Journal of IoT in Social, Mobile, Analytics, and Cloud (ISSN: 2582-1369)
www.irojournals.com/iroismac/

Performance Analysis of Various Scheduling Algorithms using CloudSim

Pratima Chapagai[1], Subarna Shakya[2]

Department of Electronics and Computer Engineering, Pulchowk Campus, Tribhuvan University, Lalitpur, Nepal

E-mail: [1]pchapagain28@gmail.com, [2]drss@ioe.edu.np

Abstract

Cloud computing technologies have quickly changed how companies and organizations manage their IT resources. The core of this transformation has evolved as cloud datacenters, which offer scalable and affordable options for hosting and administering a variety of applications and services. One information technology typology that has been widely employed to deliver a range of services via the Internet is cloud computing. It guarantees simpler access to premium services and resources. Cloud systems' operation needs to be planned in order to effectively deliver services to individuals. Task scheduling seeks to maximize system throughput and distribute diverse computational resources to software programs. The unpredictability of the scenario grows as the task and has a strong potential for successful resolution. The study begins with an experimental setup to analyse the various performance metrics of task scheduling algorithms. Every experiment has several important stages. To replicate scenarios found in the real world where jobs are divided across many computing resources, the tasks are assigned to available data centers. A number of experiments were carried out to analyse the performance of First Come First Service (FCFS), Shortest Job First (SJF), Round Robin (RR) and Particle Swarm Optimization (PSO) scheduling algorithms using the parameters: makespan, average completion time, average waiting time, and average cost consumption. Thus, this study provides a description of task scheduling and the performance analysis of algorithms to task scheduling that is employed in cloud computing environments.

Keywords: Cloud Data Centers, FCFS, SJF, RR, PSO, CloudSim

DOI: https://doi.org/10.36548/jismac.2023.4.001
Received: 03.10.2023, received in revised form: 23.10.2023, accepted: 09.11.2023, published: 21.11.2023

1. Introduction

Cloud computing refers to the delivery of on-demand computing resources, including data storage, processing power, and software applications, over the internet. Instead of relying on local servers or personal devices, users can access and utilize these resources through a network connection. Cloud computing offers several advantages, such as scalability, flexibility, and cost efficiency. It allows users to easily scale up or down their computing resources based on their needs, without the need for physical infrastructure upgrades. Additionally, cloud computing provides flexibility by enabling users to access their applications and data from any device. Cloud computing has significant advantages that extend across various domains. The cloud data centers and employees are frequently requested by the clients for their cycle. It generates significant traffic in both the server farm and cloud worker. Due to the heavy traffic and potential performance degradation caused by unexpected setbacks, task scheduling becomes unbalanced. In order to recognize the virtual machine disappointment, some precautions must be performed. It is difficult to anticipate virtual machine disappointment in this way; this work suggests the appropriate scheduling algorithm among the existing ones.

Four well-known scheduling algorithms, First Come First Service (FCFS), Shortest Job First (SJF), Round Robin (RR) and Particle Swarm Optimization (PSO) are examined in this study. Four parameters are considered for analysing the performance of these algorithms. These parameters are: makespan, average waiting and completion times, and average cost. The major aim of the study is to specify the appropriate scheduling algorithm considering the makespan values, average waiting and completion times and lower cost consumption.

1.1 Theoretical Background

An important and crucial subject in resource management is task scheduling. It makes reference to the effective scheduling of computer work and resource allocation within certain limitations in the infrastructure environment services suitable for cloud computing. The various types of task scheduling algorithms used in this work are:

(i) FCFS Algorithm

The First-Come-First-Serve (FCFS) scheduling algorithm, commonly referred to as First-In-First-Out (FIFO), works on the simple principle of handling jobs in the order they arrive. In the tests carried out, FCFS demonstrated very consistent performance in a variety of settings, making it a desirable option for simple task management. Small to medium-sized task

sets were used in settings where FCFS consistently produced competitive makespan and average completion times.

The default scheduling technique, FCFS, prioritized jobs in the order they were received. This algorithm was chosen due of its fairness and ease of use.

(ii) Shortest Job First Algorithm

In order to reduce overall job processing time, the Shortest Job First (SJF) scheduling algorithm is built to prioritize activities depending on how long they will take to complete. In the trials, SJF showed astounding efficiency, especially in situations with condensed work sets. It regularly had the quickest average completion time and makespan, which made it a viable candidate for latency sensitive applications. However, when the job set expanded in size, SJF had trouble keeping up its effectiveness.

SJF was chosen because it can reduce average completion times by giving preference to activities with faster execution durations. The goal of this algorithm was to speed up task completion.

(iii) Round Robin Algorithm

Round Robin (RR) scheduling, which gives each activity a set time slice in a cyclical way, is renowned for its fairness and simplicity [1]. In the test carried out, RR showed adaptability in handling various conditions. It was an appealing option for scenarios with mixed-size task sets since it consistently produced balanced average completion times across a range of job sizes. One of the most effective algorithms, round robin's key difficulty is selecting the right time quantum. The selection of this parameter affects Round Robin's effectiveness [2]. However, the time quantum of RR had a significant impact on how well it performed, with longer time slices producing less effectiveness.

To test its ability to strike a balance between fairness and efficiency RR was chosen. Each task was given an equal amount of CPU time by cyclic scheduling, preventing potential hunger.

(iv) Particle Swarm Optimization Algorithm

A population-based optimization approach known as Particle Swarm Optimization (PSO) has been applied in a number of sectors to solve optimization and search issues. PSO can be used to optimize different aspects of cloud-based resources and applications when it comes to computing. The distribution of virtual machines (VMs) among physical servers in a cloud data center can be optimized using PSO. While maintaining the performance standards of hosted applications, the objective is to reduce operational expenses, maximize resource utilization, and minimize energy consumption. PSO can make tasks or jobs in a cloud computing environment more efficient to schedule. This is crucial in situations where several tasks vie for the same resources. PSO can assist with job scheduling to reduce completion times, energy use, or cost.

2. Related Work

Cloud computing has emerged as a promising paradigm for delivering on-demand computing resources and services over the internet. To evaluate and analyse the performance and behaviour of cloud systems, researchers have extensively used simulation frameworks like CloudSim. CloudSim is a tool that provides researchers, developers, and cloud service providers with a powerful tool to evaluate and analyse the performance of various cloud-based algorithms, policies, and architectures. CloudSim, a famous tool is actually used as a toolkit for simulation of cloud scenarios [3]. With CloudSim, users can create virtualized cloud infrastructures, define different cloud services, and simulate the execution of applications and workloads. CloudSim allows users to gain a comprehensive understanding of cloud scenarios without having to be concerned about the intricacies of low-level implementation [4]. This kind of dynamic environment necessitates effective load balancing and task scheduling to reduce response times, completion times (makespan), energy use, and service interruptions. In [5] conducted a comparative analysis and proposed that selecting virtual machines in good condition can reduce response time without compromising cost. [6] an extensive overview of different load balancing techniques in cloud environments, including static, dynamic, and nature-inspired approaches is presented. The aim of these techniques is to tackle challenges related to data center and improve overall system performance. Scheduling algorithms play a crucial part in making sure the right resource is available for every request since cloud resources should be used properly [7]. In the context of CloudSim, a data center represents a simulated infrastructure that serves as a central hub for hosting and managing cloud resources.

The objective of [8] was to assess the simulation process of cloud computing tasks and resource allocation strategies within the CloudSim environment. All cloud providers have a common problem of work scheduling; hence having a good task scheduling algorithm is essential for processing which was addressed in [9]. The load between each resource in the cloud computing system was balanced using balance scheduling approaches to achieve and improve the optimized results in [10]. Hosts play a crucial role in the simulation of cloud computing environments as they serve as the foundation for running virtualized environments and executing cloud-based applications. [11]Combined a load balancing system with a pre-migration algorithm for cloudlet scheduling. The proposed study is implemented using CloudSim simulation framework and accompanying tools. It is crucial to use an efficient scheduling strategy that may provide minimal average waiting times and maximum resource utilization [12].

3. Materials and Methods

The related resources and procedure for scheduling tasks in a cloud computing environment are covered in this section. Both consumers and cloud service scheduling algorithms, which are thought to be appropriate for large-scale systems, need to improve their make span and make use of resources as effectively as possible [13]. A brief explanation of the cloudSim toolkit's features is provided in paper [14]. CloudSim is a tool that provides researchers, developers, and cloud service providers with a powerful tool to evaluate and analyse the performance of various cloud-based algorithms, policies, and architectures.

(i) Experimental Setup

We created a comprehensive experimental setup to assess the effectiveness of various job scheduling algorithms in cloud computing settings. Three alternative task load scenarios—200 tasks, 2000 tasks, and 20,000 tasks—were taken into consideration when creating synthetic computing tasks to match real-world cloud workloads. These distinct job loads were designed to mimic the dynamics of cloud systems at various demand levels.

The data volume was added as a variable to the analysis to take the effect of data volume into account. To model scenarios with low, medium, and high data processing requirements, we constructed datasets of various sizes, all with file sizes of 300 MB. This gave us the chance to explore the complex connection between algorithm performance and data volume.

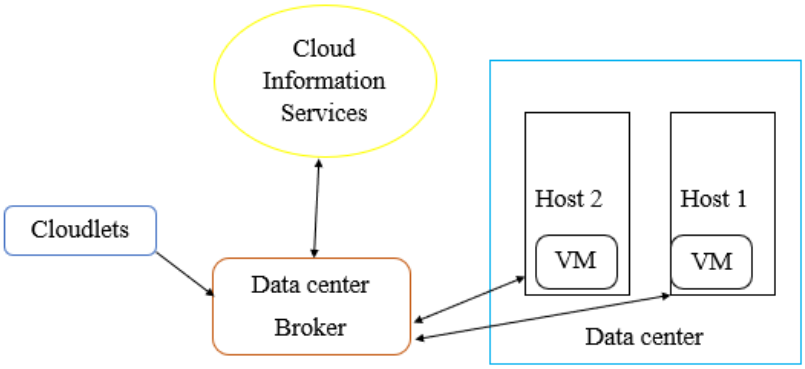

Figure 1. System Architecture

The Figure 1 above shows the system architecture which consist cloudlets, broker, datacentre, cloud information services, hosts and virtual machines. These components are simulated using cloudSim simulator version 3.0. In cloudSim, a data center is a fundamental component of the simulation framework and provides the necessary infrastructure for deploying and executing virtualized environments. While the hosts are configurable entities with various parameters that can be set, such as the number of processing cores, processing capacity, memory capacity, storage capacity, and bandwidth. Hosts in cloudSim can accommodate multiple virtual machines (VMs) that share their resources. Similarly, in the cloudSim simulation framework, virtual machines (VMs) represent the simulated instances of virtualized computing environments within a cloud data center. VMs can be dynamically provisioned, allocated, and scheduled on the available hosts within the data center.

The cloudlet will be created with some parameters including id, length, file size and output size as per requirement. The creation of data center will include the parameters as Os, cost, cost per mem, cost per storage and cost per bw. The Cost refers to the numerical value associated with the expenditure of using processing power in this resource. The cost per mem represents the numerical value associated with the cost of utilizing memory in this resource. The Cost per storage indicates the numerical value associated with the cost of using storage in this resource. The cost per bw represents the numerical value associated with the cost of utilizing bandwidth in this resource. The host will be created with parameters as host id, storage, RAM, bandwidth and processors. The host id can be a numeric or textual value that is solely used for identification purposes. The storage details and RAM are numeric values indicating the capacity, measured in megabytes (MB).The creation of a virtual machine includes several parameters such as VM id, size of VM image, RAM, bandwidth (bw), and

processor count. The VM id (Virtual Machine Id) can be a numeric or textual value used solely for identification purposes.

The parameter of simulation environment includes: twenty data centers, each having 512 MB of RAM, 1000 MIPS, and a bandwidth of 1000 Mbps, were used to simulate the cloud infrastructure. The Xen virtual machine monitor (VMM) was chosen as the virtualization technology to manage virtual machines effectively, and tasks were assigned to a single processing element (pesNumber: 1).

(ii) Experiment Execution

Each experiment included a number of crucial phases. In order to simulate real world situations where tasks are distributed across different computer resources, first, synthetic workloads were randomly assigned to the 20 data centers. The trials were then carried out with various dataset sizes, spanning from small to large data volumes, to assess the algorithms' capacity to adapt to changing data processing requirements.

A wide range of performance measures, including make span, total completion time, average completion time, total cost, average cost, average waiting time, and average turnaround time were measured in order to conduct a thorough analysis. These measurements made it possible to evaluate the algorithms in a variety of contexts. The scheduling process is depicted in Figure.2 below.

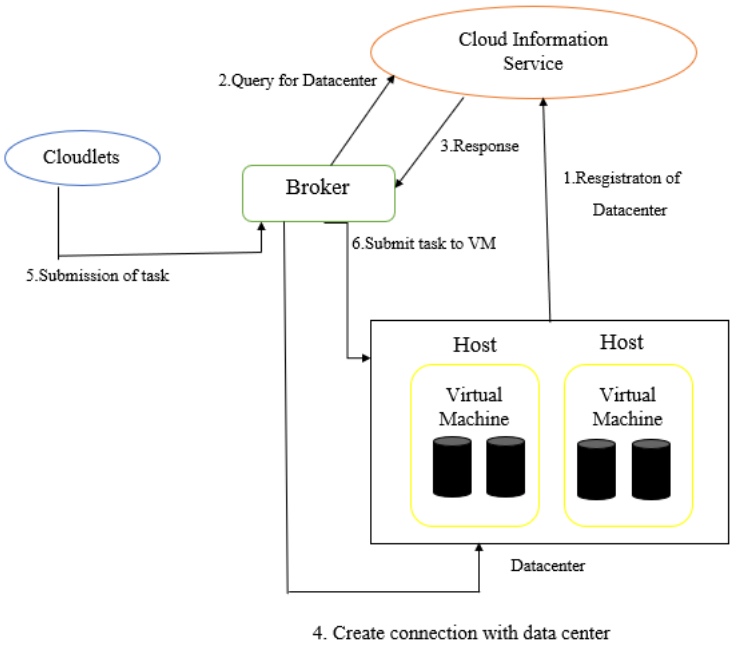

Figure 2. Scheduling Process

The scheduling performed during the experimental procedure involves the following steps:

i. The initialization process begins with the launch of data center.

ii. Creation of virtual machines.

iii. Submission of cloudlets to broker. Broker can now have access to cloudlets.

iv. Cloudlets are redirected to virtual machines for execution.

v. After successful execution of the cloudlets, the virtual machines are destroyed.

vi. Finally, the required information about the processed cloudlets, which includes make span, average waiting time, average completion time and average cost is got.

(iii) Data Collection and Analysis

The studies generated a large amount of data, including task execution times, completion times, and associated metrics. Each and every data point was carefully documented for further study to glean relevant insights from the dataset. The Table .1-4 illustrates the makespan, average completion time, average cost, and average waiting time for scenario 1 with 200 tasks and population size 250, scenario 2- with 2000 tasks, and population size: 250, Scenario 3 with tasks: 2000, population size: 25, scenario 4- with tasks: 20000, population size: 2500 respectively.

Table 1. Scenario 1- Number of Tasks: 200, Population Size: 250

Algorithm	Makespan	Average Completion time	Average Cost	Average Waiting time
FCFS	10182.86	3287.08	1674.37	2728.96
SJF	9084.51	3377.04	1659.32	2823.93
RR	8560.5	3539.04	1795.2	2940.64
PSO	9114.78	2605.44	1068.78	2249.18

Table 2. Scenario 2- Number of Tasks: 2000, Population Size: 250

Algorithm	Makespan	Average Completion time	Average cost	Average waiting time
FCFS	7687.98	5832.37	183.88	5771.07
SJF	54744.89	5590.04	176.49	5531.2
RR	53059.65	5492.67	170.96	5435.68
PSO	9115.78	5817.06	109.6	5870.53

Table 3. Scenario 3- Number of Tasks: 2000, Population Size: 25

Algorithm	Makespan	Average Completion time	Average cost	Average waiting time
FCFS	8559.24	5228.02	164.54	5173.17
SJF	43990.35	568.69	173.94	5550.93
RR	51101.32	5457.36	172.31	5399.93
PSO	8074.91	5013.54	124.68	4971.97

Table 4. Scenario 4- Number of Tasks: 20000, Population Size: 2500

Algorithm	Makespan	Average Completion time	Average cost	Average waiting time
FCFS	10396.58	5782.23	17.76	5776.31
SJF	64603.85	5976.4	18.44	5970.59
RR	83996.29	6035.21	18.54	6029.03
PSO	8257.26	7298.42	9.67	7295.19

(iv) Graphical Representation

Bar graphs were used as visualization techniques to make the results more understandable. These graphical depictions provided an easy-to-understand picture of how each scheduling method performed in various scenarios, such as those with differing job loads and data volumes.

4. Results and Discussion

On the implementation of this work on cloudSim, The following results were observed based on the evaluations of the algorithms: FCFS, SJF, RR and PSO. The interpretations of the results are mentioned below.

(i) Interpretation of Results

To find patterns and trends in the performance of the algorithm, the collected results were carefully contrasted and examined. The responses of each method to changes in job load and data volume were examined. Significant discoveries were looked into in order to better understand each algorithm's advantages and disadvantages in various cloud computing contexts.

Popular Python module Matplotlib is used to produce the following visualizations in the form of bar graph as it offers a sophisticated user interface for creating visually appealing and instructive plots and charts.

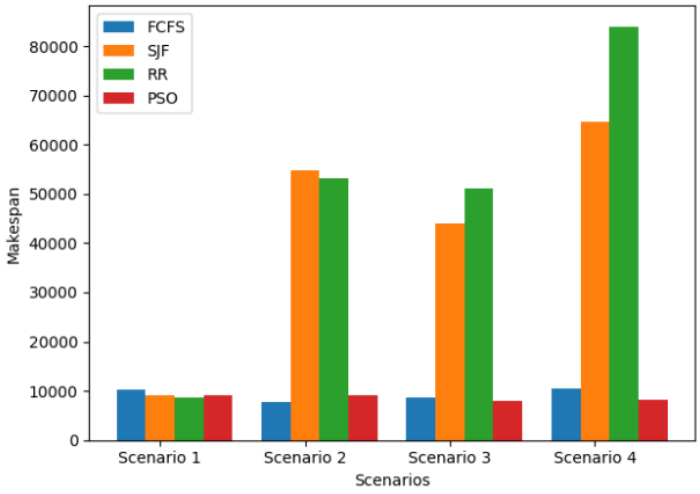

Figure 3. Comparison of Makespan of Algorithms

Makespan has significantly increased in a number of circumstances by incorporating Particle Swarm Optimization (PSO) into traditional scheduling techniques. Even though PSO might not always result in the shortest makespan, its use in the scheduling process has shown to significantly speed up job completion times. This method significantly improves work sequencing and resource allocation, leading to quicker and more effective job execution.

PSO's ability to adapt and fine-tune scheduling decisions is demonstrated by its incorporation as an optimization layer within common scheduling techniques. This strategy makes use of both the advantages of conventional scheduling techniques and the optimization powers of PSO, resulting in more competitive makespan values and increased overall task scheduling effectiveness.

In conclusion, the addition of Particle Swarm Optimization to traditional scheduling techniques has shown to be advantageous, resulting in appreciable increases in makespan values and enhancing scheduling optimization. By optimizing current scheduling frameworks, this method increases their effectiveness and responsiveness to a variety of scheduling demands.

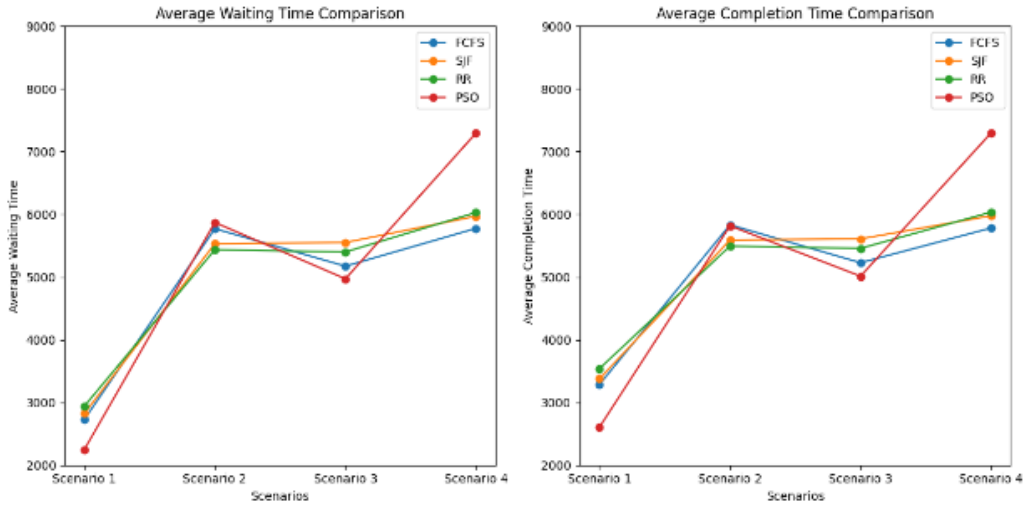

Figure 4. Comparison of Average Waiting and Completion Time of Algorithms

The Particle Swarm Optimization (PSO) was used to optimize scheduling, concentrating on average waiting time and average completion time across a range of cases and algorithms. In Scenario 4, where the task's size is greatly increased to 20,000 tasks and the population is 2,500, a slight increase was observed in these measures while using PSO.

It's crucial to keep in mind, though, that PSO is designed for optimization and aims to find a balance between multiple goals. The modest increase in waiting and completion times in this specific case can be attributed to PSO's attempts to optimize other important factors, like cost. These minor variations are clearly within acceptable bounds, demonstrating PSO's adaptability in handling various scheduling difficulties.

In summary, the journey in optimizing scheduling using PSO demonstrates its effectiveness in adapting to different task and population sizes. While minor variations emerge in specific scenarios, they reflect the deliberate optimization process rather than any shortcomings, reinforcing the process of PSO in efficient task scheduling.

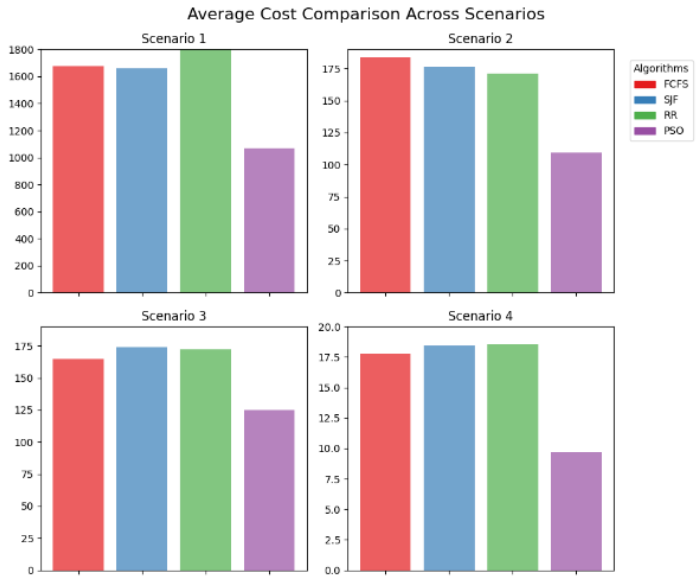

Figure 5. Analysis of Average Cost

The effectiveness of using Particle Swarm Optimization (PSO) for scheduling optimization is examined in this cost comparison plot across four different scenarios. PSO consistently performs better than conventional scheduling algorithms like FCFS (First-Come-First-Served), SJF (Shortest Job First), and RR (Round Robin) in all cases. This finding highlights PSO's potential to reduce average costs, making it an appealing option for effectively scheduling activities.

The benefit of PSO becomes more and clearer in each scenario from scenario 1 to scenario 4, which has an increasing number of jobs. PSO performs in cost reduction even in the most difficult scenario 4 with 20,000 tasks, demonstrating its scalability and adaptability. This shows that PSO offers a reliable alternative for streamlining scheduling procedures and generating cost savings when dealing with intricate and substantial workloads.

In conclusion, this graphic shows how PSO may be successfully used as a schedule optimization method for a range of workloads. It shows that PSO regularly lowers average costs when compared to conventional algorithms, making it an important tool for businesses looking to improve scheduling effectiveness and cut operating costs.

5. Analysis and Implications

The ramifications of the findings for cloud-based task scheduling were covered in detail in the discussion section. The trade-offs between algorithm fairness, scheduling decisions, and algorithm efficiency are discussed. Additionally, based on the findings of the research, useful suggestions for choosing appropriate scheduling algorithms are also offered.

(i) Cost Optimization

This study shows PSO's effectiveness in reducing scheduling expenses. PSO regularly beats FCFS, SJF, and RR, as seen by the cost comparison plot for the four situations. For companies looking to cut operational costs while retaining scheduling effectiveness, this result is of utmost significance.

(ii) Scalability and Adaptability

PSO's advantage becomes more obvious from the test carried out from scenario 1 to scenario 4, where there is a significant rise in the number of tasks. Even in the most difficult circumstances, it excels in cost reduction, demonstrating its scalability and adaptability. This is a crucial realization for businesses managing, increasing workloads, and challenging scheduling demands.

(iii) Practical Implementation

The study emphasizes the viability of using PSO for schedule optimization outside of academic inquiry. The research fits with real-world circumstances and resonates with practitioners looking for answers to operational difficulties by framing the findings as an effort to optimize scheduling operations rather than just the simple execution of an algorithm.

6. Conclusion

In conclusion, this research clarifies the complex interplay between scheduling methods, data volume, and task completion timeframes. It emphasizes the necessity for

customized scheduling methods to meet the various demands of contemporary cloud computing systems. By providing a useful assessment of the scheduling algorithms FCFS, RR, PSO and SJF across diverse task volumes and data sizes, this research work makes a contribution to the field of cloud computing. It offers insightful information on how well the algorithms work in actual use, especially in terms of makespan, typical completion time, and cost effectiveness. Additionally, this study offers convincing proof that Particle Swarm Optimization (PSO) is a promising strategy for increasing scheduling effectiveness and lowering operational expenses. PSO is a great asset for firms in a variety of industries due to its consistent superiority to conventional algorithms and scalability. The results of this research provide a road map for utilizing PSO to effectively enhance scheduling procedures as firms deal with the ever-increasing demands of scheduling jobs efficiently.

References

[1] Balharith, Taghreed, and Fahd Alhaidari. "Round robin scheduling algorithm in CPU and cloud computing: a review." In 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), pp. 1-7. IEEE, 2019.

[2] Fiad, Alaa, Zoulikha Mekkakia Maaza, and Hayat Bendoukha. "Improved version of round robin scheduling algorithm based on analytic model." International Journal of Networked and Distributed Computing 8, no. 4 (2020): 195-202.

[3] Wu, He-Sheng, Chong-Jun Wang, and Jun-Yuan Xie. "Terascaler elb-an algorithm of prediction-based elastic load balancing resource management in cloud computing." In 2013 27th International Conference on Advanced Information Networking and Applications Workshops, pp. 649-654. IEEE, 2013.

[4] Belkhouraf, Mohamed, Ali Kartit, Hassan Ouahmane, Hamza Kamal Idrissi, Zaid Kartit, and Mohamed El Marraki. "A secured load balancing architecture for cloud computing based on multiple clusters." In 2015 International Conference on Cloud Technologies and Applications (CloudTech), pp. 1-6. IEEE, 2015.

[5] Gangwar, Rakesh Chandra. "Comparative study of load balancing algorithms in cloud environment." International Journal on Recent and Innovation Trends in Computing and Communication 2, no. 10 (2014): 3195-3199.

[6] Shafiq, Dalia Abdulkareem, N. Z. Jhanjhi, and Azween Abdullah. "Load balancing techniques in cloud computing environment: A review." Journal of King Saud University-Computer and Information Sciences 34, no. 7 (2022): 3910-3933.

[7] AlMansour, Njoud, and Nasro Min Allah. "A survey of scheduling algorithms in cloud computing." In 2019 International Conference on Computer and Information Sciences (ICCIS), pp. 1-6. IEEE, 2019.

[8] Su, Yue. "Virtual machine allocation strategy based on cloudsim." In International Conference on Multi-modal Information Analytics, pp. 455-463. Cham: Springer International Publishing, 2022.

[9] Nayak, Ankitha A., and Shashank Shetty. "A Systematic Analysis on Task Scheduling Algorithms for Resource Allocation of Virtual Machines on Cloud Computing Environments." In *2023 International Conference on Recent Trends in Electronics and Communication (ICRTEC)*, pp. 1-6. IEEE, 2023..

[10] Chaudhary, Divya, and Bijendra Kumar. "Analytical study of load scheduling algorithms in cloud computing." In *2014 International Conference on Parallel, Distributed and Grid Computing*, pp. 7-12. IEEE, 2014..

[11] Nanthiya, D., P. Keerthika, S. B. Gopal, and V. Gokul. "Effective pre-migration mechanism for dynamic load balancing in cloud computing environment." In IOP Conference Series: Materials Science and Engineering, vol. 1055, no. 1, p. 012098. IOP Publishing, 2021.

[12] Sahoo, Bidush Kumar, Anita Sardana, VikasSolanki, Sunil Gupta, and Kamal Saluja. "Novel approach of diagnosing significant metrics of load balancing using cloudsim." In 2022 10th International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing (ICETET-SIP-22), pp. 1-6. IEEE, 2022.

[13] Vishnoi, Sushant Kumar, and Sanjeev Patel. "Comparison of Average Completion Time and Makespan for Various Task Scheduling Techniques." In 2023 4th International Conference on Computing and Communication Systems (I3CS), pp. 1-6. IEEE, 2023.

[14] Buyya, Rajkumar, Rajiv Ranjan, and Rodrigo N. Calheiros. "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities." In 2009 international conference on high performance computing & simulation, pp. 1-11. IEEE, 2009.

Author's biography

Pratima Chapagai is a dynamic individual whose passion for innovation and technology has been the driving force behind her academic pursuits. She holds a Bachelor's degree in Electronics and Communication Engineering from Purwanchal Campus, Tribhuvan University. She has established herself as a proficient engineer with a keen interest in cutting-edge advancements in the field. Currently, Pratima is furthering her academic journey by pursuing a Master's degree, where she continues to explore and contribute to the ever-evolving landscape of electronics and communication.Throughout her academic career, Pratima has demonstrated a strong commitment to excellence, earning accolades for her academic achievements and problem-solving abilities. Her dedication to staying abreast of the latest developments in technology has defined her academic success.

Prof. Dr. Subarna Shakya is computer engineering professor with a track record of accomplishments in the IT and services sectors. He is competent in databases, software design, E-government system , Information systems project management, and information system and cloud computing and security. He is one of the dedicated engineers with a Doctor of Philosophy (PhD) from Lviv Polytechnic National University in Computer Engineering.

Prof. Dr. Subarna Shakya is a professor of Computer Engineering at Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Nepal. He has served as Head of Department, Executive director (National IT Center), Asst. Dean (Institute of Engineering) , published over 150 scientific/technical articles and 5 books. He has been serving as an Editor/Guest Editor for over 15 internationaljournals.He was awarded 100 most dedicated professors from world education congress.