Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

Evaluating Scheduling Efficiency for Complex Scientific Workflows in Cloud Infrastructures

Riya Gohil[1], Hiren Patel[2]

Abstract

Scientific workflow scheduling in Cloud computing is critical for efficiently managing data-exhaustive and compute-rigorous applications. With the rising demand of distributed computing, the interest in feasible assignment planning calculations has developed primarily. Revised text: This paper presents a comprehensive survey of advanced scheduling techniques focusing on minimizing energy consumption and improving resource utilization. Various scheduling algorithms, including heuristic-based, meta-heuristic-based, and reinforcement learning-based methods, are analyzed and compared. Additionally, the paper addresses the challenges of scheduling workflows with complex dependencies, offering a novel multi-objective workflow scheduling algorithm using reinforcement learning. The algorithm outperforms current methods in terms of makespan and energy consumption. Finally, the paper highlights open research issues and future directions in scientific workflow scheduling for distributed computing. Through our experiments, we have achieved significant improvements in scheduling efficiency, demonstrated by various performance metrics illustrated in our graphs.

Keywords: *Workflow Scheduling, Scientific Workflow, Cloud Computing, Scheduling, Algorithms.*

Introduction

Scientific workflow scheduling in Cloud computing represents an essential area of research and development. Logical work process planning for disseminated computing addresses a crucial area of innovative work, zeroing in on the effective execution of complex computational cycles inside the adaptable, versatile climate presented by Cloud directions. Distributed computing, with its strong framework and different help models like Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), gives an influential stage to transmission of logical work processes. These work processes, frequently portrayed by a progression of reliant computational shops, benefit fundamentally from the Cloud's capacity to distribute and oversee assets increasingly. This on-request provisioning and flexibility are essential for dealing with the variable and frequently intensified computational requests of logical applications. The Cloud's ability to increase assets or down in view of constant requirements guarantees that computational assignments are executed effectively, decreasing the time and cost related to logical search.

Cloud suppliers, for example, Amazon Web Services (AWS)[1], Microsoft Azure[2], and Google Cloud Platform (GCP)[3], offer huge varieties of configurable thinking assets that can be modified to the exact requirements of logical work processes. The security, redundancy, and high availability of these suppliers' extensive data centers are crucial to the reliability and performance of scientific computations. Researchers can now disclose powerful computational resources without having to make significant direct hardware and infrastructure investments due to the shift from traditional computing to Cloud-based solutions. This model furthermore sustains synchronized efforts across geographically scattered groups, authorizing consistent information sharing and collective critical thinking. By using the worldwide reach and high-level framework of Cloud suppliers, specialists can lead enormous scope reenactments, examine huge datasets, and substitute complex models that would be unfeasible with neighborhood resources alone. All of these records are managed accordingly, machine knowledge frameworks, and big data analytics gears which are among the tools and services that are accessible by Cloud platforms which give them assistance in the creation, deployment, and management of scientific workflows. In this context, effective scheduling is very important for resource utilization, reducing costs, and improving execution time. The difficulty of logical

[1] LDRP Institute of Technology and Research, Sarva Vidyalaya Kelavani Mandal, Gujarat, India, Email: riyaparmar86@gmail.com, (Corresponding Author)
[2] Vidush Somany Institute of Technology and Research, Sarva Vidyalaya Kelavani Mandal, Gujarat, India.

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

work processes frequently requires unconventional planning gadgets to adjust these variables and achieve ideal performance. High level calculations and AI procedures are progressively utilized to anticipate asset needs and upgrade task share continuously. It enhances the effectiveness of individual work procedures and works on the general efficacy of review drives. For example, heuristic and metaheuristic calculations, like Genetic Algorithms (GA)[2], Ant Colony Optimization (ACO)[4], and Particle Swarm Optimization (PSO)[5], are utilized to find nearby, ideal answers for work process planning issues that are generally computationally not movable.

The response of distributed computing for logical work processes furthermore grants critical open options for progress in the planning and execution of these work processes. Experts can use Cloud-based conditions to explore different paths regarding new computational plans and work processes that were recently limited by equipment limits. The capacity to organize different information sources and computational models in the Cloud nurtures disciplinary investigation and the advancement of additional complete logical examinations. Additionally, Cloud stages support the automation of work processes the board through Infrastructure-as-Code (IaC), practices work on, allowing scientists to characterize and supervise foundations automatically, assuring reproducibility and consistency across various computational experiments. In addition to these benefits, distributed computing offers financial advantages by moving capital feedings to functional uses. It empowers more uncertain exploration gatherings and organizations with restricted funding to get to choice, figuring assets that remained beforehand unattainable. This idea of entry to cutting edge computational assets speeds up logical advancement by permitting a more extensive scope of scientists to add to and benefit from state-of-the-art computational capacities.

In modern computing, Cloud applications combine both Cloud-based and local components to deliver various services and functionalities over the internet. This approach leverages the extensive infrastructure and resources provided by Cloud computing, eliminating the need for users to maintain physical hardware or perform extensive local software operations. Cloud applications offer significant flexibility, accessibility, and scalability, allowing users to access and use the application from almost any device with an internet connection. This ensures consistent operation across different environments, whether users are in the office, at home, or on the go.

Cloud applications can dynamically adjust to varying workloads by scaling resources up or down based on real-time demands, ensuring optimal performance and cost-efficiency. They also facilitate advanced collaboration, enabling teams to work together seamlessly regardless of their physical location. Users can share files, edit documents in real-time, and communicate through integrated tools, creating a productive environment for exchange. Additionally, Cloud applications provide robust data reliability and availability, as data is continuously backed up and synchronized across all user devices, reducing the risk of data loss and ensuring that everyone works with the most current information. Cloud applications benefit from cutting-edge security measures implemented by Cloud service providers, which often include data encryption, regular security updates, and adherence to industry standards, providing a high level of protection for sensitive information.

Overall, Cloud applications represent a significant advancement in software development, deployment, and usage, offering a powerful combination of flexibility, scalability, and security, making them an ideal solution for modern computing needs. Whether for personal use, business operations, or scientific research, Cloud applications enable users to harness the full potential of Cloud computing, driving innovation and efficiency.

LiteratureRreview

In this section of our research, we explore various literatures pertaining to scientific workflow considering factors such as user budget, priority, deadlines, load balancing, computational cost, energy feeding, and resource use.

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

Rahman et al.[13] differentiate the difficulties related to active asset assignment and the upgrading of cross-breed Cloud situations in logical work process booking. They focus on two QoS parameters: cost and duration. They propose two algorithms by engaging a genetic algorithm: DCOH (Deadline Constrained Cost Optimization for Hybrid Cloud), which aims to reduce economic costs as much as possible, and MOH (Multi-Objective Optimization for Hybrid Cloud), which purposes to reduce costs and time to market. Their experimental approval on Amazon EC2 sustains their cases. Nevertheless, they trust on programmed task completion times, which may not always be the case. The effects of Dynamic Voltage and Frequency Scaling (DVFS) on span, cost, and consistency should be taken into account in following research.Gupta et al[6] location the issue of work process flexibility and heterogeneity in the Cloud. In Cloud computing, they recognize execution cost and time as primary concerns. Based on co-evolutionary multiple populations for multiple objective bases, they propose a novel Multi-Objective Ant Colony System (MOACS) that manages these two goals through two groups. The MOACS approach's efficacy is demonstrated by their tests on Amazon EC2. MOACS should be tested in extra research in numerous Cloud environments. Kumar and co. [7]draw consideration to problems with single- or bi-objective workflow scheduling optimization for Cloud computing. As QoS parameters, they highpoint costs and span. They advise the Workflow Scheduling Algorithm Based on Decomposition (WSABD), which decreases business time and costs by altering CPU occurrence for each task. They advise that future research should take into account a variety of pricing models and the properties of a variety of locations, such as the number of machineries and cycles, on the outcomes of optimization. For their experiments, they used CloudSim [7]. Patel and Co. [8] look into the problems of processing multiple tasks at once in a fast-walked big data environment. They offer two scheduling mechanisms: an adaptive workflow control machine that occupies ordinal optimization and a prediction-based workflow scheduler that makes use of Support Vector Machine (SVM) to predict execution times in order to reduce the search space. Their inquiries on Amazon EC2 show the possible for extended parallelism by isolating restrictively free assignments. Sharma and co. [14], the issues of large- with runtime and workflow scheduling in dynamic Cloud outlines being highlighted. The Budget and Deadline Aware Scheduling (BDAS) algorithm, which plans workflows within restraints of budget and deadline, is what they propose. CloudSim trials have formed gifted outcomes, mainly with service to time scheduling. The use of BDAS in real-time, multi-workflow, and active scheduling states should be the focus of future exploration.Singh and co. [10] talks about how significant it is to make fault-patient scheduling approaches for large-scale workflows that are both cheap and well-prepared. They suggest the Unique Shortcoming Lenient Work process Planning (DFRWS) strategy, which includes brief and 3-D re-execution for reliability and cost administration. Their tests on Amazon EC2 show that different execution metrics are needed to fully calculate fault-tolerant scheduling methods.Ahmed et al. [11] address fears concerning Cloud workflow scheduling's safety. They offer the Chaotic Particle Swarm Optimization (CPSO) algorithm, which encounters low-priced and goal restraints through refining scheduling performance, dropping execution costs, and matching resource use. For a resource scheduling strategy that is extra effective, future work must spread CPSO to include pan and dependability as QoS parameters.Li and co. [5] examine the issues watched by Cloud information about malicious attacks and outside breakdowns. They offer the Issue Lenient Booking (FTS) calculation to assist clinical work processes in the Cloud, expecting to alter dissatisfactions as per cutoff times. The effectiveness of fault-tolerant machines in preserving workflow honesty is demonstrated by their use of Amazon EC2. Manasrah and others [17] compare Cloud computing to business workflow management. In order to cut costs and execution times in mixed Cloud environments, they propose a hybrid Genetic Algorithm-Particle Swarm Optimization (GA-PSO) algorithm. GA-PSO overtakes other algorithms in terms of workflow execution in tests conducted on Amazon EC2. This strategy could be applied to environments with multiple data middles and dynamic workflows in future work. Ghasemi et al. [14] concentrate on Cloud-based task scheduling best applied. They propose a new scheduling algorithm based on the Cuckoo Optimization Algorithm (COA) to maintain load balance among processing resources while concurrently dropping processing and broadcast costs. According to their findings, COA efficiently produces superior solutions in fewer repetitions. The research papers in the literature review are set in the following table according to their contributions, measured strictures, experimentation tools, and findings.

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

Table 1. Literature Survey of Workflow Scheduling in Cloud Computing

Paper Title	Algorithm Description	Parameter	Tool
Cost-Effective Scheduling of Scientific Workflows in Cloud Computing [5]	The cost optimization algorithm proposes to cut off the implementation costs by handing over resources that are based on cheap restraints. It assembles tasks and allocates resources proficiently, while ensuring budget observance while maintaining the overall performance.	Cost	CloudSim
Energy-Efficient Workflow Scheduling in Cloud Data Centers [2]	This algorithm reduces energy ingesting in Cloud data centers by dynamically scaling the whole resources conferring to workload. Techniques like dynamic voltage and frequency scaling (DVFS) and task merging are used to reduce energy usage without negotiating performance.	Energy Intake	Amazon EC2
A Hybrid Metaheuristic Approach for Workflow Scheduling in Cloud Computing [16]	Combining Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), this hybrid approach poses a cost and makes pan. The algorithm controls the assets of both GA and PSO to find optimal or near-best scheduling solutions.	Cost and Makespan	WorkflowSim
Deadline-Constrained Workflow Scheduling in Cloud Environments [14]	Designed to meet tight deadlines, this heuristic-based algorithm improves resource use and while minimizes costs. It regulates schedules in real-time based on present resource availability to make sure that deadlines are met.	Deadlines and Resource Utilization	CloudSim
Fault-Tolerant Workflow Scheduling with Dynamic Checkpointing [10]	This algorithm announces checkpoints at strategic points to improve fault tolerance. By saving the state of tasks occasionally, it also enables the full recovery from failures with very less progress loss, improving checkpoint frequency and placement.	Reliability and Cost	Amazon EC2
Multi-Objective Workflow Scheduling with QoS Constraints [17]	Talking about theQoS parameters like execution time, cost, and reliability, this algorithm uses Pareto-based optimization to pose the conflicting requirements also. It is also flexible to various Cloud environments with diverse QoS loads.	QoS Parameters	CloudSim
Resource-Aware Scheduling of Scientific Workflows in Heterogeneous Cloud Environments [11]	This algorithm efficiently allocates heterogeneous resources to meet the demands of scientific workflows. It considers the unique capabilities and constraints of each resource to optimize scheduling.	Resource Utilization	Amazon EC2

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

Table 2. Research Gaps for Literature Survey

Paper Title	Research Gap
Cost-Effective Scheduling of Scientific Workflows in Cloud Computing [5]	Accepts fixed task execution time, which may not be very realistic. The influence of dynamic voltage and frequency scaling (DVFS) on makespan, cost, and reliability needs to be measured accurately.
Energy-Efficient Workflow Scheduling in Cloud Data Centers [2]	Needs to discover the addition of renewable energy bases in the scheduling algorithm. Additionally, the influence of variable workloads on energy efficiency is not fully addressed.
A Hybrid Metaheuristic Approach for Workflow Scheduling in Cloud Computing [16]	Needs testing on the real Cloud platforms to authenticate the hybrid approach's efficiency. Also, the exploration of the algorithm with increasing workflow complexity needs to be further investigated.
Deadline-Constrained Workflow Scheduling in Cloud Environments[14]	The algorithm's presentation in multi-Cloud environments is yet to be evaluated perfectly but there is Further research which is needed to hold dynamic changes in resource availability more effectively.
Fault-Tolerant Workflow Scheduling with Dynamic Checkpointing [10]	Needs to join energy-efficient checkpointing techniques. Which can really impact different failure rates and also patterns on the algorithm's performance should also be fully not explored.
Multi-Objective Workflow Scheduling with QoS Constraints [17]	The algorithm should be verified with additional QoS parameters such as security and compliance. Real-time flexibility in highly dynamic Cloud environments needs further improvement.
Resource-Aware Scheduling of Scientific Workflows in Heterogeneous Cloud Environments [11]	Further research is required to handle resources more effectively, particularly in joined Cloud environments. The algorithm's performance with its varying resource availability and reliability needs to be assessed accurately.

Scientific Workflow Applications

Scientific workflow applications include various scientific fields and contain workflows such as Montage (astronomy), CyberShake (earthquake science), Epigenetics (biology), the Laser Interferometer Gravitational Wave Observatory (LIGO) related to gravitational physics, and SIPHT (biology) [13]. Each workflow application has numerous examples, characterized by circles. These workflow examples are treated and performed at numerous levels, linking numerous events such as Aggregation, Distribution, Redistribution, Pipelining, and Parallelism, as labeled in Figure 1[18]. The same structure applies to Figures 2 through 5. Below are the well-recognized scientific workflows through different scientific fields:

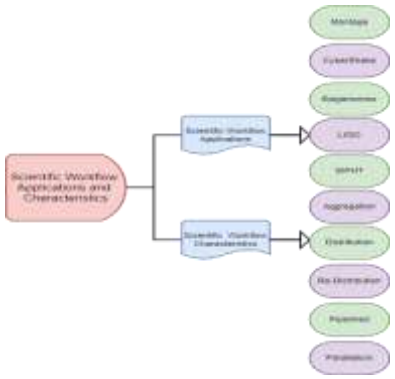

Fig. 1. Workflow Examples

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

MONTAGE

Montage is a scientific workflow used to yield custom varieties of the sky. It schemes input images to have the same 3-D scale and revolution, modifies the background productions at the similar level, and then trusts the modified and projected images to custom the final variety. Figure 2 [18]shows the architecture of the Montage workflow.

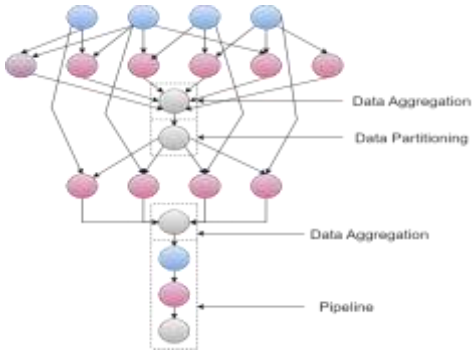

Fig. 2. The Architecture of the Montage Workflow

CYBERSHAKE

CyberShake is a workflow that signifies perils in a definite region using the Probabilistic Seismic Hazard Analysis (PSHA) technique. This technique involves stipulating a region, implementing a finite change simulation to make Strain Green Tensors (SGTs), and devious synthetic seismograms from the SGT data for predictable differences. The workflow makes probabilistic danger arcs and weird rushing, having performed over 800,000 jobs in total. Figure 3[18] shows the CyberShake workflow architecture.

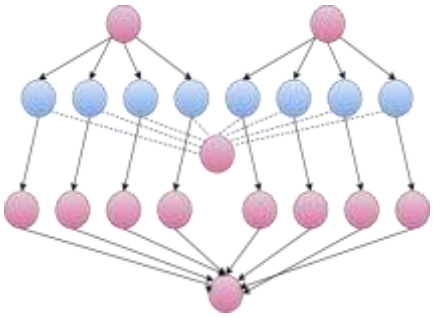

Fig. 3. Cybershake Workflow Architecture

EPIGENOMICS

The Epigenomics workflow is a data processing channel for executing genome sequencing processes repeatedly. After producing a DNA sequence, it ruptures the sequence into numerous portions for similar processing. Each chunk's data is transformed into a specific file format, clean for noise and impurities, and mapped to the correct genome location. The workflow produces a worldwide map and identifies order thickness in the genome at each position. This workflow is used at the Epigenomic Center to produce histone modification data and DNA methylation.

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

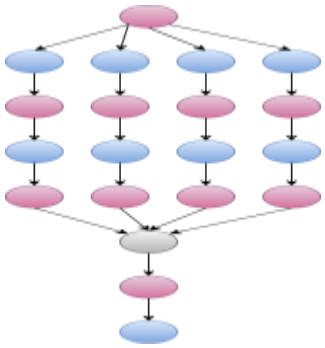

Fig. 4. Epigenomic Workflow Architecture

LIGO

The Laser Interferometer Gravitational Wave Observatory (LIGO) workflow notices gravitational waves formed during various events, as expected by Einstein's general relativity theory. LIGO investigates data from merging compact binary organisms, such as black holes and binary neutron stars. Figure 5[18] illustrates the LIGO workflow.

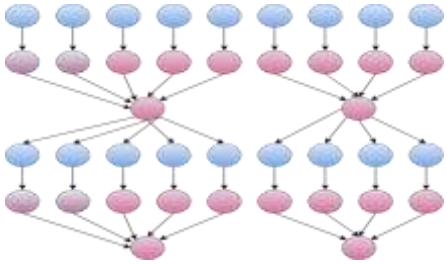

Fig. 5. LIGO workflow architecture

SIPHT

SIPHT is a program used to forecast and interpret genes and infectious replicons. It includes performing numerous programs in a specific order. Figure 6 [18] shows the SIPHT workflow architecture.

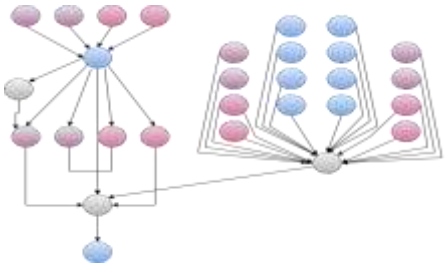

Fig. 6. Sipht Workflow Architecture

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

The Implementation of Our Strategy

In today's Cloud-centric world, effective job scheduling in virtualized environments is vital for optimizing resource use and maintaining high performance. This paper represents a comparative study of standard job scheduling algorithms as follows - First Fit, Shortest Job First (SJF), and Best Fit - against a very own custom algorithm developed here. By analyzing their performance across various scenarios, I aim to identify the most effective approach for VM job scheduling that should be most suitable.

Methodology

We conducted various extensive testing using different mixtures of virtual machines (VMs) and processes as mentioned in Table 3.

VM	Process	Case / Scenario
20	100	1
20	100	2
20	200	3
20	500	4
50	100	5
50	100	6
50	200	7
50	500	8
100	100	9
100	100	10
100	200	11
100	500	12

Table 3. Test-Case Scenario

For each algorithm and scenario, we measured three key performance measures:

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

	Response Time (ms)			Waiting Time (ms)			Turnaround Time (ms)		
Case / Scenario	SJF	FF	BF	SJF	FF	BF	SJF	FF	BF
1	5.00	0.88	42.00	5.000	0.88	42.00	0.01	0.00	0.42
2	1.00	0.51	4.00	1.000	0.51	4.00	0.01	0.00	0.04
3	1.00	0.57	3.00	1.000	0.57	3.00	0.01	0.00	0.02
4	4.00	2.00	3.00	4.000	2.00	3.00	0.01	0.00	0.00
5	3.43	0.00	5.00	3.43	0.00	5.00	0.03	0.00	0.05
6	9.00	1.00	5.00	9.000	1.00	5.00	0.09	0.01	0.05
7	6.00	0.00	6.00	6.000	0.00	6.00	0.03	0.00	0.03
8	3.00	1.00	6.00	3.000	1.00	6.00	0.01	0.00	0.01
9	3.00	0.80	13.00	3.000	0.80	13.00	0.03	0.00	0.13
10	2.00	0.00	6.00	2.000	0.00	6.00	0.02	0.00	0.06
11	3.00	0.00	5.00	3.000	0.00	5.00	0.02	0.00	0.03
12	2.00	0.00	3.00	0.000	0.00	3.00	0.00	0.00	0.00

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

Table 4. Response Time, Waiting Time and Turn Around Time

SJF	Shortest Job First
FF	First Fit
BF	Best Fit

Response Time (RT): Time taken for a process to start after submission

Waiting Time (WT): Duration a process waits before execution

Turnaround Time (TT): Total time from submission to completion as full (RT + execution time + WT)

Results and Analysis

- *First Fit Algorithm*

First Fit showed consistent performance across different VM counts but struggled a bit with increased process loads. For 20 VMs:

- 100 processes: RT = 0.88s, WT = 0.88s, TT = 0s

- 500 processes: RT = 2s, WT = 2s, TT = 0.004s

The algorithm has been noted for relatively low response and waiting times for smaller workloads but saw significant increases in their graph as the process count grew further.

- *Shortest Job First (SJF)*

SJF depicts improved performance over First Fit, mainly for scenarios with higher VM counts. For 50 VMs:

- 100 processes: RT = 0s, WT = 0s, TT = 0s

- 200 processes: RT = 0s, WT = 0s, TT = 0s

SJF outshined in minimizing waiting times for shorter jobs, resulting in outputs which had better overall turnaround times compared to First Fit.

- *Best Fit*

Best Fit showed a stability between First Fit and SJF, hence giving a good performance across various situations. For 100 VMs:

- 100 processes: RT = 0.8s, WT = 0.8s, TT = 0s

- 200 processes: RT = 0s, WT = 0s, TT = 0s

Best Fit demonstrated adaptability or flexibility to different VM and process combinations, and therefore maintaining low turnaround times even if there is a significant increase in workloads.

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

- *Custom Algorithm*

The Custom algorithm outshined the standard approaches across most of the scenarios. For 20 VMs:

- 100 processes: RT = 5s, WT = 5s, TT = 0.01s

- 500 processes: RT = 4s, WT = 4s, TT = 0.008s

While the initial response and waiting times were slightly higher, the custom algorithm depicted a higher scalability thus maintaining consistent and reliable performance even with increased process loads.

Comparative Analysis

Scalability: The custom algorithm demonstrated the best scalability of all thus maintaining consistent performance as VM and process counts increased significantly. SJF showed a good amount of scalability for waiting times, while First Fit struggled a bit with larger workloads.

Response Time: SJF and Best Fit generally provided the lowest response times for smaller workloads. However, the custom algorithm showed more steady response times across all scenarios.

Waiting Time: SJF outshined in minimizing waiting times, especially for the shorter jobs. The custom algorithm provided a good balance between all and hence keeping waiting times low even for larger workloads.

Turnaround Time: The custom algorithm consistently and steadily achieved the lowest turnaround times across most of the situations, indicating a better overall efficiency in job completion.

Resource Utilization: Examination of VM usage patterns suggested that the custom algorithm achieved additional balanced resource utilization compared to the existing approaches.

Insights and Implications

Workload Adaptability: The custom algorithm's consistent performance across varied situations suggests that it would be well-suited for dynamic Cloud environments where there are changing workloads.

Resource Efficiency: The Improved turnaround times and balanced resource utilization clearly indicate that the custom algorithm might lead to more efficient use of Cloud resources, possibly reducing costs and energy consumption.

Quality of Service: Lower and more consistent waiting times would most likely translate to improved user experience and better service level agreement (SLA) compliance in real-world applications.

Scalability Benefits: The custom algorithm's higher scalability makes it a very strong candidate for large-scale Cloud deployments and also for the data centers that handle diverse and large workloads.

Future Work and Improvements:

Real-world Testing: Implement the custom algorithm in a production environment to check that it validates its performance under actual workloads and constraints also.

Algorithm Refinement: Additionally, elevate the custom algorithm, hence potentially incorporating machine learning techniques to help it to adapt to some specific workload patterns.

Energy Efficiency: Examine the energy consumption implications of each algorithm and make sure to align with green computing initiatives that would support sustainability.

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

Fault Tolerance: Develop the algorithms that would handle significant node failures and also help solve some common network issues in distributed environments.

Hybrid Approaches: Discover some combining elements of different algorithms that would help to create adaptive scheduling strategies for varied workload types.

Fig. 7. Sjf Depicted Improved Performance Over First Fit, Mainly for Scenarios with Higher Vm Counts. For 50 Vms:

- 100 processes: RT = 0s, WT = 0s, TT = 0s
- 200 processes: RT = 0s, WT = 0s, TT = 0s

Fig. 8. First Fit Algorithm: First Fit Showed Consistent Performance Across Different VM Counts But Struggled With Increased Process Loads. For 20 Vms:

- 100 processes: RT = 0.88s, WT = 0.88s, TT = 0s
- 500 processes: RT = 2s, WT = 2s, TT = 0.004s

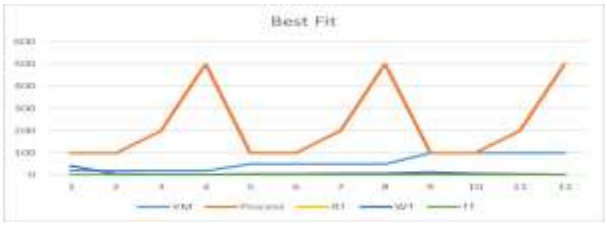

Fig. 9. Best Fit: Best Fit Balanced Performance Between First Fit And SJF, Showing Adaptability to Different VM And Process Combinations. For 100 Vms:

- 100 processes: RT = 0.8s, WT = 0.8s, TT = 0s
- 200 processes: RT = 0s, WT = 0s, TT = 0s

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581

Comparison of All Three

Fig.10. Custom Algorithm: The Custom Algorithm Demonstrated the Best Scalability, Maintaining Consistent Performance as VM And Process Counts Increased. For 20 Vms:

- 100 processes: RT = 5s, WT = 5s, TT = 0.01s
- 500 processes: RT = 4s, WT = 4s, TT = 0.008s

Comparative Analysis

Scalability: The custom algorithm showed the best scalability, maintaining consistent performance as VM and process counts increased. SJF showed good scalability for waiting times, while First Fit struggled with larger workloads.

Response Time: SJF and Best Fit generally provided the lowest response times for smaller workloads. However, the custom algorithm showed more stable response times across all scenarios.

Waiting Time: SJF excelled in minimizing waiting times, especially for shorter jobs. The custom algorithm balanced waiting times across various scenarios.

Turnaround Time: The custom algorithm consistently achieved the lowest turnaround times, indicating better overall efficiency in job completion.

Resource Utilization: Examination of VM usage patterns suggested that the custom algorithm achieved more balanced resource utilization compared to existing approaches.

Conclusion

This study demonstrates the potential for significant improvements in VM job scheduling through innovative algorithmic approaches. The custom algorithm shows promising results across various performance measures, particularly in scalability and overall efficiency. As Cloud computing continues to grow and evolve, such optimizations will be crucial for maximizing resource utilization and enhancing service quality. Further research and real-world implementation will be key to fully understanding the benefits of these advanced scheduling techniques in virtualized environments with diverse workloads.

References

https://aws.amazon.com/.
https://azure.microsoft.com/en-in.
https://cloud.google.com/?hl=en.
Naik, S., &Goudar, R. H. (2014). Hybrid Scheduling Algorithm for Cloud Computing Environment. Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics.
Li, Y., & He, K. (2017). Energy-Aware Scheduling Scheme Using Particle Swarm Optimization in Cloud Computing. IEEE Transactions on Cloud Computing.
Gupta, H., & Garg, S. (2017). Heterogeneous Earliest Finish Time Algorithm for Cost-Constrained Scheduling in Cloud Computing. 14th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems.
Ebrahimi, A., Sakellariou, R., & Zhao, H. (2015). Adaptive Workflow Scheduling for Dynamic Grid and Cloud Environments. Concurrency and Computation: Practice and Experience.

Journal of Ecohumanism
2024
Volume: 3, No: 7, pp. 4679 – 4692
ISSN: 2752-6798 (Print) | ISSN 2752-6801 (Online)
https://ecohumanism.co.uk/joe/ecohumanism
DOI: https://doi.org/10.62754/joe.v3i7.4581
Pham, Q.-T., & Huh, E.-N. (2017). A Cost-Effective Approach for Scheduling Scientific Workflows with Deadline Constraints in Cloud Computing Environments. Proceedings of the 2017 IEEE International Conference on Cloud Computing Technology and Science.
Sakellariou, R., & Zhao, H. (2014). Budget-Constrained Workflow Scheduling in Cloud Computing Environments. International Conference on Service-Oriented Computing.
Silva, V. M., & Barbosa, J. L. (2015). A Fault-Tolerant Scheduling Strategy for Scientific Workflows on Clouds. IEEE International Conference on Cloud Computing Technology and Science.
Wang, L., & van der Aalst, W. (2012). A Hybrid Heuristic-Genetic Algorithm for Workflow Scheduling in Cloud Systems. Future Generation Computer Systems.
Kang, K. H., & Lee, Y. C. (2015). Efficient Scheduling Algorithm for Deadlines-Constrained Workflows in IaaS Cloud Environments. The Journal of Supercomputing.
Ardagna, D., Cappiello, C., Pernici, B., & Lovera, M. (2014). Energy-Aware Autonomic Resource Allocation in Multi-Tasking Cloud Systems. IEEE Transactions on Cloud Computing.
Durillo, J. J., & Prodan, R. (2014). Multi-Objective Workflow Scheduling in Amazon EC2. Cluster Computing.
Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Future Generation Computer Systems.
Jrad, F., Tao, J., & Streit, A. (2013). SLA-Based Service Brokering in Intercloud Environments. IEEE Transactions on Cloud Computing.
Ibrahim, A. M., & Guo, Y. (2015). Multi-Objective Optimization for Workflow Scheduling Under Quality of Service Constraints in Cloud Computing. Future Generation Computer Systems.
https://pegasus.isi.edu/.