Indian Institute of Information Technology, Sricity

Cloud Computing Project Report

An In-Depth Empirical Investigation of State-of-the-Art Scheduling Approaches for Cloud Computing Task Scheduling

Members :

- Aarav Nigam - S20210010001
- Akansh Vaibhav - S20210010010
- Akash Singh Narvariya - S20210010012
- Pranav Singh - S20210010180

Introduction to Task Scheduling Approaches for Cloud Computing

Cloud computing has become an integral part of modern technology, providing scalable and flexible resources for various applications. One of the key challenges in cloud computing is efficient task scheduling, which involves allocating tasks to virtual machines (VMs) in a way that optimizes resource utilization and meets performance requirements.

In recent years, researchers have focused on developing scheduling heuristics that aim to achieve load balancing, scheduling optimization, and energy-aware resource/task scheduling. These heuristics play a crucial role in improving the efficiency and profitability of cloud resources.

The Need for Task Scheduling Heuristics

Task scheduling in cloud computing is the process of efficiently assigning computing tasks and workloads to available resources within a cloud infrastructure. The Task Scheduling Problem is NP Hard in NAture. They help in reducing downtime, improving response time, and ensuring that tasks are executed in a timely manner. Additionally, these heuristics enable load balancing, which distributes tasks efficiently across the cloud, preventing resource overload and underutilization.

State-of-the-Art Scheduling Algorithms

Min-Min Algorithm:
The Min-Min task scheduling algorithm minimizes the makespan by assigning the smallest task to the earliest available machine in a distributed computing environment.

First Come First Serve:
The First Come First Serve (FCFS) task scheduling algorithm allocates tasks to available resources in the order they arrive, without considering their execution time or priority.

Round Robin:
The Round Robin task scheduling algorithm allocates tasks to available resources in a circular order, each receiving a fixed time slice, promoting fairness in execution.

Shortest Job First:
The Shortest Job First (SJF) task scheduling algorithm assigns tasks to available resources based on their execution time, favoring the shortest jobs first.

Enhanced Max-Min Algorithm:
The approach is as follows: initially the task with maximum burst time is executed first then tasks with minimum burst time is selected for execution till the summation of their burst time is less than or equal to the burst time of task that has been executed recently. This procedure continues till every task in the meta- task set is executed completely.

Maximum Completion Time (MCT) Algorithm:
Maximum Completion Time task scheduling algorithm prioritizes tasks based on their maximum expected completion times, aiming to minimize overall job completion time.

Suffarage:
The Suffrage Time task scheduling algorithm assigns tasks to resources by considering the difference between their remaining execution time and the execution time of the second most capable resource, aiming to balance the load.

Task Aware Scheduling Algorithm:
Task Aware Scheduling is an approach that considers the specific characteristics and requirements of each task to optimize their allocation to resources in a computing system for improved performance and efficiency. It alternates between Min-Min and Suffarage Algorithm to achieve this result.

Resource Aware Scheduling Algorithm:
Resource-Aware Scheduling is an approach that considers the availability and capabilities of resources in a computing system to optimize the allocation of tasks, promoting efficient resource utilization. It alternates between Min-Min and Max-Min Algorithm to achieve this result.

VM Migration and its Challenges

Another method for achieving fault tolerance, schedule optimisation, and load balancing in cloud computing is virtual machine migration. It enables the transfer of workloads from occupied virtual machines (VMs) to empty ones, decreasing downtime and increasing resource efficiency. However, there are a number of unique difficulties associated with virtual machine migration, such as overhead and memory usage during the process.

In an effort to address these issues, academics have presented a number of VM migration strategies. In order to get the appropriate level of resource utilisation efficiency, these strategies integrate virtual machine migration with effective scheduling techniques.

In order to assess the efficacy of work scheduling heuristics, multiple assessment criteria are employed. Makespan, throughput, individual level load imbalance, and average resource utilisation rate (ARUR) are some of these indicators. Makespan is a measure of how long a job takes to execute, whereas throughput is the quantity of tasks finished in a specified amount of time. An important statistic for determining the average cloud resource utilisation is ARUR. Improved resource utilisation is indicated by higher ARUR values. Furthermore, SLA breach quantifies how much the scheduling strategy falls short of the service level agreements that were established with the users.

To sum up, work scheduling heuristics are essential for achieving performance requirements and optimising resource utilisation in cloud computing. Modern strategies like Min-Min, Max-Min, TASA, and others have been created to solve the problems of effective job distribution and load balancing. Another strategy that can improve resource utilisation is virtual machine migration, but it has drawbacks of its own. To evaluate the efficacy of scheduling techniques, evaluation metrics including makespan, throughput, ARUR, and Individual Level Load Imbalance are employed. Researchers and cloud service providers can make well-informed judgements to increase the effectiveness and profitability of cloud resources by consistently refining and assessing these scheduling strategies.

Objective of the Work

Th primary motivation behind the work was to meet the demands of both, the cloud consumer and cloud service provider. For users minimum SLA violation was of primary concern and for Cloud Service Providers the focus was to obtain maximum profitability from their services.

The objective of this work is to conduct an in-depth empirical investigation of state-of-the-art scheduling approaches for cloud computing. The study aims to evaluate and compare different static task scheduling algorithms using the HCSP benchmark dataset implemented using CloudSim. The research focuses on analyzing the performance of these scheduling approaches in terms of

Makespan, Average Resource Utilization Rate (ARUR), Throughput, and Individual Level Load Imbalance. The goal is to identify the strengths and weaknesses of each approach and provide recommendations for improving machine-level load balancing, energy efficiency, and resource utilization in cloud environments. The study also aims to highlight the motivations and aspirations of the research, discuss the working of task scheduling algorithms, present experimental results, and provide conclusions and future directions for this study.

Major Contributions :

The document discusses an empirical investigation of various scheduling approaches for cloud computing. It mentions the use of four different dataset instances, each consisting of 1024 tasks (cloudlets) and 32 virtual machines (VMs). The focus is on evaluating heuristic approaches and their impact on resource utilization.

Approaches:
The document presents an empirical evaluation and comparative analysis of state-of-the-art static task scheduling approaches for cloud computing. The study aims to highlight the motivations and objectives of the research in the domain of task scheduling and load balancing in the cloud computing domain.

Investigation of Machine-Level Load Imbalance:
The research focuses on investigating and addressing machine-level load imbalance in order to improve the profitability of cloud resources. The study emphasizes the need to achieve a higher value of Average Resource Utilization Ratio (ARUR) and explores the impact of machine-level load balancing on resource utilization and profitability.

Recommendations for Cloud Service Providers and Users:
Based on the comparative analysis results, the research provides recommendations for both cloud service providers and users. These recommendations focus on machine-level load balancing, energy consumption, and resource utilization. The study aims to guide cloud service providers and users in making informed decisions regarding task scheduling and resource management.

Comparison of Heuristic Approaches :
The document compares several heuristic approaches, including FCFS, SJF, Roud-Robin, Min-Min, Enhanced Max-Min, MCT, Sufferage, RASA, and TASA. It states that MCT severely overloaded the faster machines and mapped zero percent of tasks to the slower VMs.

Based on the empirical investigation, the document suggests that the use of TASA as scheduling approaches can lead to better resource utilization, especially for tasks with smaller sizes. However, it highlights the need for further research and evaluation of different scheduling approaches in cloud computing.
The document concludes with a summary of the research findings and outlines future work in the field of task scheduling and load balancing in cloud computing. The study highlights the significance of machine-level load balancing, energy consumption, and resource utilization in improving the performance and profitability of cloud resources. Future work may involve further optimization and refinement of scheduling algorithms and exploring new approaches to address the challenges in cloud resource management.

Experimental Setup

The experimental evaluation of the scheduling algorithms in this study was conducted using the CloudSim simulation platform. The simulation experiments were executed on a workstation equipped with an Ryzen 5 5600H Hexa-core processor and 8 GBs of main memory.

There are three major ways to perform experiments in the Cloud Computing namely Experimental, Analytical and Stimulation method.
The experimental techniques are expensive and difficult to set up and may require an expert to design the testbeds. Using this may result in high monetary costs.
The analytical techniques are often limited in evaluating the proposed scheduling heuristics.
Simulation approaches are extensively used to evaluate the performance of the underlying scheduling approaches using a wide variety of configurations.

To analyze the performance of the state-of-the-art heuristic algorithms, the researchers chose the Heterogeneous Computing Scheduling Platform (HCSP) dataset proposed by Braunt et al. The dataset is based on the Expected Time to Compute (ETC) matrix and consists of a number of tasks and virtual

machines (VMs). The instance with a size of 1024 x 32 was considered for the evaluation.

The experiments were performed on both hilo and lohi datasets, which have different levels of task and resource heterogeneity.

hilo: Heavy set of tasks with the light capacity of resources
lohi: Light set of tasks and high capacity of resources

The VMs used in the experiments were also inconsistent in terms of heterogeneity. The performance of the scheduling heuristics was evaluated based on parameters such as makespan, throughput, ARUR and load imbalance.

The results of the experiments were obtained and analyzed to compare the performance of the different scheduling approaches. The researchers plotted the results in figures to visualize the performance in terms of average resource utilization, makespan, throughput, and load imbalance.

Overall, the experimental setup involved using the CloudSim simulation platform, selecting the HCSP dataset, and conducting experiments on both hilo and lohi datasets to evaluate the performance of the scheduling heuristics. The results were then analyzed and compared to draw conclusions about the effectiveness of the different approaches.

Performance Measuring Metrics

In the study on state-of-the-art scheduling approaches for cloud computing, several performance metrics were used to evaluate the effectiveness and efficiency of the task scheduling algorithms. These metrics include makespan, average resource utilization (ARUR), throughput, and individual level VM load imbalance.

Makespan:

Makespan refers to the completion time of the virtual machines (VMs) in the cloud. It is measured as the maximum completion time among all the VMs. The scheduling approaches aim to minimize the makespan to improve the overall efficiency of task execution.

Makespan=Max{CTj}=Max{CT1,CT2,...,CTm}

The document provides a comparative analysis of different scheduling approaches based on makespan. The lower the makespan the better the task scheduling Algorithm will be. All the algorithms were tested based on their makespan. The mapping of tasks on available VMs had a clear impact on the overall execution time for task scheduling.

Average Resource Utilization (ARUR):

ARUR is a metric that measures the resource utilization achieved by the scheduling approaches. It is calculated by dividing the average makespan by the makespan. A higher ARUR value indicates better load balancing and resource utilization.

ARUR = avgMakespan / Makespan

The obtained results from the research demonstrate the average resource utilization of various scheduling approaches. The more the Average Resource Utilization will be the better the Algorithm will be. All the algorithms were tested based on their ARUR. Overall, the research highlights the importance of balanced task mapping for efficient resource utilization.

Throughput:

Throughput is a measure of the number of tasks completed per unit of time. It is calculated by dividing the number of tasks by the makespan. Higher throughput indicates better task execution efficiency.

Throughput = numberoftasks / Makespan.

The document provides a comparison of the throughput results for different scheduling heuristics on the available HCSP datasets. Higher throughput values indicate better resource utilization across the available VMs. All the algorithms were tested based on throughput for both hilo and lohi datasets. Overall, the research heighlights the importance of hight throughput for efficient resource utilization.

Individual Level VM Load Imbalance:

VM load imbalance refers to the uneven distribution of tasks among VMs. It can lead to underutilization of resources and decreased performance. The scheduling approaches aim to achieve load balancing by distributing tasks evenly among VMs.

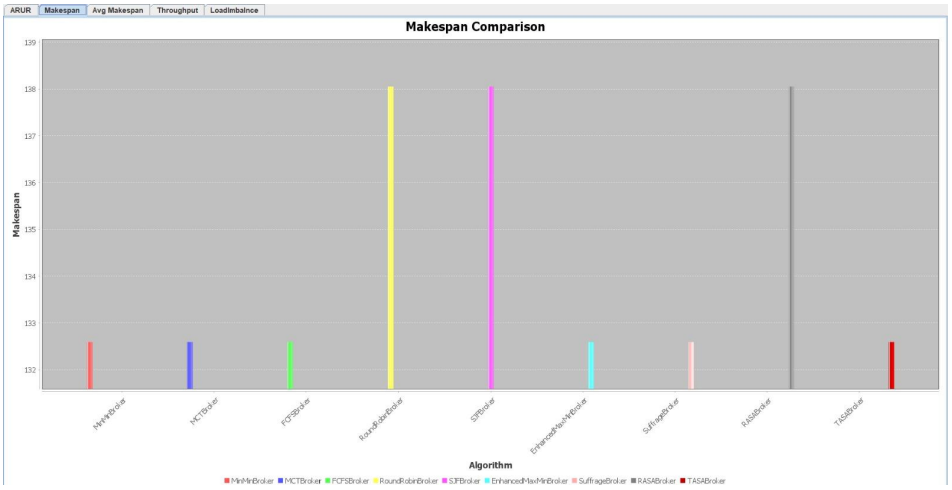
Load Imbalance = Σ(Load_VMi - Average_Load)

The study evaluated the performance of various scheduling approaches for cloud computing based on individual VM level load imbalance. The lower the Load Imbalance the better that algorithm will be. If there is a higher imbalance then that algorithm turns out to be less efficient. Overall, the research shows importance of a a low Load Imbalance for better efficiency.

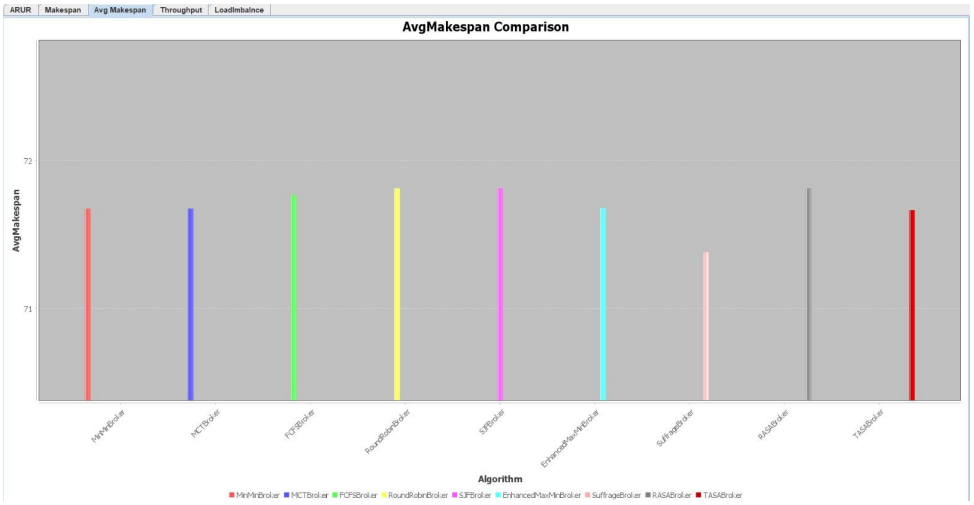
Results/Outcomes of the Study

The study conducted an in-depth empirical investigation of state-of-the-art scheduling approaches for cloud computing. The study compared different scheduling approaches and observed that a higher value of ARUR (Average Resource Utilization Ratio) was achieved. However, there was a need to address machine-level load imbalance and energy consumption for improving the profitability of cloud resources. The study also provided recommendations for cloud service providers and users based on the comparative analysis results. The study highlighted the importance of task scheduling and load balancing in cloud computing and provided insights for future research.

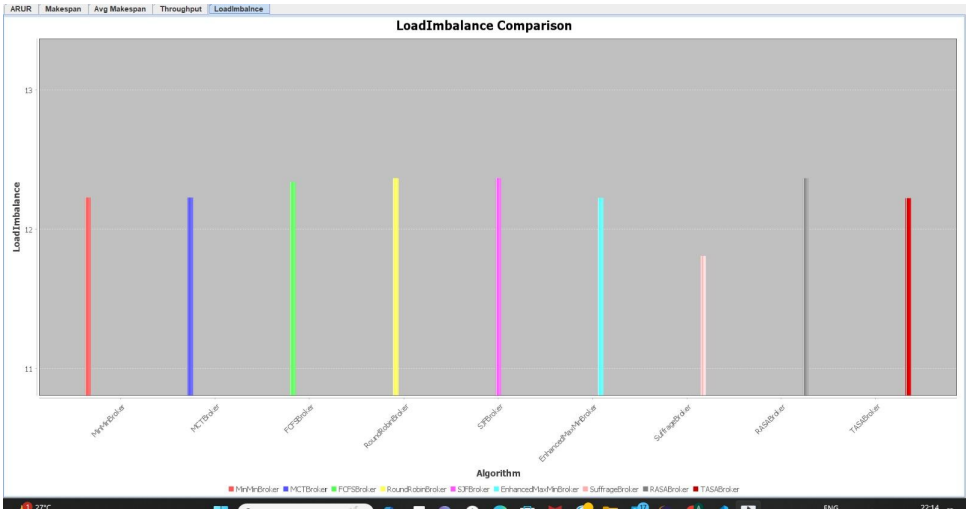
Results Based on Makespan:



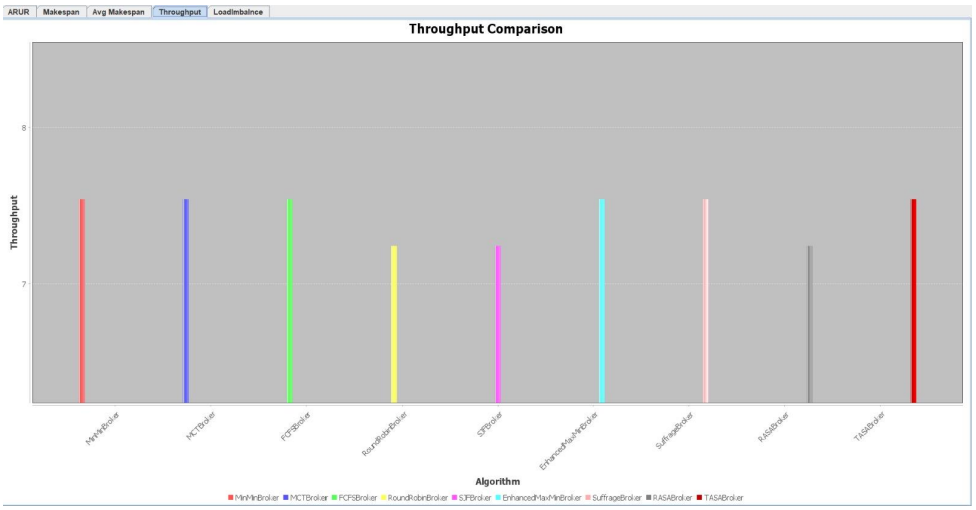
Result Based on Average Makespan:



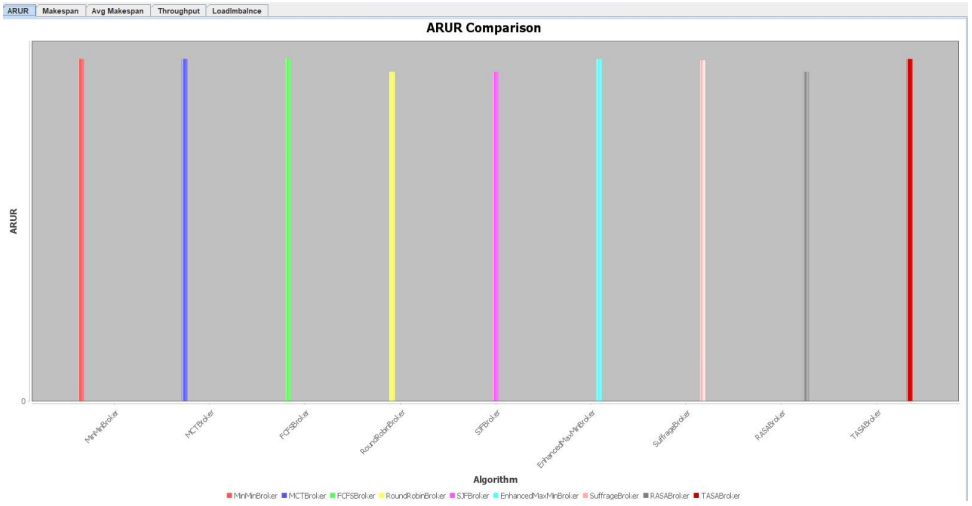
Result BAsed on Load Imbalance:



Result Based on Throughput:



Result Base on ARUR:



Degree of Imbalance:

VmId	Minmin	MCT	FCFS	RoundRobin	SJFS	MaxMin	Sufferage	RASA	TASA
0	9	0.806	0.802	0.941	0.941	0.806	0.822	0.941	0.807
1	0.897	0.897	0.892	0.89	0.89	0.897	0.8	0.89	0.897
2	0.64	0.64	0.636	0.634	0.634	0.64	0.647	0.634	0.633
3	0.706	0.706	0.702	0.7	0.7	0.706	0.619	0.7	0.706
4	0.503	0.503	0.499	0.497	0.497	0.503	0.516	0.497	0.503
5	0.471	0.471	0.467	0.466	0.466	0.471	0.484	0.466	0.471
6	0.643	0.643	0.639	0.637	0.637	0.642	0.657	0.637	0.643
7	0.335	0.335	0.436	0.33	0.33	0.335	0.347	0.33	0.335
8	0.325	0.325	0.322	0.321	0.321	0.325	0.265	0.321	0.326
9	0.152	0.152	0.149	0.148	0.148	0.152	0.131	0.148	0.152
10	0.284	0.284	0.281	0.28	0.28	0.284	0.255	0.28	0.285
11	0.244	0.244	0.241	0.24	0.24	0.244	0.256	0.24	0.245
12	0.05	0.05	0.047	0.046	0.046	0.049	0.059	0.046	0.05
13	0.058	0.058	0.056	0.055	0.055	0.058	0.047	0.055	0.059
14	0.038	0.038	0.041	0.042	0.042	0.038	0.03	0.042	0.038
15	0.061	0.061	0.063	0.064	0.064	0.061	0.052	0.064	0.06
16	0.04	0.04	0.043	0.044	0.044	0.04	0.032	0.044	0.04
17	0.119	0.119	0.121	0.122	0.122	0.119	0.116	0.122	0.123
18	0.199	0.199	0.201	0.202	0.202	0.2	0.192	0.202	0.199
19	0.16	0.16	0.162	0.163	0.163	0.16	0.152	0.163	0.16
20	0.265	0.265	0.267	0.268	0.268	0.265	0.258	0.268	0.265
21	0.293	0.293	0.295	0.296	0.296	0.293	0.287	0.296	0.293
22	0.296	0.296	0.298	0.298	0.298	0.296	0.305	0.298	0.296
23	0.266	0.266	0.268	0.269	0.269	0.266	0.26	0.269	0.266
24	0.367	0.367	0.368	0.369	0.369	0.367	0.361	0.369	0.366
25	0.44	0.44	0.442	0.442	0.442	0.441	0.436	0.442	0.44
26	0.529	0.529	0.53	0.531	0.531	0.529	0.525	0.531	0.529
27	0.576	0.576	0.577	0.577	0.577	0.576	0.586	0.577	0.576
28	0.633	0.633	0.634	0.634	0.634	0.633	0.63	0.634	0.633
29	0.578	0.578	0.579	0.579	0.579	0.578	0.574	0.579	0.577
30	0.634	0.634	0.635	0.635	0.635	0.604	0.642	0.635	0.634
31	0.62	0.62	0.649	0.649	0.649	0.648	0.467	0.649	0.616

Limitations of the Work

Limited Dataset:

The study utilizes a specific benchmark dataset (HCSP instances) implemented using CloudSim. This limited dataset may not fully represent the diverse nature of real-world cloud computing scenarios, potentially limiting the generalizability of the findings.

Single Data Center:

The simulation in the study considers only a single data center comprised of 32 servers. This may not accurately reflect the complexities and scale of

large-scale cloud computing environments with multiple data centers, potentially limiting the applicability of the results.

Lack of Energy Consumption Analysis:

Although the study mentions the need to investigate energy consumption, the specific analysis of energy consumption for the compared approaches is not provided. This omission limits the understanding of the energy efficiency implications of the scheduling algorithms.

No Real-World Deployment:

The evaluation and analysis of the scheduling algorithms are conducted within a simulation environment (CloudSim). While simulations can provide valuable insights, the lack of real-world deployment and testing may limit the practical applicability and real-world performance of the proposed approaches.

Limited Performance Metrics:

The study primarily focuses on execution time (Makespan) and average resource utilization (ARUR) as performance metrics. Other important metrics, such as throughput, SLA violation, and profitability, are mentioned but not extensively analyzed. This limited focus may overlook important aspects of scheduling effectiveness and efficiency.

No Comparison with External Approaches:

The study compares the proposed scheduling approaches with state-of-the-art approaches mentioned in various studies. However, there is no direct comparison with external approaches beyond the ones mentioned in the document, potentially limiting the comprehensive evaluation of the proposed approaches.

No Discussion on Scalability:

The document does not explicitly discuss the scalability of the proposed scheduling approaches. The performance and effectiveness of the algorithms in larger-scale cloud environments with a higher number of tasks and resources are not addressed, which may limit their practical applicability in real-world scenarios.

Future scope

The document briefly mentions that there is a need to investigate and address machine-level load imbalance and energy consumption for improving the profitability of cloud resources. It also suggests recommendations for cloud service providers and users based on the comparative analysis results.

Based on this information, the future scope of the work could involve further research and development in the following areas:

Machine-level load balancing: Investigating and developing more efficient algorithms and approaches to address load imbalance issues in cloud computing environments.

Energy consumption optimization: Exploring methods to reduce energy consumption in cloud data centers, such as optimizing resource allocation and workload distribution.

Performance evaluation: Conducting more empirical evaluations and comparative analyses of scheduling approaches using different datasets and benchmarks to gain further insights into their performance and effectiveness.

SLA violation and profitability analysis: Investigating the impact of scheduling approaches on SLA violation rates and profitability for cloud service providers, and developing strategies to minimize SLA violations and maximize profitability.

Observation of the Study

The study conducted an in-depth empirical investigation of state-of-the-art scheduling approaches for cloud computing. The researchers evaluated and compared the performance of different task scheduling algorithms using the HCSP dataset implemented using CloudSim. They analyzed various parameters such as makespan, throughput, load imbalance, and ARUR. The results showed that the TASA approache performed well in terms of resource utilization, achieving a balanced task distribution on available VMs. Overall, the study provided valuable insights into the performance trade-offs of different

scheduling approaches and made recommendations for cloud service providers and users based on the comparative analysis results.

The results showed that Suffrage and TASA Algorithm performed best in all the parameters, showing that they are the best possible option for doing the Task Scheduling in Cloud Computing Environment for best results.
TASA and Suffarage performed best in terms of Makespan and Average Makespan, with roundrobin and SJF performing the worst.
In terms of Throughput Suffarage and RASA showed the best output.
Suffrage and TASA again showed best results based on Load Imbalance with again RoundRobin & SJF showing the worst results.
Suffrage and TASA showed better ARUR Results however, there was not a significant difference from other algorithms.

While the study provides valuable insights and recommendations, there are still aspects of the original research questions that remain unanswered. For example, the study does not explicitly address the impact of task scheduling on SLA violation or the level of scalability of the scheduling algorithms. Further research may be needed to explore these aspects in more detail.

References

M. Ibrahim et al., "An In-Depth Empirical Investigation of State-of-the-Art Scheduling Approaches for Cloud Computing," in IEEE Access, vol. 8, pp. 128282-128294, 2020, doi: 10.1109/ACCESS.2020.3007201.

Pradhan, Pandaba & Behera, Prafulla & Ray, B. (2020). Enhanced Max-Min Algorithm For Resource Allocation In Cloud Computing. 29. 1619-1628.

T. Aladwani, ‘Types of Task Scheduling Algorithms in Cloud Computing Environment’, Scheduling Problems - New Applications and Trends. IntechOpen, Jul. 08, 2020. doi: 10.5772/intechopen.86873.

oai:ojs.journal.utem.edu.my:article/1243

Alaei, N., Safi-Esfahani, F. RePro-Active: a reactive–proactive scheduling method based on simulation in cloud computing. *J Supercomput* 74, 801–829 (2018). https://doi.org/10.1007/s11227-017-2161-0

taherian dehkordi, Somayeh & Bardsiri, Vahid. (2015). TASA: A New Task Scheduling Algorithm in Cloud Computing. Advances in Computer Engineering and Technology.

https://www.ijcseonline.org/pub_paper/6-IJCSE-04551.pdf