover a range of task sizes and arrival rates [17]. Variables operationally defined included the scheduling algorithm (independent) and performance metrics (dependent, e.g., Makespan, energy efficiency). Extraneous variables, such as network latency and VM heterogeneity, were controlled by assuming ideal conditions in the simulation. The study recommended further research into integrating real-time data for more accurate predictions and exploring the algorithm's scalability in larger, heterogeneous cloud setups. This aligns with our focus on EPSO and EACO's adaptability to dynamic environments. However, part of our study's limitation is handling traffic models or network-simulation variables.

Patle and Sahu [18] introduced a hybrid optimization approach combining Firefly Algorithm (FA) and Ant Colony Optimization (ACO) to enhance resource allocation and load-balancing in cloud computing environments. Their method leverages the exploratory capabilities of FA and the exploitative strengths of ACO to address the NP-hard nature of load-balancing problems. By integrating these two metaheuristic algorithms, the study aims to improve key performance metrics such as Makespan, response time, and resource utilization. Simulation results demonstrated that the hybrid FA-ACO algorithm outperforms traditional scheduling techniques, offering a more efficient solution for dynamic and complex cloud infrastructures. This research underscores the potential of hybrid metaheuristic strategies in optimizing cloud resource management, providing valuable insights for developing more effective load-balancing algorithms.

The study by Daniel, Partheeban, and Sriramulu [4] introduces an enhanced version of the Ant Colony Optimization (ACO) algorithm aimed at improving load-balancing in cloud computing environments. The research focuses on optimizing the allocation of computational resources to ensure efficient distribution of workloads across the cloud infrastructure. By leveraging the inherent parallelism of the ACO algorithm, the study proposes an enhancement that increases the algorithm's ability to find optimal solutions for load-balancing, thus improving system performance. The proposed method enhances the basic ACO by incorporating adaptive parameters, which enable dynamic adjustments to the search process. This improvement results in better resource utilization, reduced Makespan, and enhanced scalability, all of which contribute to more efficient cloud resource management. The findings in this study are particularly relevant to our research as they highlight the potential of dynamic, metaheuristic algorithms like ACO for optimizing load-balancing in cloud systems, similar to the dynamic approaches explored in our enhanced PSO and ACO models.

Cao et al. [19] address the challenge of attacks in cloud computing by proposing an optimization-based real-time secure virtual machine (VM) allocation strategy. Their approach models the allocation problem as an optimization task that simultaneously minimizes security risks, power consumption, and workload imbalance across physical servers. To tackle the NP-hard nature of the problem, they employ Ant Colony Optimization (ACO) within time-windowed VM clustering, enabling dynamic and adaptive allocation decisions. Experimental results, based on real-world cloud traces, validate the effectiveness of their method in enhancing security and energy efficiency in cloud environments. This study underscores the potential of integrating metaheuristic algorithms like ACO for secure and energy-aware resource management in dynamic cloud infrastructures.

Kumar and Shinde [20] found that improved PSO implementations can boost response time by 23–41%. Devaraj et al. [21] introduced the FIMPSO algorithm—a hybrid of the Firefly Algorithm and Improved Multi-objective PSO (IMPSO)—designed to optimize resource allocation using geometric criteria and solution convergence mechanisms. Their results showed high efficiency with 98% CPU utilization, 93% memory utilization, and an average response time of 13.58 ms.

Energy efficiency has emerged as a critical performance concern as contemporary cloud systems increase in size and complexity. Scheduling algorithms must now address performance metrics and power consumption, particularly in IoT-integrated cloud networks where connected devices operate continuously. Notwithstanding their encouraging outcomes, these tactics have drawbacks, including processing expenses and migration delays that can restrict their use in cloud systems that are smaller or more sensitive to latency. However, the development of green computing infrastructures still depends heavily on improvements in energy-aware scheduling. Despite the enormous advancements in scheduling techniques, several methodological constraints remain. Many studies use modeling systems like CloudSim, which cannot accurately replicate real-time latency, hardware limitations, and network topologies. These drawbacks raise concerns about the external validity of the simulation results. Only a few studies, such as Kshama and S. R. [15], have incorporated real-world testing alongside simulation to validate their findings.

Theoretical Background

This study will employ several theoretical perspectives to substantiate it. EPSO is fundamentally based on Swarm Intelligence (SI), which is inspired by naturally occurring decentralized, self-organizing systems [22]. Algorithms based on SI, especially particle swarm optimization (PSO), improve task scheduling and resource allocation by modifying workload fluctuations. Recent studies underscore the significance of PSO in improving cloud settings.

The study includes Ant Colony Optimization (ACO), a bio-inspired algorithm based on concepts of Swarm Intelligence, alongside PSO. Ant Colony Optimization (ACO) simulates the foraging behavior of ants, wherein they employ pheromone trails to communicate and identify the most efficient pathways to food sources and their colony. Ant Colony Optimization (ACO) has been utilized in cloud computing to address complex optimization issues by probabilistically distributing the workload across resources, informed by current and historical system variables. In heterogeneous and dynamic situations, the adaptive and heuristic nature of ACO presents a compelling argument for addressing cloud load-balancing.

Utilizing simulation-based models, such as CloudSim, to analyze cloud computing environments enables researchers to investigate load-balancing techniques, like Enhanced Particle Swarm Optimization (EPSO) and Enhanced Ant Colony Optimization (EACO), inside a regulated cloud computing framework. This study assesses the aforementioned parameters without impacting an actual cloud computing setup. Simulation facilitates objective assessments, and current research emphasizing its significance has confirmed its value in the examination of cloud computing [23].

Utilizing these theoretical frameworks, the study provides a comprehensive overview of existing practices, challenges, and prospective solutions in cloud load-balancing. Through theoretical research and practical studies, the objective is to aid in the development and evaluation of EPSO and EACO to enhance the efficiency and scalability of cloud computing infrastructures.

Conceptual Framework

This study utilizes Enhanced Particle Swarm Optimization and Enhanced Ant Colony Optimization for cloud load-balancing integrated into a simulation system to test their effectiveness based on the five key main metrics of this study. Figure 1 illustrates the Input-Process-Output model, which systematically outlines how the researchers will conduct the study.

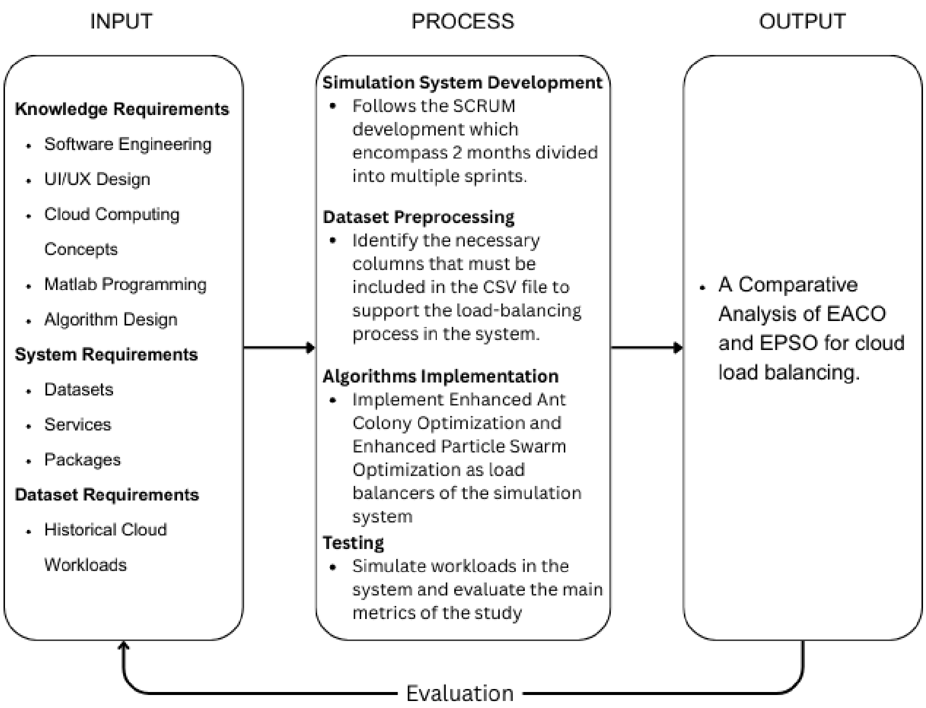

Figure 1. The Conceptual Framework of the Study

This study uses an Input-Process-Output (IPO) model to evaluate the efficiency of Enhanced Ant Colony Optimization and Enhanced Particle Swarm Optimization through the simulation system.

The input is the collection of all process requirements, including ability in software engineering, UI/UX design, cloud computing concepts, Matlab programming, and algorithm design. The system will also need particular datasets, relevant services, and supporting software tools. It will gather historical cloud workloads to provide the foundation for testing and simulation.

The process involves the simulation system development following the Scrum-based methodology organized over a two-month timeframe and split into time-boxed sprints. The datasets will be ready and preprocessed by the researchers throughout this period to assist the load-balancing simulation demands of the system. Using both the Enhanced Ant Colony Optimization (EACO) and Enhanced Particle Swarm Optimization (EPSO) techniques, the simulation system itself will operate as the load balancer distributing workloads and offering a visual representation of task distribution. The performance of both algorithms will then be assessed and compared using simulated testing.

The output will ultimately comprehensively compare EACO and EPSO algorithm performance. This study will focus on key performance metrics since these will indicate how effectively each load-balancing strategy works.

Synthesis

Two adaptive and metaheuristic algorithms that demonstrate a significant shift in the current corpus of knowledge on cloud load balancing are Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). These approaches have demonstrated superior performance in dynamic cloud environments, particularly in improving key metrics like response time, resource utilization, energy efficiency, degree of imbalance, and Makespan. As workloads in cloud infrastructures grow increasingly heterogeneous and unpredictable, the adaptability of these algorithms has made them highly relevant for modern cloud computing challenges.

A notable trend is the development of enhanced variants of these algorithms, designed to improve convergence speed, stability, and optimization quality. For example, the study by Zhang and Li demonstrated the energy-saving potential of Enhanced PSO (EPSO) with live migration strategies in cloud-based IoT systems. Similarly, enhanced ACO models, such as those discussed by Daniel et al., have integrated adaptive pheromone mechanisms to improve task distribution under fluctuating workloads. These enhancements show promise in addressing traditional limitations like premature convergence and local optima. Despite these advancements, a critical research gap exists in the direct comparative evaluation of EPSO and EACO, especially under controlled, realistic simulation conditions. Most existing studies either focus on a single enhanced algorithm or compare heuristic techniques to conventional scheduling methods, leaving the relative strengths of EPSO and EACO underexplored. Furthermore, evaluations are often limited to a narrow set of metrics, typically emphasizing only response time or resource usage, with energy efficiency and load variance receiving less systematic attention.

Another gap is the lack of practical validation frameworks. Many previous works have relied solely on simulation data without incorporating user-centered feedback from professionals who manage real-world cloud infrastructure. This limits the findings' applicability to actual deployment scenarios, where system usability and scalability are as critical as algorithmic performance.

This study aims to address these gaps by implementing and comparing EPSO and EACO within a CloudSim-based simulation environment that incorporates realistic workload traces from the Google Cluster Dataset. The evaluation encompasses five critical performance metrics: response time, resource utilization, energy efficiency, degree of imbalance, and Makespan. The study further integrates user evaluations from cloud specialists and IT experts to assess the system's functional suitability, scalability, and accuracy—bridging the gap between simulation theory and operational relevance. Through this multifaceted approach, the study contributes not only a rigorous performance comparison of EPSO and EACO but also offers practical insights into their applicability in cloud resource scheduling. This work advances the field of cloud computing by informing the development of more adaptive, sustainable, and high-performance load-balancing strategies.

Definition of Terms

Conceptual Terms

Cloud Computing. A technology that enables on-demand access to a shared pool of configurable computing resources—such as servers, storage, and applications—typically delivered over the internet [24].

Load-Balancing. A technique used in cloud computing to distribute workloads efficiently across multiple servers or virtual machines (VMs) to optimize resource use, reduce delays, and prevent overload [25].

Scalability. The ability of a cloud system to handle increasing workloads by dynamically adding resources without compromising overall performance [24], [26], [27].

Swarm Intelligence. A concept inspired by natural collective behavior—such as bird flocking or ant foraging—underpinning algorithms like PSO and ACO, which optimize cloud task scheduling adaptively [22].

Technical Terms

Enhanced Particle Swarm Optimization (EPSO). An improved form of the Particle Swarm Optimization algorithm that applies adaptive velocity clamping and nonlinear inertia weight reduction to efficiently assign tasks in cloud environments without relying on VM migration [7], [28].

Enhanced Ant Colony Optimization (EACO). A modified ACO algorithm that employs adaptive pheromone evaporation and heuristic load-based reinforcement to optimize task distribution across virtual machines under changing workloads [4].

Response Time. The total time from the submission of a task to its completion. It is a key performance indicator in cloud computing, reflecting system responsiveness [29].

Resource Utilization. A metric indicating how efficiently computing resources (CPU, memory) are allocated and used across VMs in the cloud to minimize idle capacity [30].

Energy Efficiency. The ability of the cloud system to reduce energy consumption while maintaining performance, measured using power models based on CPU utilization [7].

Degree of Imbalance. A performance metric in cloud computing that quantifies the unevenness of workload distribution across virtual machines (VMs). It is calculated by comparing the difference between the execution time of the most and least loaded VMs relative to the average execution time. A lower degree of imbalance indicates a more efficient and fair allocation of tasks, which contributes to improved system stability and performance [3].

Makespan. The total time required to complete all tasks in a cloud environment within a given scheduling period. The VM determined that the assigned workload finishes last. Makespan is a key indicator of scheduling efficiency, with shorter makespan values reflecting better task distribution and faster execution of workloads [3].

Adaptive Pheromone Evaporation. Technique in the Enhanced Ant Colony Optimization (EACO) algorithm that adjusts the rate at which pheromone trails—signals used by virtual "ants" to mark task distribution paths—fade over time [4].

Adaptive Velocity Clamping. Adjusts particle velocity limits in PSO based on swarm diversity, enhancing adaptability, also the inspiration of researchers proposed EPSO algorithm. It balances exploration and exploitation for optimized task scheduling in cloud load-balancing [6].

Heuristic load-based reinforcement. An approach that combines heuristic algorithms with adaptive feedback to optimize load distribution in cloud environments, improving performance based on system responses over time. [31]

Non-Linear Inertia Weight Reduction. Using a nonlinear function to decrease the inertia weight leads to improved convergence and better solutions [32]

CloudSim. A simulation framework for modeling and analyzing cloud computing environments, enabling evaluation of scheduling algorithms like EPSO and EACO under simulated workloads [33].

Google Cluster Traces. A dataset containing real-world workload traces collected from Google's data centers is used in this study to provide realistic workload input for simulation [34].

Virtual Machine (VM). A virtualized computing environment that acts as a software emulation of a physical computer, allowing flexible and scalable deployment of tasks in the cloud [26].

Task Scheduling. The process of assigning cloud tasks to available computing resources (VMs) to optimize performance metrics such as response time and energy efficiency [35].

Heuristic Algorithms. Optimization algorithms that use experience-based techniques to solve complex problems efficiently. EPSO and EACO are examples of such algorithms used for load-balancing [4] [18], [36].

Simulation Environment. A controlled setup that replicates real-world computing systems, enabling testing and analysis of algorithms under different workload scenarios without affecting live cloud infrastructure [23], [37].

CHAPTER III
Research Methodology

This chapter outlines the research design, study location, participants, and data collection procedures. The information gathered will act as a guide that describes the methodologies, context, participant selection, and data collection process crucial for this study.

Research Design

The study will employ a quantitative, simulation-based research design, leveraging computational models to estimate and analyze algorithm performance in cloud computing environments. This approach is especially suitable for complex system evaluations where real-world testing may be limited by access, infrastructure, or cost constraints. It allows for objective, repeatable experimentation under controlled conditions, providing precise measurement of key performance metrics. The core of the simulation environment is CloudSim, a widely recognized toolkit for modeling and evaluating cloud infrastructures and scheduling strategies [33]. To ensure practical relevance, the simulation integrates workload data from the Google Cluster Traces, a dataset derived from real-world cloud infrastructure. Due to the extensive volume of this dataset, a preprocessed subset will be used, which will be sufficient to preserve task variability and system load dynamics without compromising execution efficiency.

The study focuses on five key performance metrics: response time, resource utilization, energy efficiency, degree of imbalance, and Makespan. Collectively, these metrics provide a comprehensive understanding of how well each algorithm manages cloud task scheduling under dynamic conditions.

The system integrates with MATLAB to visualize and analyze simulation results. This enables detailed performance graphing and statistical interpretation, which aids in identifying patterns and validating the effectiveness of the algorithms.

In line with emerging trends in adaptive computing, the study evaluates and compares two metaheuristic algorithms: Enhanced Particle Swarm Optimization (EPSO), which incorporates adaptive velocity clamping and nonlinear inertia weight reduction, and Enhanced Ant Colony Optimization (EACO), which features adaptive pheromone evaporation and heuristic load-based reinforcement.

These enhancements are specifically designed to address the limitations of their base forms and adapt to fluctuating workloads within cloud environments. The comparative nature of the study addresses a gap in existing research, where EPSO and EACO are rarely analyzed side-by-side under identical simulated conditions.The simulation-based approach offers multiple advantages in evaluating load-balancing algorithms. It allows for precise control over experimental variables, ensuring fair and reproducible comparisons between algorithms under identical conditions. Additionally, it enables the use of historical workload patterns, enhancing the real-world relevance and applicability of the results. This method also supports the testing of theoretical models—such as those based on Swarm Intelligence—within scalable and controlled environments, eliminating the risk of disrupting live cloud systems during experimentation. In addition to simulation results, the study incorporates feedback from cloud and IT specialists, whose insights validate the practical relevance, usability, and scalability of the simulation system. These participants evaluate the system based on structured surveys and represent both academic and industry perspectives. Their evaluations cover functional suitability, interaction capability, result accuracy, and scalability—addressing the system's practical deployment potential alongside algorithmic performance. Researchers ensure that the research design has scientific rigor and applied relevance by combining simulation modeling with expert feedback. It provides a solid foundation for evaluating EPSO and EACO and contributes to developing efficient and sustainable load-balancing strategies in cloud computing.

Research Locale

The researchers will conduct the study in collaboration with Tutorials Dojo, a cloud computing training and content provider based in Manila, Philippines. Tutorials Dojo, along with expert alumni from Cabuyao, provides the ideal environment for the study, renowned for its expertise in evaluating cloud service performance and training individuals for AWS certifications. The team will meet online and in person during the data collection and evaluation phases to ensure effective communication and collaboration. Because of this structure, the researchers may collaborate closely with cloud specialists and IT experts, ensuring accurate interpretation of simulation results and valuable feedback on the load-balancing evaluation.

Respondents of the Study

The study employed purposive sampling to select research participants composed of cloud professionals from Tutorials Dojo, a cloud computing training and content provider based in Manila, Philippines, and expert alumni from Cabuyao. The participants include four (4) academic professionals with backgrounds in computing and systems analysis, eleven (11) cloud specialists who are cloud computing practitioners currently engaged in hands-on training and industry-related activities, and fifteen (15) IT experts from Tutorials Dojo's technical team and Cabuyao alumni with direct experience in managing cloud infrastructure and system performance.

Grounded on real applications and developing industry expertise, these respondents will provide insightful analysis. While the end users will offer pertinent viewpoints from the perspective of early-stage practitioners, the technical staff will give expert-level evaluations of the system's efficacy and operation. Their combined understanding of cloud computing, load-balancing algorithms, and system simulations will support a balanced and comprehensive assessment of the methodologies presented in this study.

Classification	Total Participants
IT Experts	15
End Users (Cloud Specialists and Academic Professionals)	15
Total:	30

Table 1. Respondents of the Study

Table 1 presents the classification and total number of respondents involved in the study. The study selected twenty (30) participants through purposive sampling, including fifteen (15) end users composed of eleven (11) cloud specialists, four (4) academic professionals, and fifteen (15) IT experts. These respondents were chosen based on their relevant experience and knowledge of cloud computing and load-balancing systems.

Data Gathering Procedure

Discussion of Algorithms

The study focuses on evaluating the performance of two metaheuristic load-balancing algorithms—Enhanced Particle Swarm Optimization (EPSO) and Enhanced Ant Colony Optimization (EACO)—using the CloudSim simulation tool. The objective is to optimize task scheduling in cloud computing environments by analyzing key performance metrics such as resource utilization, response time, energy efficiency, degree of imbalance, and Makespan. Among these, the degree of imbalance and Makespan are fundamental because they directly measure the evenness of workload distribution and the overall scheduling efficiency. These are critical for assessing how well EPSO and EACO optimize task execution across virtual machines (VMs). These metrics provide a comprehensive understanding of algorithm performance under realistic conditions, such as those derived from the Google Cluster Dataset and custom end users' workload data, and align our study with established research standards in cloud computing.

A. Metrics for Evaluation

The study focuses on five key performance metrics to evaluate and compare the efficacy of Enhanced Particle Swarm Optimization (EPSO) and Enhanced Ant Colony Optimization Algorithm. These metrics comprehensively assess how well each algorithm manages job distribution in a cloud simulation.

A.1. Response Time

The duration between submitting and completing a task, known as response time, indicates system responsiveness and is critical for user satisfaction in time-sensitive applications. Response time is calculated for each task in the simulation and averaged to evaluate the system's overall performance. Researchers considered response time the primary metric for the study to ensure fast task processing for universal workloads.

$$Response\ Time = \frac{1}{n}\sum_{i=1}^{n}(completion\ time_i - submission\ time)$$

The equation computes the average response time across all tasks - to measure how quickly the system processes user requests like login and fetching data, thus reflecting the user experience. Completion time represents when the task finishes execution, and submission time marks when a task is submitted through the system to define each component. In CloudSim, *n* can represent the number of cloudlets or tasks. Lastly, $\frac{1}{n}\sum$ averages the latency for all cloudlets or tasks.

A.2. Resource Utilization

Resource utilization shows how effectively the system uses resources such as CPU and memory. High utilization indicates efficient resource usage, while low or imbalanced utilization suggests underused or overloaded resources. Resource utilization is tracked per virtual machine and host in the simulation and averaged to determine overall utilization efficiency.

$$Resource\ Utilization = \frac{1}{m}\sum_{j=1}^{m} U_J^P$$

The above shows the resource utilization metric adapted from Zhang and Li's energy model. U_J^P represents the CPU utilization of the host based on allocated tasks and virtual machines, reflecting the computational load from tasks and VMs. For instance, a host running 80% CPU capacity has 0.8 normalized CPU utilization. m represents the number of hosts in the system. Lastly, $\frac{1}{m}\sum$ represents the average utilization of all hosts [7].

A.3. Energy Efficiency

Energy efficiency measures power usage during task scheduling, including active energy for processing and idle energy for powered-on machines. The simulation uses a power model based on utilization. Energy consumption was calculated using the model from Zhang and Li, which estimates server power consumption based on CPU utilization and sums it to determine total energy consumption [7]. In our modified algorithms, we have removed the VM migration component, thus eliminating the server consolidation mechanism that supports energy savings. Nevertheless, we continue to adopt Zhang and Li's energy consumption model to evaluate energy efficiency as a secondary metric, acknowledging that the potential for energy efficiency is reduced without VM migration.

$$P_j = \begin{cases} (P_j^{busy} - P_j^{idle}) \times U_j^p + P_j^{idle}, & U_j^c > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$Energy = j\sum Pj$$

P_J^{busy} represents the power the host consumes at full utilization, which measures 215 watts, according to Zhang and Li [7]. Pj^{idle} represents the power usage when the host is idle, consuming 162 watts - baseline energy to keep the system operational. U_j^p represents the normalized CPU utilization of the host - indicates the demand placed on the CPU. U_j^c acts as the binary indicator - a class of statistical tools used to represent one of two possible states [7]. The formula takes a value greater than zero if the host is active - which can be distinguished by operational and non-operational states. $j\sum Pj$ represents the total energy consumption across all the hosts.

A.4. Degree of Imbalance (DI)

The degree of imbalance is calculated based on the completion times of VMs, comparing the maximum and minimum completion times relative to the average. This approach is widely used in cloud computing literature and aligns with the simulation-based evaluation study in CloudSim [3]. The formula is:

$$DI = \frac{(MaxTime - MinTime)}{AverageTime}$$

A.5. Makespan

The Makespan measures the total time required to complete all tasks in the system. It is a direct indicator of overall scheduling efficiency, where each *Ti* represents the total execution time of the i-th VM. Essentially, Makespan is determined by the VM, which takes the longest to finish its assigned tasks. A shorter Makespan implies that tasks have been efficiently distributed, allowing all VMs to complete around the same time. In contrast, a longer Makespan points to inefficiencies—typically when one VM is overloaded compared to others. Reducing the Makespan is critical for ensuring fast task completion and optimal system performance [3].

$$MakespanTime = Max\,\Sigma^{n}_{i=0}\ CompletionTime$$

B. Enhanced Ant Colony Optimization Overview

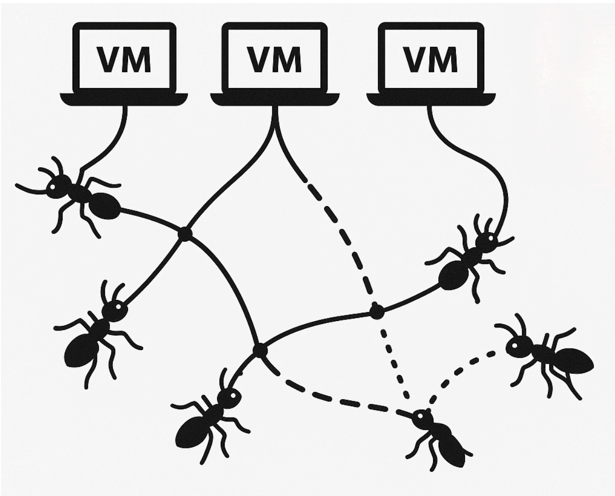

Figure 2. Illustration of EACO in a Cloud Simulation Environment

This diagram illustrates the Enhanced Ant Colony Optimization situated in a simulated environment. The ants represent intelligent agents which explore various paths for assigning tasks to virtual machines (VMs). Each path illustrates the performance outcomes of each path where thicker lines indicate strong down to the least dashed lines. The key enhancement in this approach is the adaptive pheromone evaporation: when the system has high workloads, the trail of pheromone evaporates quickly, meaning the ants will be encouraged to explore new paths, potentially leading to a much better solution, avoiding overloaded VMs. But when the system is stable, evaporation slows down, reinforcing better paths.

To enhance optimization, EACO utilizes the following formula to adaptively adjust pheromone evaporation:

$$p(t) = p_{min} + (p_{max} - p_{min}) * \left(\frac{f_{best}(t) - f_{avg}(t)}{f_{best}(t)}\right)$$

Where p(t) is the current evaporation rate at time (t), while p_{min} and p_{max} define the lower and upper bounds for evaporation. $f_{best}(t)$ is the best fitness value achieved during present iteration, and $f_{avg}(t)$ is the average fitness across all solutions. The difference between the best and average fitness represents the diversity or convergence of the population. When the average is close to the best, implying less diversity, the evaporation rate is reduced to retain pheromone and exploit known good solutions.

Additional enhancement of this algorithm is implemented using the heuristic-based reinforcement formula:

$$\Delta\tau_{ij}(t) = \frac{1}{1 + L_{ij}(t)}$$

Where $\Delta\tau_{ij}(t)$ is the amount of pheromone added to path ij at time t, and $L_{ij}(t)$ represents the current load on node j, it may include CPU usage and memory load. This formula ensures that highly loaded nodes receive less pheromone, steering the algorithm towards less burdened resources.

then these enhancements are applied in the update:

$$\tau_{ij}(t + 1) = (1 - p(t)) * \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

This updated rule combines adaptive evaporation and heuristic-based load reinforcement, enabling ACO to respond dynamically to both solution quality and real-time resource usage key for effective task scheduling in cloud environments.

Pseudocode of EACO Algorithm

Algorithm: Enhanced Ant Colony Optimization (EACO)

Require: *num of ants(m), maximum iterations (MaxIterations), pheromone importance factor (α), heuristic importance facto (□) , evaporation rate bounds (p_min, p_max), initial pheromone levels (τ_ij), heuristic information (η_ij)*

Ensure: *Optimized task schedule (best_solution)*

1: ***Initialization:***

2: Initialize pheromone levels τ_ij for all task-resource pairs

3: Compute heuristic information η_ij for all task-resource pairs

4: Set initial best solution best_solution ← null

5: Initialize gbest to the best pbest in the swarm

6: ***Main loop:***

7: for iteration = 1 to MaxIterations do

8: for each ant k = 1 to m do

9: Construct a complete task-to-resource assignment S_k based on transition probabilities:

```
            For each task i, select resource j with probability:
                P_ij = [τ_ij]^α × [η_ij]^β / Σ([τ_ik]^α × [η_ik]^β) for all feasible k
            Evaluate fitness f_k of solution S_k
            If best_solution is null or f_k better than fitness of best_solution then
                Update best_solution ← S_k
        end for
        Compute average fitness f_avg across all ants
        Compute best fitness f_best among all ants
        Compute adaptive evaporation rate p(t):
            p(t) = p_min + (p_max - p_min) × ((f_avg - f_best) / f_best)
        for each task-resource pair (i, j) do:
            Compute current load L_j on resource j (e.g., CPU, memory usage)
            Compute heuristic load-based reinforcement:
                Δτ_ij = 1 / (1 + L_j)
            Update pheromone level:
                τ_ij ← (1 - p(t)) × τ_ij + Δτ_ij
        end for
    end for
Return: best_solution as the optimized task schedule
```

C. Enhanced Particle Swarm Optimization (EPSO) Overview

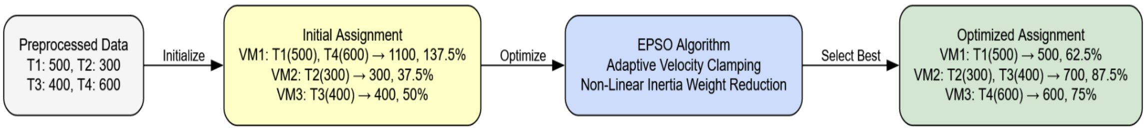

Figure 3. Implementation of EPSO in a Cloud Simulation Environment

Figure 3 shows how EPSO optimizes task scheduling in a cloud simulation environment. Starting with preprocessed Google Cluster Data (tasks T1: 500, T2: 300, T3: 400, T4: 600), an initial random assignment overloads VM1 (137.5%) while underutilizing VM2 and VM3. EPSO, using adaptive velocity clamping and non-linear inertia weight reduction, reassigns tasks to balance the load: VM1 at 62.5%, VM2 at 87.5%, and VM3 at 75%.

The population-based optimization method known as Particle Swarm Optimization (PSO), which was motivated by the social behavior of fish schools and flocks of birds, as discussed in [38], which references Reynolds' boids model that simulates collective movement through rules like separation, alignment, and cohesion. In cloud computing, PSO employs a swarm of particles, where each particle represents a potential mapping of tasks to resources, such as virtual machines (VMs). The goal is to optimize task assignments to minimize energy consumption and network traffic while ensuring efficient resource utilization and preventing server overloads. Each particle's position is iteratively updated based on its best-known position (personal best) and the best position found by the entire swarm (global best), guided by velocity updates that balance cognitive and social influences. This iterative process enables the swarm to explore the solution space effectively and converge toward optimal or near-optimal task schedules. Recent studies have significantly advanced PSO's application in cloud computing task scheduling. For instance, a study proposed a hybrid algorithm combining PSO with simulated annealing to reduce Makespan and energy consumption, achieving up to 15% improvement in scheduling efficiency compared to traditional methods [39]. Another study introduced an enhanced PSO with adaptive inertia weight and velocity clamping, dynamically adjusting parameters to improve convergence and solution quality, particularly for high-dimensional problems [40]. Additionally, a recent study developed PSOIVC, a diversity-aware PSO variant that tunes inertia weight and velocity based on swarm diversity, enhancing exploration-exploitation balance and demonstrating superior performance on benchmark functions [41]. These enhanced PSOs have improved execution times, energy efficiency, and load balancing, underscoring PSOs' effectiveness in addressing complex scheduling challenges in modern cloud environments without relying on virtual machine live migration, much like Zhang and Li do to enhance energy efficiency.

Zhang and Li proposed an energy model that provides a contextual reference for energy-efficient cloud systems by classifying servers as overloaded, underloaded, or usual and migrating virtual machines (VMs) to balance loads and reduce energy use by powering off idle servers, enhancing efficiency and sustainability and beneficial for evaluating energy efficiency [7]. However, our study employs an Enhanced Particle Swarm Optimization (EPSO) algorithm for task scheduling, optimizing task-to-VM assignments without relying on VM migration. EPSO treats each potential task-to-VM mapping as a particle in a swarm, iteratively adjusting positions to find optimal schedules based on personal best (pBest) and global best (gBest) positions. The velocity and position updates are governed by:

In PSO-based scheduling, particles iteratively update their velocities and positions using the formula:

$$v[] = v[] + c1 * rand() * (pBest[] - position[]) + c2 * rand() * (gBest[] - position[])$$

Followed by:

$$position[] = position[] + v[]$$

EPSO incorporates non-linear inertia weight, decreasing quadratically over iterations to shift from exploration to exploitation [32].

$$w = w_{max} - (w_{max} - w_{min}) \times (\frac{iteration}{maxIterations})^2$$

To enhance optimization, EPSO incorporates adaptive velocity clamping inspired by the PSO-SAVL algorithm, dynamically adjusting the velocity limit as:

$$V_{max} = V_{maxInitial} - (V_{maxInitial} - V_{maxFinal}) \times \left(\frac{iteration}{maxIterations}\right)^2$$

This iteration-based approach, inspired by adaptive strategies in PSO-based studies including PSO-SAVL by Li et al. [6], balances exploration and exploitation, improving convergence for task scheduling.

Pseudocode of EPSO Algorithm

Algorithm: Enhanced Particle Swarm Optimization (EPSO)

Require: *num particles, max iterations, w max, w min, c1, c2, v max initial, v max final*

Ensure: *Optimized task schedule (gbest)*

```
Initialization:
    Initialize swarm with random task-to-VM mappings
    Initialize velocity for each particle
    Initialize pbest for each particle to their initial position
    Initialize gbest to the best pbest in the swarm
Main loop:
    for iteration = 1 to max iterations do
        w ← w_max - (w_max - w_min) × (iteration/max_iterations)²
        v_max ← v_max_initial - (v_max_initial - v_max_final) × (iteration/max_iterations)²
        Evaluate and update pbest and gbest:
            for all particle in swarm do
                Evaluate fitness of particle
                if fitness better than pbest then
                    Update pbest to current position
```

```
        end if
        if fitness better than gbest then
            Update gbest to current position
        end if
    end for
Update velocities and positions with adaptive clamping:
    for all particle in swarm do
        for all dimension do
            v[dimension] ← w × v[dimension] + c1 × rand() × (pbest[dimension] - position[dimension]) + c2 × rand() × (gbest[dimension] - position[dimension])
            if v[dimension] > v_max then
                v[dimension] ← v_max
            else if v[dimension] < -v_max then
                v[dimension] ← -v_max
            end if
            position[dimension] ← position[dimension] + v[dimension]
        end for
    end for
end for
```

*33: **Return:** gbest as optimized task schedule*

Methodology

System Development

The simulation system will be designed by the researchers using the Scrum approach, which is augmented by timeboxing [42]. Scrum splits the development process into iterations of fixed length, called sprints, each of a fixed duration (usually 1-2 weeks). The team collaborates to produce a potentially discernable increment of the system at the end of each sprint for testing due to the nature of this experimental study. The process calls for continuous feedback and adjustment, enabling constant refinement and project goal alignment. During every sprint planning, the team determines the volume of work, such as preprocessing datasets, wherein dataset needs are assessed, and necessary columns are selected for the load-balancing simulation. The tasks and goals for every sprint are made to be accomplished within the two months allocated for this proposed system.

The development method emphasizes incremental system construction and development throughout phases. Main features are built in initial sprints and improved later on through performance evaluation and feedback. Each sprint has a testing component that utilizes workload simulation to measure algorithm performance and track principal parameters.

The findings from these tests feed the subsequent sprint, which allows for continued improvement to the system and load-balancing algorithms. The use of timeboxing in Scrum ensures that the sprint follows its specified timeline, thus ensuring team concentration and reducing wasteful time. The suggested system follows a cycle of continuous improvement through an iterative process done over a sequence of two-month periods. Russo et al. state that timeboxing sets deadlines to ensure the timely completion of intricate projects or systems [42].

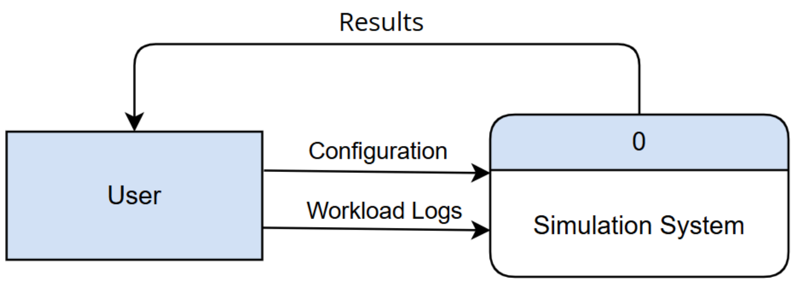

Figure 4. Context Diagram: User Interaction with the Simulation

This context diagram shows a User interacting with a Simulation by inputting configuration settings and workload logs as input. The Simulation System processes these inputs and returns the results to the User. The diagram represents a simple flow where the User provides the necessary data, and the system performs the simulation and delivers the output.

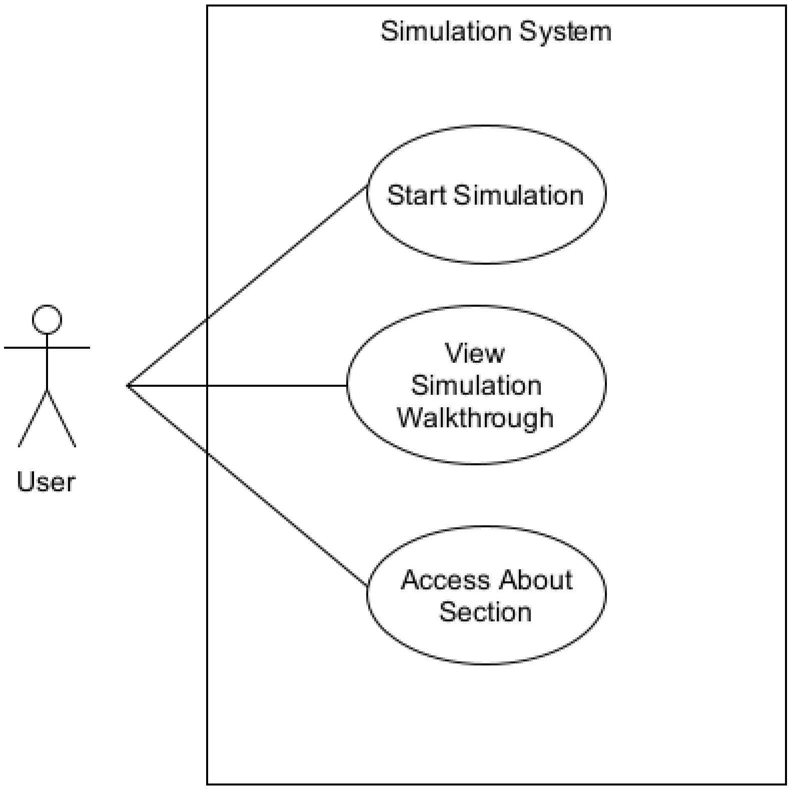

Figure 5. Use Case Diagram for User Interaction with the Home Page

The use case diagram illustrates the User's interaction with the system when they are on the home page, where they can start the simulation, view a walkthrough of how the simulation works, and find other information about the system and the developers.

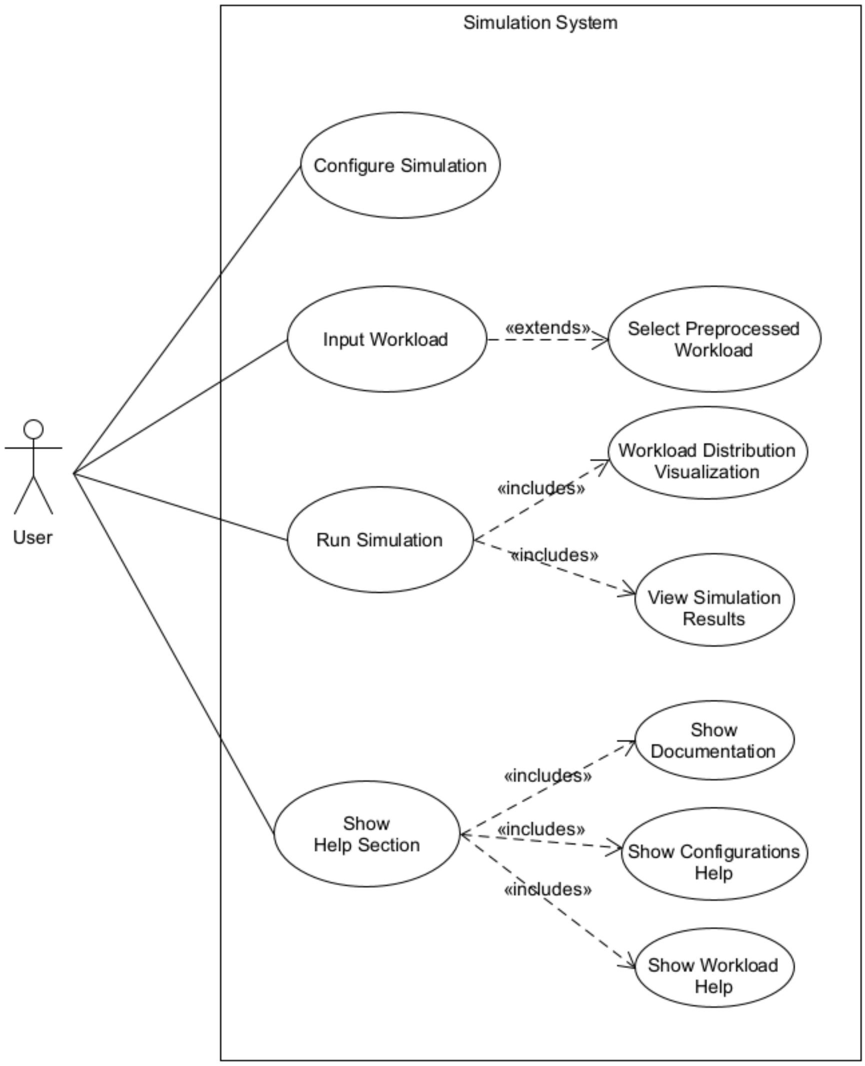

Figure 6. Use Case Diagram for User Interaction with the Simulation Page

The use case diagram illustrates the User's interaction with the system. They can configure the simulation, input workload or select a provided preprocessed workload, run the simulation, which includes the load distribution visualization across virtual machines, view the detailed results after the process, and access help, which aids them with ample information on configurations, workloads, and system documentation.

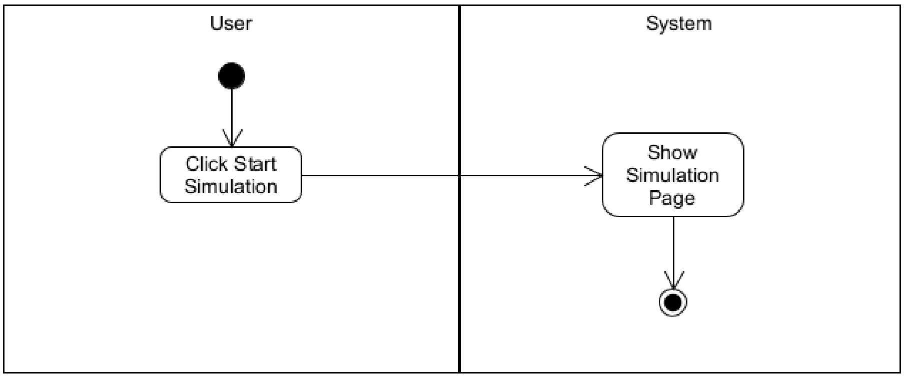

Figure 7. Activity Diagram for Starting the Simulation

The activity diagram illustrates the process of starting a simulation. It begins with the user action "Click Start Simulation," which triggers the system to display the "Simulation Page," marking the start of the simulation interface.

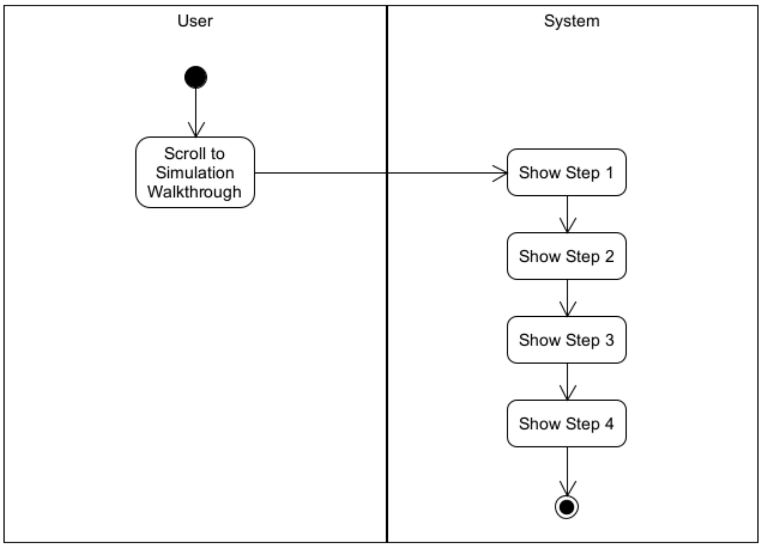

Figure 8. Activity Diagram for Viewing the Simulation Walkthrough

This activity diagram depicts the sequence for viewing the simulation walkthrough. When the user scrolls to the walkthrough section, the system sequentially displays four steps—Step 1 through Step 4—guiding the user through the simulation process.

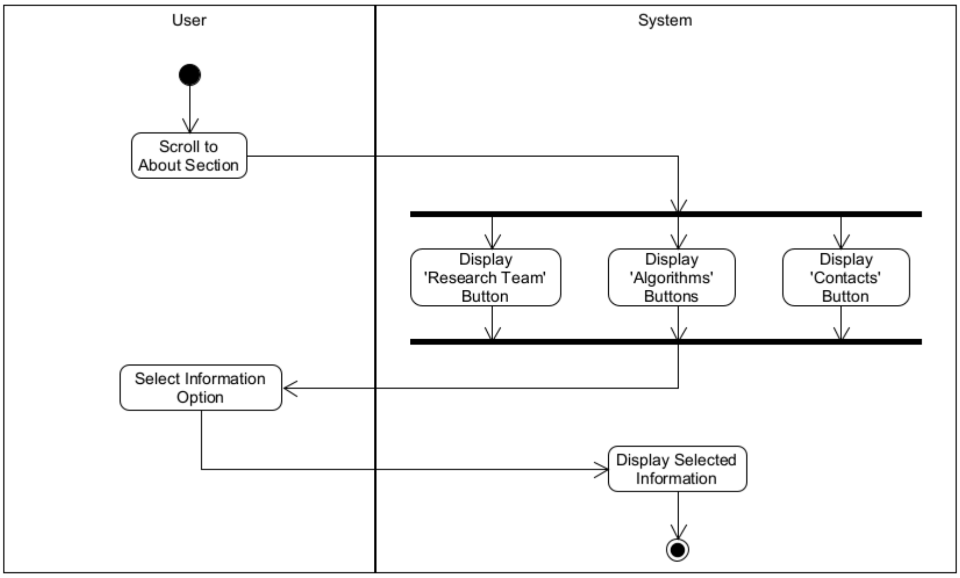

Figure 9. Activity Diagram for Assessing About Section

This activity diagram outlines the process of accessing the About section. Upon scrolling to the section, the system displays three buttons: Research Team, Algorithms, and Contacts. The user then selects an information option, prompting the system to display the corresponding content.

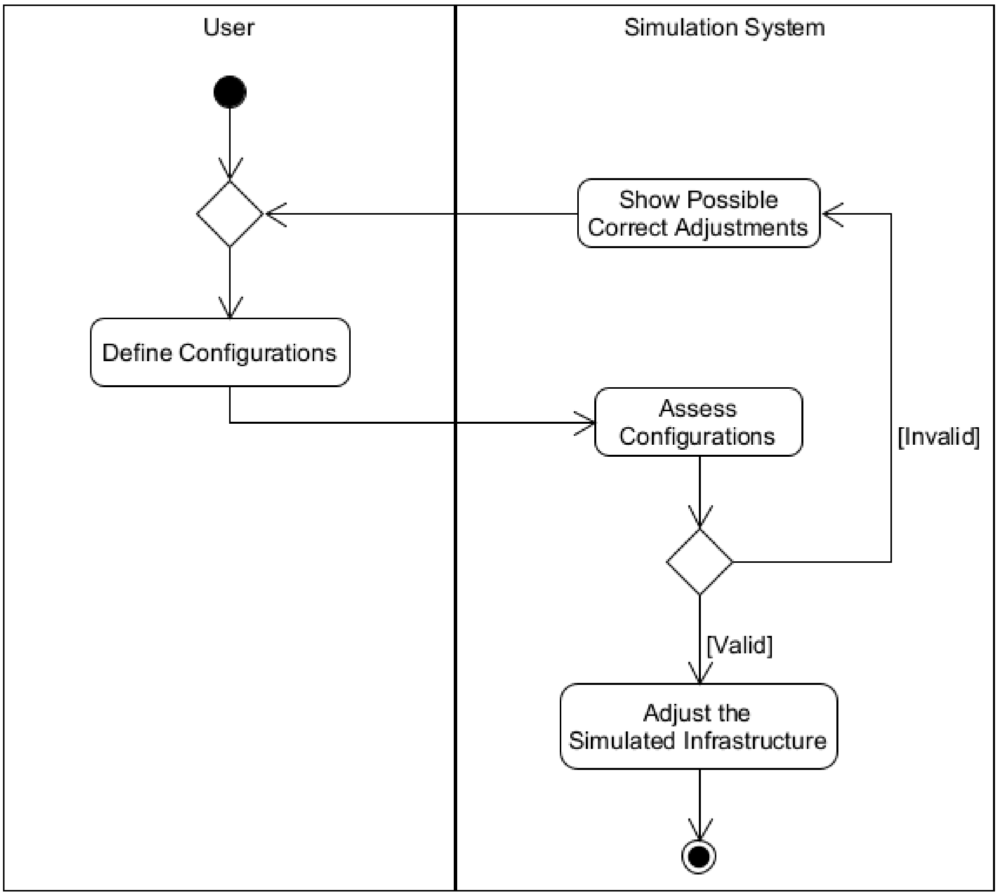

Figure 10. Activity Diagram for Configurations

This activity diagram represents the User defining the configuration for the cloud infrastructure. The system assesses the User's defined configuration to check its validity. If the system accepts the configuration, it adjusts the simulation accordingly; otherwise, it displays possible configuration adjustments.

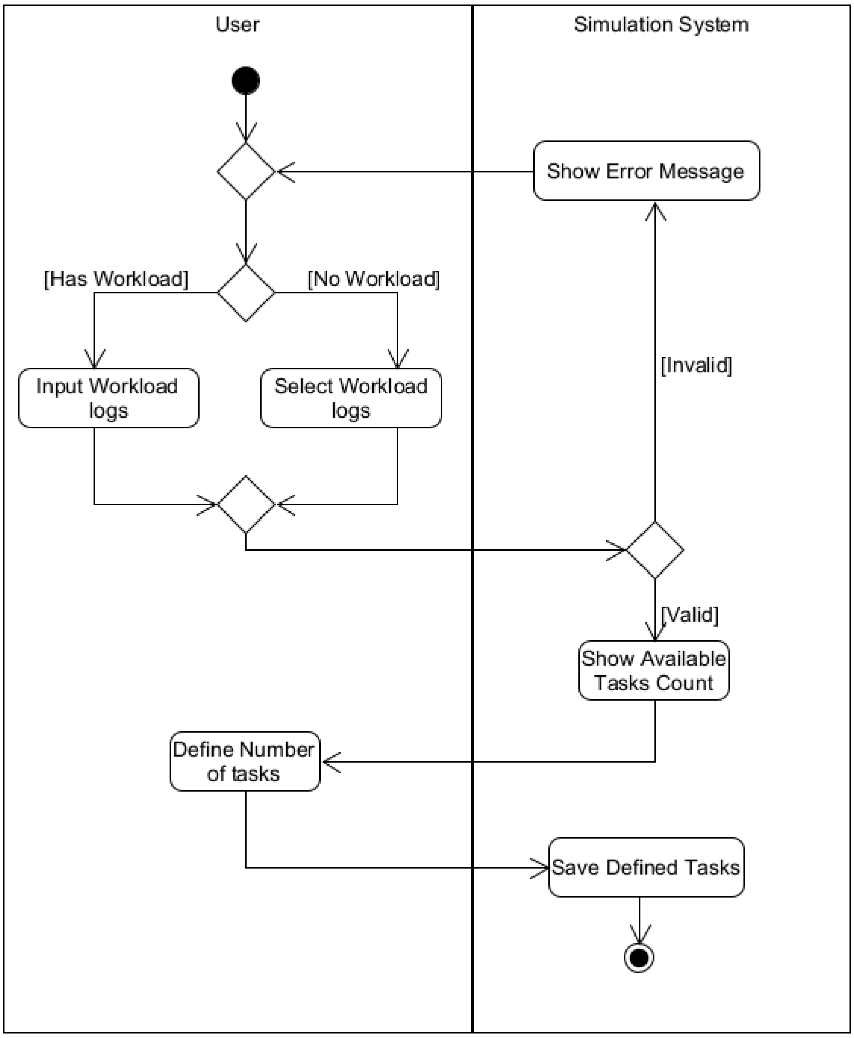

Figure 11. Activity Diagram for Workload Input

This activity diagram represents the option of inputting the workload if the User already has predefined workload logs; otherwise, they can select from available preprocessed workloads. Then, it validates if their input is valid to avoid errors, processes the data, shows the available task count, lets the User define the task to process in the simulation, and saves it for simulation later.

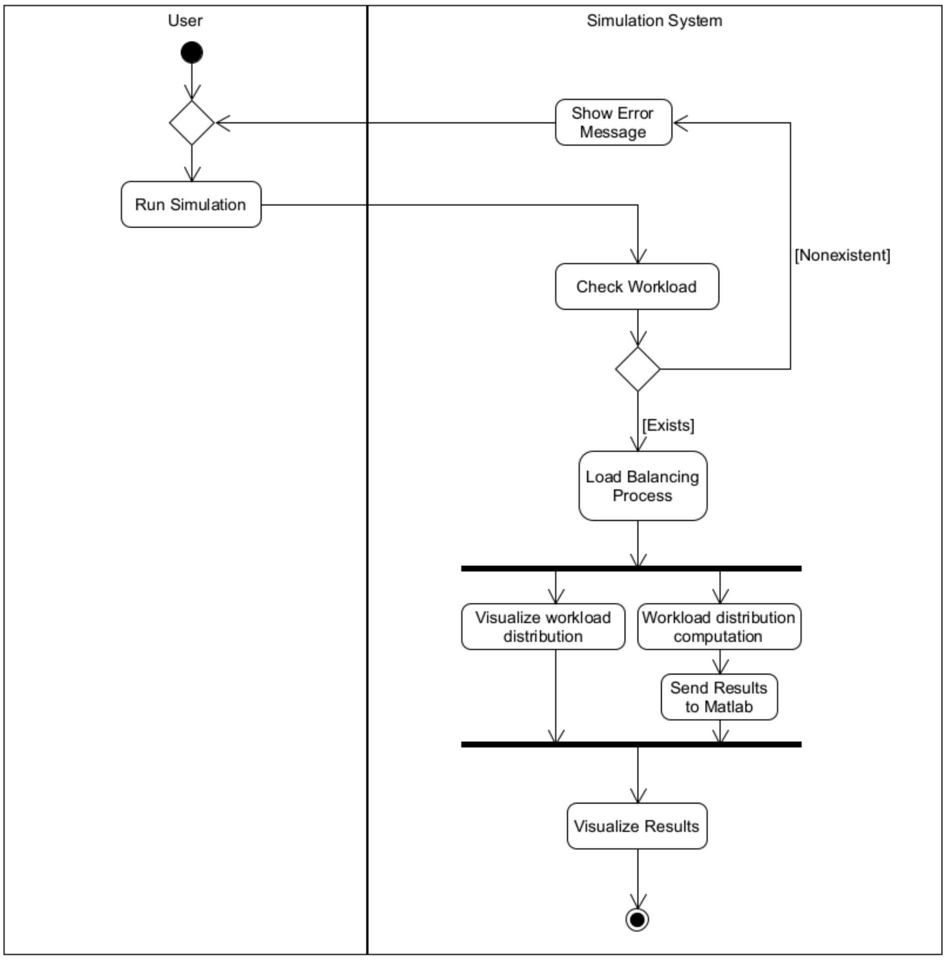

Figure 12. Activity Diagram for Running Simulation

This activity diagram represents the simulation process, where the system first checks if the workload exists and then proceeds to the load-balancing process. Afterward, it visualizes the workload distribution and sends the results to MATLAB, which returns the data for a more transparent and understandable visualization.

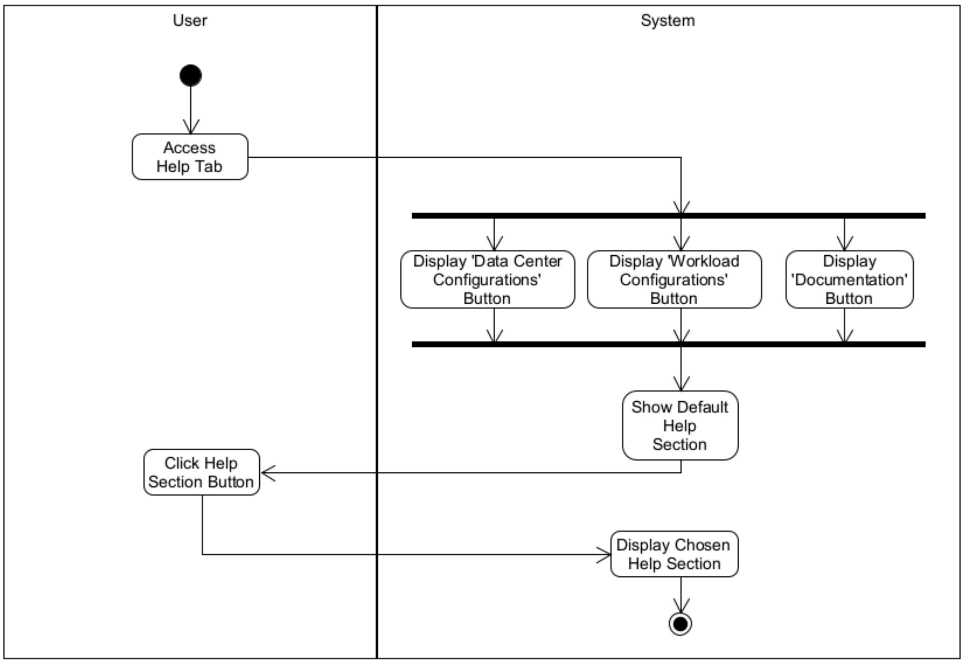

Figure 13. Activity Diagram for Showing Help

This diagram illustrates the sequence of actions between a user and the system when accessing help features. The process begins with the user accessing the Help tab, prompting the system to display multiple help-related buttons such as “Data Center Configurations,” “Workload Configurations,” and “Documentation.” The system then shows a default help section. If the user selects a specific help section, the system responds by displaying the chosen content and completing the interaction.

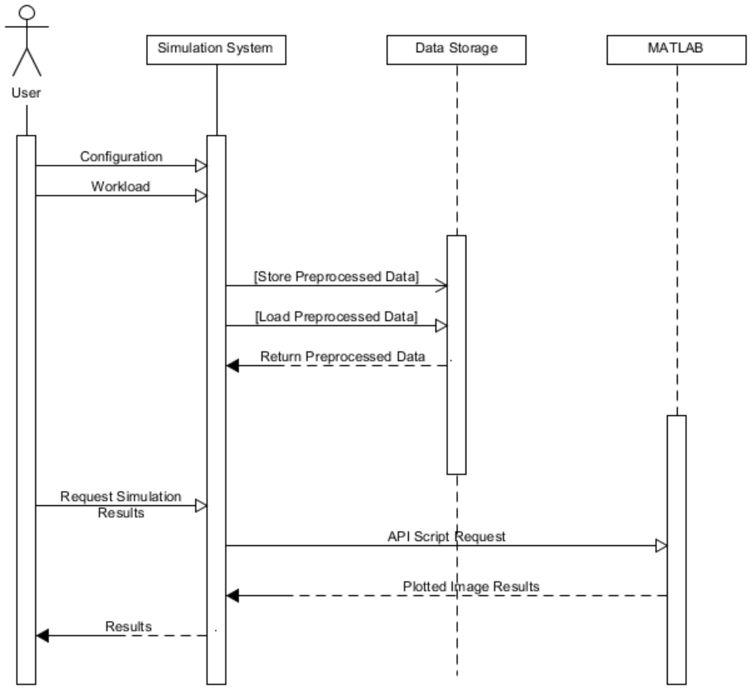

Figure 14. Sequence Diagram of Data Processing and Result Generation

This sequence diagram shows how a user interacts with a simulation system to configure and run a simulation, with support from data storage and MATLAB for processing and visualization. The process starts when the user sends configuration and workload details to the simulation system. The system then stores this preprocessed data in a storage module and later retrieves it when needed. Once the user requests the simulation results, the simulation system sends a request to MATLAB through an API script to generate visual output. MATLAB processes the data and returns plotted image results, which the simulation system then sends back to the user as the final output. The diagram highlights the flow of data and coordination between different components to deliver simulation results in a visual format.

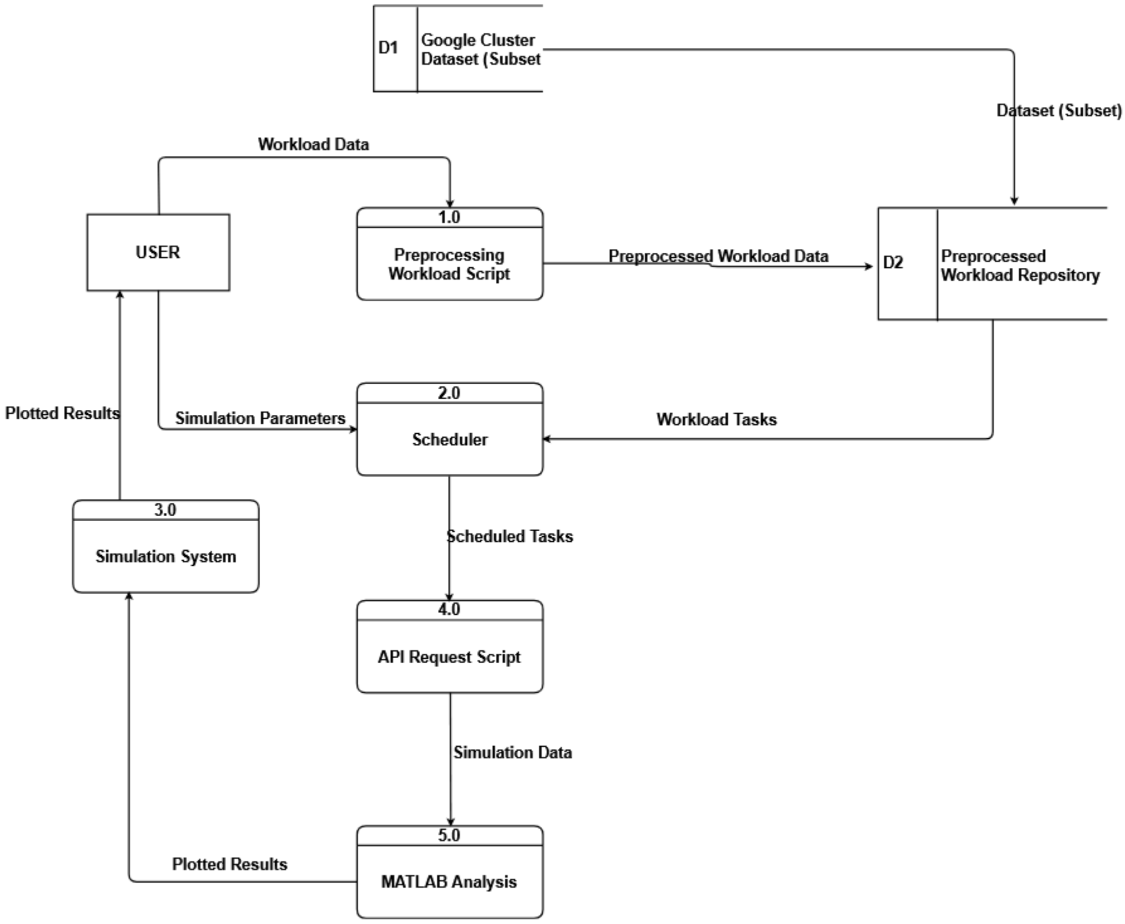

Figure 15. Data Flow Diagram of the Simulation System

This figure illustrates the data flow within the simulation system for comparing Enhanced Particle Swarm Optimization (EPSO) and Enhanced Ant Colony Optimization (EACO) in cloud load balancing. The User initiates the process by providing simulation parameters (e.g., VM settings, algorithm selection) and workload data, which can be custom inputs or selected from the preprocessed subset of the Google Cluster Dataset, prepared during development as a default baseline for evaluating cloud deployment performance. The Preprocessing Workload Script (1.0) preprocesses the user-provided workload data, producing preprocessed user workload data

5stored in the Processed Workload Repository (D2). As the default baseline, the preprocessed Google Cluster Subset is directly stored in the Preprocessed Workload Repository without further preprocessing. The Scheduler (2.0) retrieves the workload tasks from the Preprocessed Workload Repository and assigns them to virtual machines using EPSO or EACO, producing scheduled tasks. These scheduled tasks are passed to the API Request Script (4.0), which generates simulation data (e.g., performance metrics) for analysis. MATLAB Analysis (5.0) processes this simulation data to create plotted results, visualizing response time, resource utilization, energy efficiency, degree of imbalance, and makespan. The final plotted results are returned to the User via the Simulation System (3.0) for review and interpretation.

Instrumentation and Validation

Instrumentation

The researchers will employ a quantitative approach to gather comprehensive data on the performance of cloud-based systems utilizing Enhanced Particle Swarm Optimization (EPSO) and Enhanced Ant Colony Optimization (EACO) as load-balancing strategies. To support this investigation, a combination of research methods will be used—including an internet-based literature review, virtual consultations with cloud specialists and IT experts, and the distribution of structured survey questionnaires.

The simulation system, built on the CloudSim toolkit, will model cloud system performance under realistic workloads using preprocessed subsets of the Google Cluster Dataset. This setup enables the analysis of five key performance metrics: response time, resource utilization, energy efficiency, throughput, and load variance. MATLAB will be used to generate visual representations of the simulation results to support quantitative evaluation and clarity.

To assess the practical relevance of the performance results, the primary instrument for data collection will be a Likert scale-based online survey questionnaire. This tool will be administered to cloud specialists and IT experts to gather expert feedback on the simulation system's functionality and effectiveness. Their input will validate whether the observed algorithm performance aligns with real-world expectations and operational requirements.

Additionally, virtual meetings and consultations will be held with selected professionals to contextualize and expand upon the survey responses. These qualitative insights will enhance the researchers' interpretation of the simulation outcomes, particularly in understanding system usability, deployment feasibility, and algorithm scalability.

The collected data will form the basis for a comparative evaluation of EPSO and EACO. A paired sample t-test will be used to statistically analyze the results, identifying whether there are significant differences between the two algorithms across the selected performance metrics. This statistical evaluation supports data-driven recommendations for optimal workload distribution strategies in cloud environments.

Validation

The questionnaire will first undergo expert review by IT experts to ensure the reliability and validity of the survey tool. These experts will evaluate the content for technical accuracy, clarity, and alignment with the study's objectives. The questionnaire will be subjected to expert assessment by internal validators—specifically, IT professionals with cloud computing and system evaluation experience—to guarantee the validity and dependability of the survey instrument. Based on their feedback, the researchers will make necessary revisions to improve the questionnaire before its final distribution. The researchers will disseminate a validated, in-house questionnaire through Google Forms, which participants can access via a QR code or direct link. Following appropriate approval and informed consent, participants will be chosen based on their availability and willingness to participate. The form includes key details about the researchers, the study's purpose, potential risks and benefits, confidentiality measures, and data privacy protocols to ensure ethical and voluntary participation.

Population and Sampling

The study employs a purposive sampling method, selecting individuals based on their relevance and familiarity with cloud technologies, load-balancing techniques, and system simulations. While the industry-based respondents offer practical, experience-driven viewpoints, academic participants are supposed to provide theoretical insight and research-informed assessment. The combination of experienced professionals and practitioners guarantees a well-to-need assessment approach that reflects both applied industry demands and academic rigor. This structure aims to produce a meaningful interpretation of the system's performance across real-world and research-oriented contexts.

Twenty (20) participants in this study will assess the performance of the simulated load-balancing algorithms using predefined criteria. Respondents from academic and industry sectors are selected to ensure a fair and credible evaluation of the system's functional suitability and performance efficiency.

Specifically, the population includes four (4) academic professionals with backgrounds in computing and systems analysis, five (5) IT experts from the technical team of Tutorials Dojo, and eleven (11) cloud specialists, including one expert advisor from the organization. All respondents from the cloud specialists and academic professionals category hold cloud computing certifications and have at least one (1) year of practical experience, while the selected IT experts possess a minimum of five (5) years of hands-on experience in the field. These participants will evaluate the simulation system's performance through a Likert scale survey assessing key performance indicators such as response time, resource utilization, energy efficiency, degree of imbalance, and makespan upon determining the effectiveness of the compared algorithms.

Evaluation and Scoring

The study uses a Likert-scale approach to evaluate the performance of the proposed load-balancing system for cloud workload simulation. The researchers chose the Likert scale for its effectiveness in quantifying subjective feedback in a structured and standardized way. It enables respondents to indicate how much they agree or are satisfied with particular aspects of the system. such as functionality, usability, and performance, providing valuable insights into user experience and system acceptance. This method is widely used in system evaluations because it simplifies the interpretation of user perception data and enables comparative analysis across multiple evaluation dimensions [43]. The following sections outline the criteria and methodology used to score the system's performance:

POINT	RANGE	DESCRIPTION
4	3.26 – 4.00	Strongly Agree
3	2.51 – 3.25	Agree
2	1.76 – 2.50	Disagree
1	1.00 – 1.75	Strongly Disagree

Table 2. Likert Scale

Table 2 displays the Likert Scale used to assess participants' responses based on their level of agreement with statements related to system performance. The responses were evaluated using median scores to ensure a more accurate and stable representation of the group’s overall sentiment, particularly effective for small respondent groups. The scale includes four response options: 4 (Strongly Agree), 3 (Agree), 2 (Disagree), and 1 (Strongly Disagree). Each point corresponds to a specific range: 3.26–4.00 for Strongly Agree, 2.51–3.25 for Agree, 1.76–2.50 for Disagree, and 1.00–1.75 for Strongly Disagree.

The median is calculated by ordering all the responses and selecting the middle value or by averaging the two middle values when the number of responses is even. This statistical approach reduces the impact of outliers or extreme answers, allowing the researchers to systematically interpret patterns, identify strengths and weaknesses in the system, and derive meaningful insights from the collected data.

Paired T-Test

To determine a statistically significant difference in the performance of two load-balancing algorithms—Enhanced Particle Swarm Optimization (EPSO) and Enhanced Ant Colony Optimization (EACO)—this study employs a paired samples t-test. The evaluation focuses on five primary performance metrics using simulation outputs from multiple preprocessed workloads via CloudSim. The paired t-test is particularly appropriate here, as it compares two algorithms under identical workload conditions, allowing a controlled evaluation of their performance. This approach is validated by the study of Chandrashekar et al. [43], which used paired t-tests to statistically compare simulation results of their proposed HWACOA algorithm against several state-of-the-art task scheduling algorithms. Their use of the paired t-test supports its relevance and applicability in measuring optimization algorithm performance in cloud computing. In our case, the paired t-test will confirm whether observed performance differences between EPSO and EACO are not the result of chance variation and are statistically significant, thereby strengthening the reliability of our conclusions.

$$t = \frac{\overline{X}_d}{\left(\frac{S_d}{\sqrt{n}}\right)}$$

Whereas, $\overline{X}_d$ = mean of the differences between paired values, S_d = standard deviation of the differences, n = number of paired observations.

Ethical Considerations

This study involves multiple layers of ethical responsibility, particularly regarding simulation procedures, dataset usage, and respondent involvement. Since the research utilizes CloudSim to simulate cloud-based workload scenarios, the researchers must ensure they handle all outputs and performance evaluations objectively and transparently. Data from simulations must be stored, interpreted, and reported free from any manipulation or bias that could misrepresent the comparative results of the Enhanced Ant Colony

Optimization and Enhanced Particle Swarm Optimization (EPSO) algorithms. Furthermore, while the Google Cluster Dataset provides realistic workload patterns, the study acknowledges its limitations and avoids overextending claims beyond the dataset's support.

The ethical standards concerning the participants evaluating the system are equally critical. All respondents—including cloud specialists, IT experts, and academic professionals—will participate voluntarily and be fully informed of their feedback's nature, purpose, and use through a digital informed consent form. The researchers will preserve confidentiality and anonymity throughout the study to protect participants' identities and professional insights. However for academic evaluation, their responses will not link to personal identification. These regulations ensure that the survey maintains fairness, respects intellectual contributions and follows research ethics for data use and human interaction.

List of Reference

[1] X. Li et al., "Particle swarm optimization with state-based adaptive velocity limit strategy," arXiv:2308.00936, 2023.

[2] A. Khan et al., "AdPSO: Adaptive PSO-Based Task Scheduling Approach for Cloud Computing," Sensors, vol. 22, no. 3, p. 920, 2022.

[3] J. Zhou *et al.*, “Comparative analysis of metaheuristic load balancing algorithms for efficient load balancing in cloud computing,” *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 12, no. 85, pp. 1–21, Jun. 2023. [Online]. Available: https://doi.org/10.1186/s13677-023-00453-3

[4] A. Daniel et al., "Enhanced Ant Colony Optimization Algorithm for Optimizing Load Balancing in Cloud Computing Platform," IFIP AICT, vol. 611, pp. 64–70, 2021.

[5] M. Zakarya, L. Gillam, A. A. Khan, and I. U. Rahman, "PerficientCloudSim: a tool to simulate large-scale computation in heterogeneous clouds," The Journal of Supercomputing, vol. 77, no. 4, pp. 3959–4013, Sep. 2020, doi: 10.1007/s11227-020-03425-5.

[6] X. Li, K. Mao, F. Lin, and X. Zhang, "Particle swarm optimization with state-based adaptive velocity limit strategy," *Neurocomputing*, vol. 447, pp. 64–79, Mar. 2021, doi: 10.1016/j.neucom.2021.03.077.

[7] T. Zhang and H. Li, "Energy Consumption Reduction Strategy and a Load Balancing Mechanism for Cloud Computing in IoT Environment," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 11, pp. 1-10, 2022.

[4] A. Daniel, N. Partheeban, and S. Sriramulu, "Enhanced Ant Colony Optimization Algorithm for Optimizing Load Balancing in Cloud Computing Platform," in Computational Intelligence in Data Science, Springer, 2021, pp. 64–70.

[8] "Getting Started with Matlab," in *WORLD SCIENTIFIC eBooks*, 2020, pp. 1–18. doi: 10.1142/9789811216039_0001.

[9] L. M. Bokıye and I. A. Ozkan, "Hybrid Load Balancing Policy to Optimize Resource Distribution and Response Time in Cloud Environment," International Journal of Applied Mathematics Electronics and Computers, vol. 10, no. 4, pp. 101–109, Dec. 2022, doi: 10.18100/ijamec.1158866.

[10] Wu, X., Sun, Q., Pu, Z., Zheng, T., Ma, W., Yan, W., Xia, Y., Wu, Z., Huo, M., Li, X. and Ren, W., 2025. Enhancing GPU-Acceleration in the Python-Based Simulations of Chemistry Frameworks. Wiley Interdisciplinary Reviews: Computational Molecular Science, 15(2), p.e70008.

[11] I. Journal, "ReactJS: A Comprehensive Analysis of its features, Performance, and Suitability for Modern Web Development," *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 07, no. 09, Sep. 2023, doi: 10.55041/ijsrem25667.

[12] M. Zeedan, G. Attiya, and N. A. El-Fishawy, "Enhanced Load Balancing Based on Hybrid Artificial Bee Colony with Enhanced β-Hill climbing in Cloud," *Mansoura Eng. J.*, vol. 48, no. 1, pp. 1-10, 2023.

[13] A. Ghosh and S. Saha, "A comparative analysis of metaheuristic load balancing algorithms for cloud computing," Journal of Cloud Computing:

Advances, Systems and Applications, vol. 12, no. 1, pp. 1-18, 2023, doi: 10.1186/s13677-023-00453-3. Available: https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-023-00453-3.

[14] Borg: The next generation - researchgate. 2020. https://www.researchgate.net/publication/341123014_Borg_the_next_generation

[15] K. S. B and S. R, "Load Balancing in Cloud Computing Environment," in *Proc. 2022 IEEE Int. Conf. Women Innov., Technol. Entrepreneurship (ICWITE)*, 2022, pp. 1-5

[16] N. Devi *et al.*, "A systematic literature review for load balancing and task scheduling techniques in cloud computing," *Artificial Intelligence Review*, vol. 57, no. 10, Sep. 2024, doi: 10.1007/s10462-024-10925-w.

[17] Dynamic RL-ACO: Reinforcement learning-based Ant colony optimization for load balancing in cloud networks," IEEE Conference Publication | IEEE Xplore, Sep. 18, 2024. https://ieeexplore.ieee.org/document/10722410

[18] A. Patle and R. Sahu, "Adaptive Resource Allocation and Load Balancing in Cloud Computing Using Firefly and Ant Colony Optimization," Procedia Computer Science, vol. 228, pp. 142–150, 2024. doi: 10.1016/j.procs.2024.01.117.

[19] L. Cao, R. Li, X. Ruan, and Y. Liu, "Defending against Co-residence Attack in Energy-Efficient Cloud: An Optimization based Real-time Secure VM Allocation Strategy," arXiv preprint arXiv:2203.10734, Mar. 2022.arXiv

[20] M. Kumar and A. A. Shinde, "Integrating Particle Swarm Optimization with Cloud Computing Task Scheduling for Optimal Resource Utilization," *Int. J. Innov. Res. Comput. Sci. Technol.*, vol. 12, no. 2, pp. 45-52, 2024.

[21] A. Devaraj, M. Elhoseny, S. Dhanasekaran, E. Lydia, and K. Shankar, "Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization algorithm for energy efficient load balancing in Cloud Computing environments," *J. Parallel Distrib. Comput.*, vol. 142, pp. 36-45, Aug. 2020.

[22] I. Pan, M. A. Elaziz, and S. Bhattacharyya, *Swarm Intelligence for cloud computing*. 2020. doi: 10.1201/9780429020582.

[23] D. Dong, “Agent-based cloud simulation model for resource management,” *Journal of Cloud Computing Advances Systems and Applications*, vol. 12, no. 1, Nov. 2023, doi: 10.1186/s13677-023-00540-5.

[24] Buyya, R., Vecchiola, C., & Selvi, S. T. (2021). Mastering Cloud Computing: Foundations and Applications Programming. Elsevier.

[25] R. Tasneem and M. A. Jabbar, “An Insight into Load Balancing in Cloud Computing,” in Lecture notes in electrical engineering, 2022, pp. 1125–1140. doi: 10.1007/978-981-19-2456-9_113.

[26] Chatterjee, A., & Mukherjee, S. (2022). "Nonlinear inertia weight strategies for improved swarm convergence in cloud load balancing," Applied Soft Computing, 114, 107637.

[27] Luo, H., & Zhang, Y. (2024). "Scalability challenges in cloud computing: A survey of solutions and future directions," ACM Computing Surveys, 56(1), 1-33.

[28] A. K. Pandey and S. Singh, "An Energy Efficient Particle Swarm Optimization based VM Allocation for Cloud Data Centre: EEVMPSO," EAI Endorsed Transactions on Scalable Information Systems, vol. 10, no. 5, Aug. 2023.

[29] Patel, N., & Mehta, R. (2023). "Response time analysis in cloud computing using heuristic approaches," Cloud Computing Journal, 9(3), 23-39.

[30] Ahmed, I., & Khan, A. (2022). "Optimizing resource utilization in cloud computing: A machine learning approach," Future Generation Computer Systems, 127, 67-82.

[31] Y. Liu, W. Li, D. Huang, Y. Li and Y. Zhang, “Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 3743–3755, 2021, doi: 10.1007/s12652-020-02614-7.

[32] PSO algorithm with decreasing inertia weights based on sigmoid function | IEEE conference publication | IEEE Xplore. (2024.-b). https://ieeexplore.ieee.org/document/10691989/

[33] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2021). "CloudSim: A toolkit for modeling and simulation of cloud computing

environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, 51(7), 1256-1275.

[34] Reiss, C., Wilkes, J., & Hellerstein, J. L. (2023). "Google cluster usage traces: Applications and implications," Google Research Papers, 15(4), 102-119.

[35] I. N. Daniel, A. Ezenwoke, and E. Adetiba, "A Comparative Performance Analysis of Task Scheduling Algorithms in Federated Clouds," in *Proc. 2024 Int. Conf. Elect., Comput. Energy Technol. (ICECET)*, 2024, pp. 1-6.

[36] A. Mishra and S. Majhi, "A systematic literature review for load balancing and task scheduling techniques in cloud computing," *Artificial Intelligence Review*, 2024.

[37] Armbrust, M., Fox, A., Griffith, R., & Joseph, A. (2021). "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report No. UCB/EECS-2021-1, University of California, Berkeley.

[38] M. A. Tawhid and K. B. Dsouza, "Particle swarm optimization: A comprehensive survey," PeerJ Comput. Sci., vol. 6, p. e248, 2020.

[39] A. A. Al-Moalmi, J. Luo, A. Salah, and K. Li, "Task scheduling in cloud computing using hybrid meta-heuristic: a two-way approach," Algorithms, vol. 14, no. 2, p. 29, 2021.

[40] J. Tang, G. Liu, Q. Qi, and Q. Pan, "Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments," Symmetry, vol. 16, no. 6, p. 661, 2024.

[41] B. Morales-Castañeda, D. Zaldivar, E. Cuevas, A. Rodríguez, and M. Pérez-Cisneros, "A Novel Diversity-Aware Inertia Weight and Velocity Control for Particle Swarm Optimization," Appl. Soft Comput., vol. 110, p. 107637, 2023.

[42] C. Verwijs and D. Russo, “A theory of scrum team effectiveness,” *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 3, pp. 1–51, Nov. 2022, doi: 10.1145/3571849.

[43] A. Joshi, K. Kale, and D. Chavan, “Use of Likert Scale in a Questionnaire: A Critical Review,” *International Journal of Applied Research in Social Sciences*, vol. 2, no. 4, pp. 01–04, Apr. 2020.

APPENDIX A

COPY OF CONFIDENTIALITY AND NON-DISCLOSURE AGREEMENT (IF PNC IS A RESEARCH SITE-PNC PRE-FO-68 CONFIDENTIALITY AND NON-DISCLOSURE AGREEMENT (NDA) ON STUDENT RESEARCH)

Republic of the Philippines
University of Cabuyao
(Pamantasan Ng Cabuyao)
College of Computing Studies
Katapatan Mutual Homes, Brgy. Banay-banay, City of Cabuyao, Laguna, Phillippines 4025

PNC:AA-LE-286.90.1
April 07, 2025

MR. JON BONSO
Co-founder of Tutorials Dojo
Manila, Philippines

Dear Mr. Bonso,

In accordance with the Data Privacy Act of 2012, we, the undersigned students of University of Cabuyao (Pamantasan ng Cabuyao), hereby pledge to safeguard and maintain the confidentiality of all confidential information obtained during our research endeavors. We commit not to share or disclose this information with any third parties without proper consent.

Furthermore, upon completion of our study, we agree to furnish the Tutorials Dojo with the Ethics Review Clearance issued by the Ethics Review Board as evidence of our compliance with all ethical review procedures mandated by law.

Sincerely,

JOHN DANMEL C. LARANGA
Student Researcher

KIER CHRISTIAN F. REYES
Student Researcher

JAN ALFRED G. VIOLANTA
Student Researcher (CSA7)

Noted by:

ASST. PROF. CHRISTIAN M. BANA
Research Adviser, College of Computing Studies

ASST. PROF. FE L. HABLANIDA
Research Teacher, College of Computing Studies

DR. GIMA B MONTECILLO
Dean, College of Computing Studies

DR. GEORGE V. LAMBOT
Vice President for Academics and Students Services

mcrm/04072025

Dangal ng Bayan.®

APPENDIX E

SHORT REPORT OF PLAGIARISM SOFTWARE

General metrics

79,445	10,813	709	43 min 15 sec	1 hr 23 min
characters	words	sentences	reading time	speaking time

Score

99

This text scores better than 99% of all texts checked by Grammarly

Writing Issues

1		1
Issues left	Critical	Advanced

Plagiarism

This text seems 100% original. Grammarly found no matching text on the Internet or in ProQuest's databases.

APPENDIX F

REPORT OF LANGUAGE SOFTWARE

General metrics

79,445	10,813	709	43 min 15 sec	1 hr 23 min
characters	words	sentences	reading time	speaking time

Score

99

This text scores better than 99% of all texts checked by Grammarly

Writing Issues

1	✓	1
Issues left	Critical	Advanced

Plagiarism

✓ This text seems 100% original. Grammarly found no matching text on the Internet or in ProQuest's databases.

APPENDIX G

CURRICULUM VITAE OF STUDENT RESEARCHERS

PNC:AA-FO-27 rev.0 02012023

JOHN DANMEL C. LARANGA

Student leader, project manager, and graphic designer with experience in leading organizations, managing projects, and creating impactful designs. Skilled in team coordination, strategic planning, and visual storytelling to drive engagement and results.

CONTACT

larangajohndanmel31@gmail.com
+63 956 073 5626
Blk 187 Lot 16, Mamatid, Cabuyao, Laguna, Phase 2 Mabuhay City Philippines

EXPERTISE

- Basic Java, Python, React, Ruby, PHP Programming
- Public Speaking
- Project Implementation and Development
- Resource Management
- Formal Writing
- Organizational Leadership
- Partnership Building & External Relations

REFERENCES

Kate Callao
Vice President and Chief Evangelist | AWS Cloud Club Philippines
katedanielle03@gmail.com

Rob Fritz Abayari
Vice President | College of Computing Studies – Student Government
abayarirobfritz31@gmail.com

EDUCATION

Bachelor Program 2022 - 2026
University of Cabuyao (Pamantasan ng Cabuyao)
- Certificate of Completion -Wadhwani Opportunity's 21st Century Core Employability Skills Program.

Secondary 2022
St. Ignatius Academy – Cabuyao Campus
- With Honors

Elementary 2015
Mamatid Elementary School
- N/A

WORK EXPERIENCE

Secretary 2025 - 2026
DevCon Laguna Chapter
- Supported partnership initiatives by drafting formal communications, contributing to the successful launch of the "SMART CONTRACT CODE CAMP 2025" with PUP Sta. Rosa Campus.

Director 2024 - 2026
AWS Cloud Club Philippines – Resource Department
- Effectively managed resources, created sponsorship proposals, implemented organizational databases, and led workforce coordination to ensure the success of programs and partnerships.

Administrative Assistant 2025
Unreal Community Philippines – Metro Manila Chapter
- Secured partnerships with various university academic organizations across the Philippines, managed formal communications for collaborations, and handled essential administrative paperwork.

President 2024 - 2025
College of Computing Studies – Student Government
- Spearheaded academic-focused programs to support student development, while overseeing organizational paperwork and documentation to ensure smooth operations.

University Captain & Founder 2024 - 2026
AWS Cloud Club – University of Cabuyao
- Pioneered access to external learning platforms and certification programs, connecting students to AWS resources, industry professionals, and opportunities beyond the curriculum to enhance technical skills and career readiness.

LATEST SEMINAR ATTENDED

AWS User Group Philippines
AWS User Group Philippines Community Day 2024
- September 22, 2024

APPENDIX G

CURRICULUM VITAE OF STUDENT RESEARCHERS

PNC:AA-FO-27 rev.0 02012023

Reyes, Kier Christian F.

Kier Christian F. Reyes, born in Binan, Laguna in 2003, is a dedicated and aspiring Data Scientist with a passion for cloud computing. Currently pursuing a Bachelor of Computer Science at the University of Cabuyao (UCPnC). He has already made notable strides in the tech field including passing the AWS Cloud Practitioner Exam (CLF-C02).

CONTACT

reyeskierchristian64@gmail.com

09495873031

Lt. 137 Bigaa, Cabuyao Laguna

EXPERTISE

- AWS
- Kubernetes
- Java
- Python

REFERENCES

Marilyn Reyes
Production Worker | BYPPO
09202794036

Jojo Francisco
Electrician | SMC Slex Inc.
09202794036

EDUCATION

Bachelor of Computer Science 2022-Present
University of Cabuyao (PNC)
- AWS Certified Cloud Practitioner
- Dean Lister 2nd Yr 1st Semester
- Google Data Analytics Professional
- DataCamp Donates Scholar

Secondary Education 2021
Bigga Integrated National High School
- Graduated with High Honors

Elementary Education 2015
Dita Elementary School
- Graduated with Honors

WORK EXPERIENCE

Ushers Committee 2024-Present
PAWS Cloud Club University of Cabuyao

Intern (Junior Officer) 2021
City Information Technology Office Sta. Rosa
- Gain knowledge and experience how to use VB macros and Power Query with Excel.
- Experience in basic to advance networking through series of seminars and practical assessment.

LATEST SEMINAR ATTENDED

Organization

AWS Philippines Community Day 2024
- September 22, 2024

- JBECP Inaugural
- September 14, 2024

APPENDIX G

CURRICULUM VITAE OF STUDENT RESEARCHERS

PNC:AA-FO-27 rev.0 02012023

JAN ALFRED G. VIOLANTA

Certified AWS Cloud Practitioner and student officer with hands-on experience in cloud initiatives. Actively participates in programming competitions outside of university, showcasing strong problem-solving and individual skills.

CONTACT

violantajanalfred40@gmail.com

+639945349486

BLK 23 LOT 32, Centennial Townhomes, San Isidro, Cabuyao, Laguna

EXPERTISE

- AWS Cloud Computing
- Programming (Java)
- Competitive Coding
- Internal Communications
- Leadership in Tech

REFERENCES

Roselyn G. Goyada
Infrastructure Analyst | CITIBANK N.A. ROHQ
+639566084935

EDUCATION

Bachelor of Science in Computer Science 2022-2026
University of Cabuyao (Pamantasan ng Cabuyao)
- ACSS Computer Science Cup 1 Programming Competition - Junior Division Champion
- ACSS Code Battle: Anti-Hackathon Programming Competition - Champion
- VPAA Lister - 2nd Year 1st Semester
- Deans Lister - 1st Year 2nd Semester
- Top 1 overall in Computer Science 2nd Year 1st Semester
- Top 4 overall in Computer Science 1st Year 2nd Semester

Secondary Education
University of Cabuyao (PnC) - SHS 2020-2022
- With High Honor
- Best in Entrepreneurship
- Best in Research
- Best in Science Investigatory Project

Liceo De Mamatid 2016-2020
- With Honor

Elementary
Pulo Elementary School 2015-2016

WORK EXPERIENCE

Information Officer 2024 - Present
AWS Cloud Clubs Philippines
- Managed nationwide information needs and handled internal communications for the organization.

Information Head Officer 2024 - Present
AWS Cloud Club - University of Cabuyao
- managed internal communications and helped promote AWS events and resources to students.

LATEST SEMINAR ATTENDED

AWS Cloud Clubs Philippines
AWS Cloud Clubs Philippines Student re:Invent re:Cap 2025
- March 29, 2025

AWS User Group Philippines
AWS User Group Philippines Community Day
- September 22, 2024