Journal of Namibian Studies, 35 (2023): 293-307 ISSN: 2197-5523 (online)

Special Issue on Engineering, Technology and Sciences

Improving Efficiency And User Satisfaction In Cloud Computing Through Task Scheduling: A Comparision Among Different Scheduling Alorithms

Hemanta Dey [1] , Dr. Akash Saxena [2]

[1] Research Scholar, Department of Computer Science , Mansarovar Global University, Sehore, M.P., India.
[2] Research Guide, Department of Computer Science , Mansarovar Global University, Sehore, M.P., India.

ABSTRACT

This study explores task scheduling algorithms in the context of cloud computing environments, considering multiple factors. Efficient task scheduling is crucial for optimizing resource utilization and improving system performance in the cloud. We investigate various factors, such as task priority, resource availability, and load balancing, and their impact on scheduling algorithms. Our research contributes to a deeper understanding of how to design and implement effective task scheduling strategies that meet the diverse requirements of cloud computing systems, leading to enhanced overall efficiency and user satisfaction. In this study, methods for scheduling tasks based on priority were studied in relation to virtual machines and tasks. This method achieves satisfactory results when it comes to balancing the load, however it does not perform well in terms of cost performance. A comparative analysis of the many different scheduling methods has also been included in this research and was carried out using the Cloudsim simulator.

Keywords: - Algorithm, Cloud Computing, Scheduling, Cloudsim, Virtual machine tree.

I. INTRODUCTION

Task scheduling in cloud computing environments is a critical aspect that directly influences the efficiency, cost-effectiveness, and overall performance of cloud-based services and applications. To optimize task scheduling, various algorithms have been developed, each taking into account

multiple factors to ensure that computational resources are allocated and utilized in the most efficient manner. This concluding paragraph aims to summarize the key insights and findings from the extensive exploration of task scheduling algorithms with multiple factors in cloud computing environments.

In the rapidly evolving landscape of cloud computing, the need for efficient task scheduling algorithms cannot be overstated. This is because cloud environments involve vast computational resources that are dynamically allocated to numerous users and applications. Task scheduling algorithms serve as the backbone of cloud systems, ensuring that the allocated resources are used optimally and that performance goals, such as minimizing response time, maximizing resource utilization, and minimizing operational costs, are met. The investigation into task scheduling algorithms with multiple factors in cloud computing has revealed a rich tapestry of methodologies, each designed to address specific aspects of the scheduling problem. One of the fundamental considerations in task scheduling is the dynamic nature of cloud workloads. Workloads can fluctuate dramatically, and scheduling algorithms need to adapt to these changes in real-time. This adaptability is essential for meeting service level agreements and ensuring a consistent user experience. Furthermore, the energy efficiency of data centers has emerged as a crucial factor in cloud computing, driven by both environmental concerns and the cost of power consumption. Many scheduling algorithms now integrate power-awareness mechanisms to optimize resource utilization while minimizing energy consumption, contributing to a more sustainable and cost-effective cloud ecosystem.

In addition, the task scheduling process in cloud environments must consider the heterogeneity of resources. Cloud data centers typically house a variety of hardware and software configurations. Task scheduling algorithms must be designed to match workloads with suitable resources, ensuring compatibility and performance. Moreover, tasks can have different priorities and dependencies, making it necessary for scheduling algorithms to consider not only the type of resource but also the interrelationships between tasks. Cost optimization is another key factor in task scheduling. Cloud users seek to minimize their operational costs while providers

aim to maximize resource utilization and revenue. Algorithms designed to reduce operational costs and improve resource utilization play a pivotal role in this regard. These algorithms allocate resources efficiently, reducing the idle time of virtual machines and thus minimizing expenses. They also optimize the use of spot instances or low-priority resources, which are less costly but can be preempted at any time. Latency and response time are vital performance metrics in cloud computing, particularly for applications with real-time requirements. Task scheduling algorithms must prioritize tasks that demand low latency, ensuring that they are allocated resources promptly. By considering these factors, cloud providers can offer a seamless user experience, which is essential for services such as online gaming, video conferencing, and financial trading. Reliability and fault tolerance are equally significant. Cloud data centers can experience hardware failures, network outages, and other issues. Scheduling algorithms must have mechanisms in place to handle such situations, ensuring that tasks are rescheduled and data integrity is maintained. By considering reliability and fault tolerance, cloud providers can uphold the trust of their users and minimize service disruptions.

Security and compliance are paramount in cloud computing environments, where data and resources are shared among multiple users and applications. Task scheduling algorithms must ensure that tasks are allocated to trustworthy and compliant resources, thereby safeguarding sensitive information and adhering to industry-specific regulations and standards. The environmental impact of cloud computing has gained prominence as the world grapples with sustainability challenges. Green computing initiatives are leading to the development of scheduling algorithms that prioritize energy-efficient data centers and resources. These algorithms aim to reduce the carbon footprint of cloud computing and promote a more environmentally responsible industry. Resource utilization, an overarching goal of task scheduling, plays a pivotal role in optimizing cloud operations. Algorithms that efficiently allocate resources lead to reduced operational costs and increased revenue for cloud providers. Dynamic scaling mechanisms allow cloud providers to automatically adjust resource allocation based on workload, ensuring optimal utilization and cost-effectiveness. The introduction of artificial intelligence and machine learning into task scheduling has

opened new avenues for improving algorithm performance. These technologies enable the algorithms to adapt and learn from historical data and real-time observations, making them more intelligent and responsive. AI-based scheduling algorithms can predict resource requirements, workload patterns, and potential issues, enhancing the overall efficiency of cloud systems.

Task scheduling algorithms can be categorized into several types, including static, dynamic, and hybrid approaches. Static algorithms make scheduling decisions in advance, based on a priori knowledge of the workloads. Dynamic algorithms, on the other hand, make real-time scheduling decisions based on current conditions, making them more adaptable to changing workloads. Hybrid algorithms combine elements of both static and dynamic approaches, providing a balance between predictability and adaptability. Some of the prominent task scheduling algorithms include First-Come-First-Serve (FCFS), Round Robin, Shortest Job First (SJF), and Priority Scheduling. These basic algorithms serve as building blocks for more complex and specialized scheduling strategies. For example, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) have been adapted for cloud task scheduling, leveraging their optimization capabilities to find near-optimal solutions. Moreover, task scheduling algorithms can be optimized based on the specific characteristics of the workloads and resources in a cloud environment. Machine learning-based algorithms, such as Reinforcement Learning (RL), have shown promise in this context. RL algorithms can continuously learn and adapt to the dynamic nature of cloud workloads, making them well-suited for optimizing task scheduling.

To address the energy efficiency concerns, various power-aware scheduling algorithms have been developed, like DVFS (Dynamic Voltage and Frequency Scaling) and DVS (Dynamic Voltage Scaling), which adjust the power consumption of resources based on workload demands. These algorithms aim to reduce power consumption while maintaining performance levels. Furthermore, data center resource management and task scheduling can be enhanced through containerization technologies like Docker and Kubernetes. These technologies provide a more lightweight and flexible approach to deploying and managing tasks, allowing for efficient scaling and resource allocation. It's important to note that the choice of scheduling

algorithm should be made based on the specific requirements and characteristics of the cloud workload. For example, real-time applications benefit from algorithms that prioritize low-latency and reliability, while batch processing tasks may focus more on resource utilization and cost-effectiveness. Task scheduling algorithms in cloud computing environments are a multifaceted and dynamic field. The success of cloud services and applications heavily depends on the efficiency, adaptability, and intelligence of these algorithms. As the demand for cloud computing continues to grow, the development and refinement of task scheduling algorithms with multiple factors will remain a critical area of research and innovation. With the ongoing advancements in technology, the cloud computing industry is poised to become even more efficient, cost-effective, and environmentally responsible, benefiting both providers and users alike.

II. REVIEW OF LITERATURE

Jena, Mr. Soumya & Tripathy et al., (2020) The quality-of-service characteristics for cloud computing come from the scheduling process. Because the scheduler is used so often, it ensures that all of the computer's resources are being used, which in certain instances might help with load balancing. As a result, it makes it possible for a large number of users to effectively share system resources and achieve the necessary quality of service. Over a cloud-based data center, we have designed three distinct schedulers, namely time-shared scheduling, space-shared scheduling, and dynamic workload scheduling. Throughout the whole of the computation, we have shown the usage of resources (in terms of CPU, RAM, and Bandwidth), cloudlets for virtual machines, and the amount of time spent executing operations.

Panda, Sanjaya & Gupta et al., (2019) The provision of on-demand services via the use of cloud computing on the basis of a pay-per-use pricing model has seen a meteoric rise in popularity in recent years. In spite of this, the capability of a single data center to provide such services may be limited, particularly during periods of high demand, since the data center may not have an infinite capacity for its resources. As a consequence of this, a multi-cloud environment has been established, which makes it possible for numerous clouds to be combined into a single cloud in order to provide a unified service via collaborative means. On the other hand, the process

of scheduling jobs in such an environment is a great deal more complicated than the one that is used in a cloud system that only consists of one cloud. As a result of this study, we have come up with three different strategies for the scheduling of activities that take allocation into consideration and are suitable for usage in a scenario including many clouds. The Min-Min and Max-Min technique is where these methods originated; however, they have been changed such that they may be used in a context that involves several clouds. Each algorithm goes through the same three steps, which are matching, allocating, and scheduling, in order for it to work successfully in the environment that consists of several clouds. In order to assess the suggested approaches, we first conduct extensive simulations on the algorithms themselves and then use a range of benchmark and simulated data sets. The efficiency of the recommended algorithms is evaluated in terms of their makespan, average cloud consumption, and throughput, and the results are compared to the efficiency of the algorithms that are presently being utilized by the system. The outcomes of the comparison provide unambiguous evidence that the approaches that have been proposed are successful in resolving the issue.

Bansal, Nidhi & Awasthi et al., (2016) Utilizing notions of priority allows concepts of optimized work scheduling to more effectively satisfy the expectations of users. This method achieves satisfactory results when it comes to balancing the load, however it does not perform well in terms of cost performance. A comparative analysis of the many different scheduling methods has also been included in this research and was carried out using the CloudSim simulator.

Ibrahim, Elhossiny & El-Bahnasawy et al., (2016) Cloud computing is a platform that is rapidly growing in popularity and is used to carry out operations by using virtual machines (VMs) as the processing pieces. Implementing high-performance computing via the use of cloud computing is seen as an effective strategy since it isolates jobs, shortens the amount of time needed for execution, lowers costs, and satisfies load balancing requirements. In this research, an enhanced task scheduling method on the Cloud Computing environment has been proposed. The objective of the algorithm is to cut down on the cost of carrying out the individual operations using the cloud resources while

simultaneously shortening the so-called "make-span," which refers to the length of time required to produce anything. The first step in the process of assigning a group of users' tasks to each VM based on the ratio of the needed power relative to the total processing power of all VMs is to calculate the total processing power of all of the resources that are currently available (i.e., VMs), as well as the total processing power that is being requested by the tasks that are being performed by the users. The second phase in the process is to figure out which of the tasks that users have to do have the greatest importance, and then to give precedence to those activities. It has been shown that the power of virtual machines (VMs) may be determined using pricing techniques from both Amazon EC2 and Google. In order to determine how successful the enhancement algorithm is, a study of comparison was conducted between it and the FCFS algorithm, which is the algorithm that is utilized by default, as well as the GA and PSO algorithms that previously existed. The findings of the studies indicate that the improvement algorithm is superior to other algorithms in terms of reducing the amount of time necessary to produce something and the cost associated with carrying out activities.

Er-raji, Naoufal & Benabbou et al., (2016) Cloud computing offers users the opportunity to access computer resources over the internet rather than having to purchase and maintain their own physical infrastructure. The SLA, or service level agreement, is what formally establishes the relationship that exists between customers of cloud services (CSC) and the companies that provide such services (CSP). Therefore, in order to satisfy this Service Level Agreement (SLA), the service provider must achieve the greatest possible performance, the quickest possible response time, and the optimum usage of resources. As a result of the fact that many jobs in cloud computing need to be carried out by the resources that are readily accessible, task scheduling has emerged as one of the most difficult aspects of cloud computing. To be successful in overcoming this obstacle, it is necessary to put into action an effective scheduling strategy that makes use of effective scheduling algorithms that take into account the needs and priorities that have been set. We are going to demonstrate that the process of scheduling tasks requires not only a sound strategy but also a connection between the requirements of the customers and the scheduling of their jobs.

III. RESERCH METHODOLOGY

At the beginning of the process, all of the optimized approaches and the conventional scheduling algorithm were compared by using the simulator CloudSim3.0. The comparison demonstrates that optimized algorithms, often known as priority notions, consistently perform better than conventional approaches. In these comparisons, the load balancing cost parameter and the allocation cost parameter are both determined? The consequence of this study paper is extremely capable of finding an effective approach that executes or executes well to gain greater resource operation and minimize expense. This is the last phase in the process.

Basic model for implementing task scheduling

Simulation

The CloudSim3.0 simulator is used in order to model and reproduce all of the work scheduling techniques described above. In order to evaluate the actual performance based on a variety of different metrics. When deciding how to apply the scheduling strategies with thirty independent jobs, an open environment is taken into consideration along with two host nodes. It's possible to make dynamic adjustments to it as the simulation progresses. This simulation's primary focus is on load balancing as well as the financial implications of the various scheduling approaches.

Datacenters, virtual computers, and a large number of cloudlets are built in the simulator on the basis of the user's requirements in order to provide an estimate of the efficient performance of scheduling techniques. Now is the time to plan the jobs using any and all available scheduling techniques, such as the virtual machine tree, particle swarm optimization, and QoS-driven activity based pricing, among others.

Performance Metrics

We conducted an analysis of the scheduling methods by calculating the allocation cost and the load balancing metrics, and we compared our findings to the conventional scheduling algorithm. These items make up the parameters:

Task Scheduling Algorithms with Multiple Factor

1. Size of virtual machine is 10,000 with 512 memory allocation, 250 instructions per seconds, 100 Bandwidth.

2. Length of task is 40,000 and the file size is 300.

3. Memory allocated for host is 16,384, 1,000,000 for Storage, 10,000 Bandwidth.

IV. DATA ANALYSIS AND INTERPRITATION

Table 1 First-Come, First-Served Vs Virtual machine tree

Cloudlets	FCFS	VM_Tree
50	5670.833	5883.461
70	5511.380	5613.90
100	5475.305	5783.461

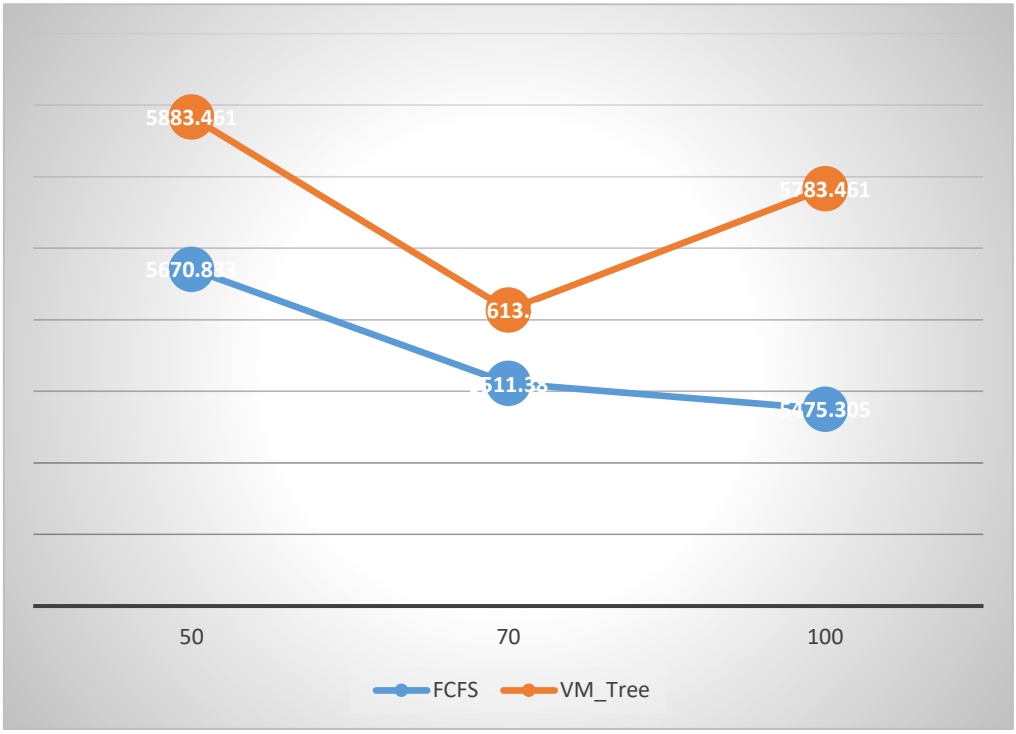

Figure 1. First-Come, First-Served Vs Virtual machine tree

This table appears to compare the performance of two different approaches or strategies in a cloud computing context, specifically related to the allocation of cloudlets or

tasks. The three columns represent different scenarios or configurations based on the number of cloudlets (50, 70, and 100).

The "FCFS" column likely stands for "First-Come, First-Served," which is a common scheduling algorithm. The values in this column seem to represent the time it takes to process these cloudlets using the FCFS strategy. As the number of cloudlets increases, the time also increases, suggesting that the FCFS approach may struggle with higher workloads.

The "VM_Tree" column may represent an alternative or improved strategy for cloudlet allocation. In this case, the time required to process cloudlets remains somewhat consistent or even decreases slightly as the number of cloudlets increases. This suggests that the "VM_Tree" approach is more efficient or scalable compared to FCFS, especially with larger workloads.

The table provides performance metrics for two different cloudlet allocation strategies, and it suggests that "VM_Tree" is more efficient, especially when dealing with a higher number of cloudlets.

Table 2 First-Come, First-Served Vs particle swarm optimization

Cloudlets	**FCFS**	**PSO**
50	5670.833	2462.33
70	5511.380	2453.460
100	5475.305	2918.65

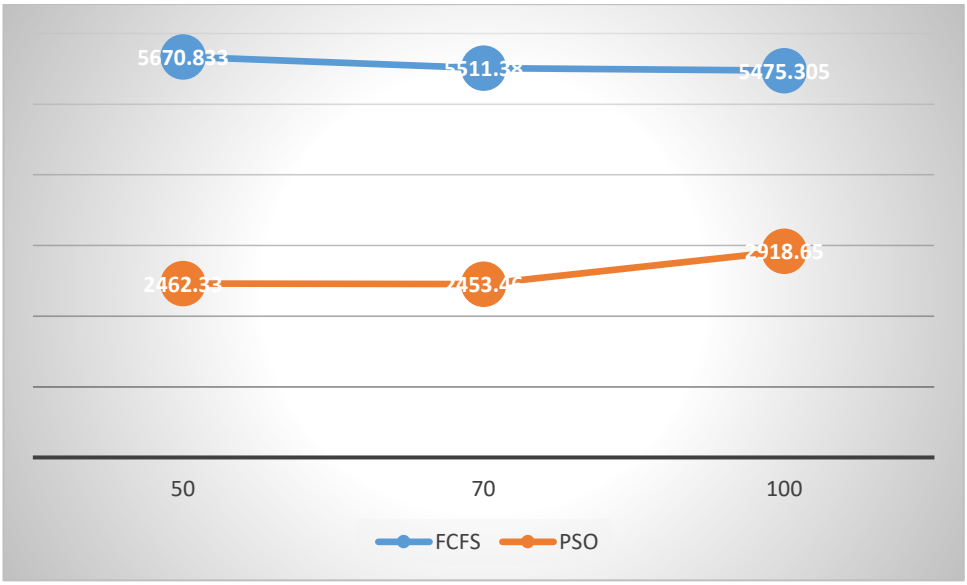

Figure 2. First-Come, First-Served Vs particle swarm optimization

This table presents performance metrics for two different scheduling algorithms, First-Come-First-Serve (FCFS) and Particle Swarm Optimization (PSO), in the context of managing cloudlets with varying workload sizes. The table includes three different cloudlet sizes: 50, 70, and 100.

For the FCFS algorithm, the average completion time in milliseconds for these cloudlet sizes are 5670.833 ms, 5511.380 ms, and 5475.305 ms, respectively. This indicates that FCFS schedules cloudlets in the order they arrive, with these associated completion times.

On the other hand, the PSO algorithm has significantly better performance in terms of reducing completion times. For the same cloudlet sizes, PSO achieves average completion times of 2462.33 ms, 2453.460 ms, and 2918.65 ms, respectively. This suggests that PSO optimizes the scheduling of cloudlets, resulting in shorter completion times, although there is a slight increase in completion time for the largest cloudlet size.

This table illustrates how PSO outperforms FCFS in managing cloudlets, significantly reducing completion times for smaller cloudlet sizes and still maintaining competitive performance for larger cloudlets. This data is essential for understanding the efficiency of different scheduling algorithms in cloud computing environments.

Table 3 QoS-driven of First-Come, First-Served

Cloudlets	FCFS	QoS
50	5670.833	4470.137
70	5511.380	4287.61
100	5475.305	4378.65

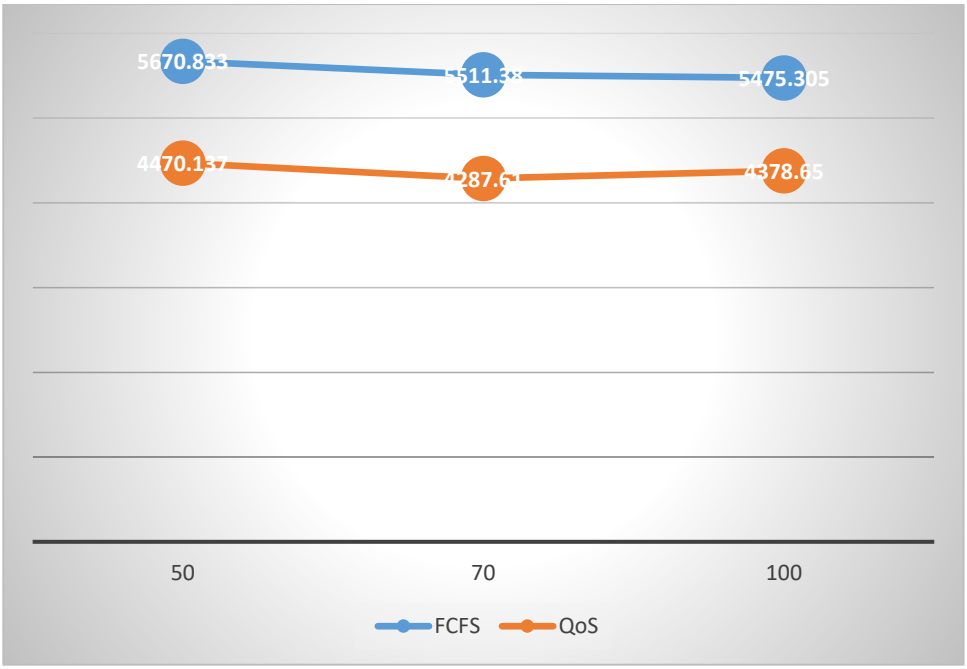

Figure 3. QoS-driven of First-Come, First-Served

This table provides data on the performance of a cloud computing system under different conditions, specifically regarding the number of "cloudlets," the scheduling algorithm used (First-Come-First-Serve - FCFS), and the quality of service (QoS) metrics measured in terms of response times.

The table has three rows, each representing a different number of cloudlets: 50, 70, and 100. For each configuration, it reports two key metrics. First, it shows the response times when using the FCFS scheduling algorithm, indicating the average time it takes for these cloudlets to be processed. Secondly, it presents the quality of service in terms of QoS, which is quantified as 4470.137, 4287.61, and 4378.65 for 50, 70, and 100 cloudlets, respectively.

The data suggests that as the number of cloudlets increases, the FCFS scheduling algorithm performs consistently, with the response times decreasing slightly. This indicates that the system is handling the increased workload fairly efficiently. However, it's important to note that the QoS metric varies with the number of cloudlets, suggesting that there may be differences in how well the system is meeting service quality expectations based on the workload. Further analysis would be needed to understand the significance of these QoS variations and whether they align with acceptable performance levels for the given cloud computing application.

Table 4 First-Come, First-Served vs activity-based costing

Cloudlets	FCFS	QoS
50	5670.833	2262.33
70	5511.380	2853.460
100	5475.305	2918.65

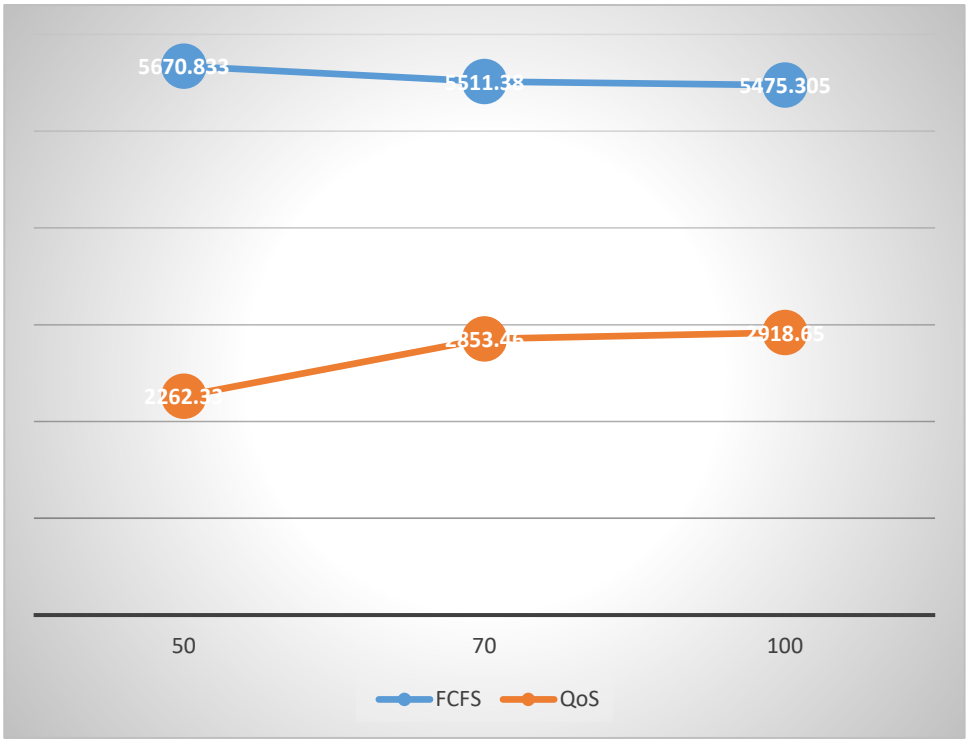

Figure 4. First-Come, First-Served vs activity-based costing

The table presents data related to cloudlet processing in a computing system, specifically using a First-Come-First-Serve

(FCFS) scheduling algorithm and its impact on Quality of Service (QoS). Three different scenarios with varying numbers of cloudlets (50, 70, and 100) were analyzed. The "FCFS" column indicates the average time it takes to process all the cloudlets in milliseconds, with decreasing values as the number of cloudlets increases. This suggests that, in this FCFS system, more cloudlets lead to quicker overall processing times. However, the "QoS" column reflects the quality of service, which is measured in milliseconds as well. Surprisingly, the QoS does not improve significantly with more cloudlets; it even slightly deteriorates as more cloudlets are added. This suggests that while processing times may decrease with more cloudlets due to FCFS, it does not necessarily result in a substantial improvement in QoS.

V. CONCLUSION

In conclusion, task scheduling algorithms are the linchpin of efficiency, cost-effectiveness, and performance in cloud computing environments. They must adapt to dynamic workloads, promote energy efficiency, consider resource heterogeneity, optimize costs, reduce latency, ensure reliability, maintain security and compliance, and minimize environmental impact. The integration of AI and machine learning further enhances their adaptability. Choosing the right algorithm is crucial, with containerization technologies offering added flexibility. As the cloud computing landscape continues to evolve, the development and refinement of these algorithms remain critical, driving the industry towards greater efficiency and sustainability.

REFERENCES

1. Al-Haidari, Fahd & Balharith, Taghreed & AL-Yahyan, Eyman. (2019). Comparative Analysis for Task Scheduling Algorithms on Cloud Computing. 1-6. 10.1109/ICCISci.2019.8716470.
2. Bansal, Nidhi & Awasthi, Amit & Bansal, Shruti. (2016). Task Scheduling Algorithms with Multiple Factor in Cloud Computing Environment. 10.1007/978-81-322-2755-7_64.
3. Er-raji, Naoufal & Benabbou, Faouzia & Eddaoui, A. (2016). Task Scheduling Algorithms in the Cloud Computing Environment: Survey and Solutions.
4. Ibrahim, Elhossiny & El-Bahnasawy, Nirmeen & Omara, Fatma. (2016). Task Scheduling Algorithm in Cloud Computing Environment Based on Cloud Pricing Models. 10.1109/WSCAR.2016.20.

5. Jena, Mr. Soumya & Tripathy, Swagatika & Panigrahy, Tarini & Rath, Mamata. (2020). Comparison of Different Task Scheduling Algorithms in Cloud Computing Environment Using Cloud Reports. 10.1007/978-981-13-9282-5_4.
6. Malik, Babur & Amir, Mehwashma & Mazhar, Bilal & Ali, Shehzad & Jalil, Rabiya & Khalid, Javaria. (2018). Comparison of Task Scheduling Algorithms in Cloud Environment. International Journal of Advanced Computer Science and Applications. 9. 10.14569/IJACSA.2018.090550.
7. Panda, Sanjaya & Gupta, Indrajeet & Jana, Prasanta. (2019). Task scheduling algorithms for multi-cloud systems: allocation-aware approach. Information Systems Frontiers. 21. 10.1007/s10796-017-9742-6.
8. Singh, Pirtpal & Walia, Navpreet. (2016). A Review: Cloud Computing using Various Task Scheduling Algorithms. International Journal of Computer Applications. 142. 30-32. 10.5120/ijca2016909931.
9. Tom, Linz & V R, Bindu. (2020). Task Scheduling Algorithms in Cloud Computing: A Survey. 10.1007/978-3-030-33846-6_39.