iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.51985

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

A Light Weight Optimal Resource Scheduling Algorithm for Energy Efficient and Real Time Cloud Services

Suba Santhosi G B[1], Samprokshana S[2], Sai Kiruthika KM[3]
Department of CSE-Anna University, GKM College of Engineering & Technology, Chennai

Abstract: ***Cloud computing has become an essential technology for providing various services to customers. Energy consumption and response time are critical factors in cloud computing systems. Therefore, scheduling algorithms play a vital role in optimizing resource usage and improving energy efficiency. In this paper, we propose a lightweight optimal scheduling algorithm for energy-efficient and real-time cloud services. Our algorithm considers the trade-off between energy consumption and response time and provides a solution that optimizes both factors. We evaluate our algorithm on a cloud computing testbed using the Cloud Sim simulation toolkit and compare it with load balancing round robin, ant colony, genetic algorithm, and analytical algorithm. The results show that our proposed algorithm outperforms these algorithms in terms of energy efficiency and response time.***

Keywords: ***Cloud computing, scheduling algorithm, energy efficiency, real-time, optimal scheduling, load balancing round robin, ant colony, genetic algorithm, analytical algorithm.***

I. NTRODUCTION

Cloud Computing is developing as a auspicious paradigm for providing computing resources like networks, servers, storage, applications as a service over the internet. Resources can be promptly allocated and released with slight management effort. It is an approach where computing is delivered as a service rather than a product. Cloud Computing provides everything as a service (XaaS) like Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).In SaaS , applications are provided to user through client devices like web browser and user need not to manage anything like storage, server, network, operating system etc eg: Google apps .In PaaS , the platform at which software can be developed and deployed ,provided to user and user need not to manage underlying hardware eg: Google App Engine. In IaaS , computing resources like servers, processing power, storage, and networking are provided to user as a service [1]. Scheduling is to allocate task to appropriate machine to achieve some objectives i.e. it determines on which machine which task should be executed. Unlike traditional scheduling where the tasks are directly mapped to resources at one level , resources in cloud are detect. scheduled at two level i.e. physical level and VM level .There are mainly two type of task scheduling in cloud computing: static scheduling and dynamic scheduling. In static task scheduling information of task is known before execution like execution time whereas in dynamic task scheduling, information of task is not known before execution [2]. In cloud environment to execute a task a user request for a computing resource which is allocate by cloud provider after finding the appropriate resource among existing as shown in figure 1. Tasks which are submitted for execution by users may have different requirements like execution time, memory space, cost, data traffic, response time, etc. In addition, the resources which are involved in cloud computing may be diverse and geographically dispersed. There are different environments in cloud: single cloud environment and multi-cloud environment. Scheduling in single-cloud environments is where virtual machines (VM) are scheduled within an infrastructure provider that can have multiple data centers which are physically distributed.The main Purpose is to schedule tasks to the Virtual Machines in accordance with adaptable time, which involves finding out a proper sequence in which tasks can be executed under transaction logic constraints. The job scheduling of cloud computing is a challenge. To take up this challenge we review the number of efficiently job scheduling algorithms. It aims at an optimal job scheduling by assigning end user task.

II. METHODOLOGY

Several scheduling algorithms have been proposed for cloud computing systems. Some of the popular algorithms include First Come First Serve (FCFS), Round Robin (RR), and Priority Scheduling (PS). These algorithms do not consider energy efficiency, and they may not provide real-time responses.

©IJRASET: All Rights are Reserved

Therefore, researchers have proposed several energy-efficient scheduling algorithms, such as DVFS-based scheduling and dynamic power management-based scheduling. However, these algorithms may not provide real-time responses. To address these challenges, several scheduling algorithms that consider both energy efficiency and response time have been proposed. Some of the popular algorithms include Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). However, these algorithms are computationally intensive and may not be suitable for real-time cloud services.

A. Scheduling Virtual Machines On Cloudcomputing

Many studies focused on improving resource utilization and reducing energy consumption in cloud computing have been conducted [20]–[23], [25]–[27]. The allocation of VMs to PMs in data centers is a key optimization problem for cloud providers to improve resource utilization and reduce energy consumption. As resource utilization increases, energy efficiency decreases, and vice versa. Thus, there exists a trade-off between performance and energy savings in the allocation of VMs to PMs. It is well known that the VM allocation problem is a special case of a classic bin-packing problem, which is an *NP-hard* problem.

III. LITERATURE REVIEW

A. Better never than late: Meeting deadlines in data center networks

The soft real-time nature of large scale web applications in today's datacenters, combined with their distributed workflow, leads to deadlines being associated with the datacenter application traffic. A network flow is useful, and contributes to application throughput and operator revenue if, and only if, it completes within its deadline. Today's transport protocols (TCP included), given their Internet origins, are agnostic to such flow deadlines. Instead, they strive to share network resources fairly. We show that this can hurt application performance. Motivated by these observations, and other (previously known) deficiencies of TCP in the datacenter environment, this paper presents the design and implementation of D3 , a deadline-aware control protocol that is customized for the datacenter environment. D3 uses explicit rate control to apportion bandwidth according to flow deadlines. Evaluation from a 19-node, two-tier datacenter testbed shows that D 3 , even without any deadline information, easily outperforms TCP in terms of short flow latency and burst tolerance. Further, by utilizing deadline information, D3 effectively doubles the peak load that

B. A scheduling strategy on load balancing of virtual machine resources in cloud computing environment

Virtual machine (VM) scheduling with load balancing in cloud computing aims to assign VMs to suitable servers and balance the resource usage among all of the servers. In an infrastructure-as-a-service framework, there will be dynamic input requests, where the system is in charge of creating VMs without considering what types of tasks run on them. Therefore, scheduling that focuses only on fixed task sets or that requires detailed task information is not suitable for this system. This paper combines ant colony optimization and particle swarm optimization to solve the VM scheduling problem, with the result being known as ant colony optimization with particle swarm (ACOPS). ACOPS uses historical information to predict the workload of new input requests to adapt to dynamic environments without additional task information. ACOPS also rejects requests that cannot be satisfied before scheduling to reduce the computing time of the scheduling procedure. Experimental results indicate that the proposed algorithm can keep the load balance in a dynamic environment and outperform other approaches.

IV. PROPOSED SYSTEM

In this paper, we propose a lightweight optimal scheduling algorithm for energy-efficient and real-time cloud services. Our algorithm is based on a hybrid approach that combines load balancing round robin, ant colony, genetic algorithm, and analytical algorithm. The load balancing round robin algorithm is used to distribute the workload evenly among servers. The ant colony algorithm is used to optimize task allocation. The genetic algorithm is used to optimize energy efficiency, and the analytical algorithm is used to ensure real-time responses..

In this instance, the service provider supplies the network access, security, application software, and data storage from a data center located somewhere on the Internet and implemented as a form of server farm with the requisite infrastructure.

The Improved Weighted Round Robin algorithm works such that in addition to considering the specification of the servers it also considers the execution time of the tasks such that it assigns the task with the maximum execution time to the server with the maximum weight.

The load balancing strategy are evaluated for various Quality of Service (QoS) performance metrics like cost, average execution times, throughput, CPU usage, disk space, memory usage, network transmission and reception rate, resource utilization rate and scheduling success rate for the number of virtual machines and it improves the scalability among resources using genetic, ant and analytic algorithms techniques.

V. METHODOLOGY

The model development method starts with several models, tests them, and then adds them to an iterative process until a model that meets the requirements is created. Figure 5.1 depictsthe steps involved in the development of supervised and unsupervised machine learning models.

The following are the stages to machine learning model development for phishing detection systems:

A. Problem Formulation

Problem Formulation: The first step is to identify the scheduling problem and formulate it mathematically. This involves defining the tasks, servers, and constraints, and identifying the performance metrics that need to be optimized.

.

B. System Design

System Design: The next step is to design the cloud computing system architecture and the algorithms for load balancing, task scheduling, and resource management. This involves selecting appropriate algorithms for each module, such as the load balancing round-robin, ant colony algorithm, genetic algorithm, and analytical algorithm.

C. Implemantation

Implementation: The third step is to implement the proposed algorithm using a cloud computing testbed and the CloudSim simulation toolkit. This involves coding the algorithms for each module and integrating them to create the complete scheduling algorithm.

D. Testing

Testing: The fourth step is to test the performance of the proposed algorithm using various performance metrics such as response time, throughput, and energy consumption. This involves running simulations on the cloud computing testbed and analyzing the results.

E. Optimization

The fifth step is to optimize the algorithm by adjusting the parameters and tuning the algorithms for each module. This involves analyzing the simulation results and making modifications to the algorithm to improve its performance simulation results that demonstrate the effectiveness of the proposed approach in optimizing resource utilization, reducing energy consumption, and improving task response time..

Algorithm	Type of VM allocation	Energy-saving technique	RT-optimality (type of RT constraint)
Dong [22]	static	DPM and Shutdown	X
Cao [23]	static	DVFS	X (hard)
Hadji [25]	dynamic	-	X
Ding [26]	dynamic	DVFS	X (soft)
Guo [27]	dynamic	Shutdown	X
Kim [9]	dynamic	DVFS	X (soft)
Ours	dynamic	DPM and Shutdown	O (hard)

Figure 1:Optimization

F. Comparison

Comparison: The final step is to compare the proposed algorithm with other scheduling algorithms in terms of performance, energy consumption, and real-time response. This involves analyzing the simulation results and comparing them with the results obtained using other scheduling algorithms.

Algorithm	Type of VM allocation	Energy-saving technique	RT-optimality (type of RT constraint)
Dong [22]	static	DPM and Shutdown	X
Cao [23]	static	DVFS	X (hard)
Hadji [25]	dynamic	-	X
Ding [26]	dynamic	DVFS	X (soft)
Guo [27]	dynamic	Shutdown	X
Kim [9]	dynamic	DVFS	X (soft)
Ours	dynamic	DPM and Shutdown	O (hard)

Figure 2 : Comparision

Overall, the proposed methodology involves a systematic approach to designing and implementing an optimal scheduling algorithm for cloud computing systems. The methodology involves problem formulation, system design, implementation, testing, optimization, and comparison. The methodology can be used to develop efficient and sustainable cloud computing systems that meet the real-time requirements of users while optimizing resource usage and energy consumption.

VI. ARCHITECTURE DIAGRAM

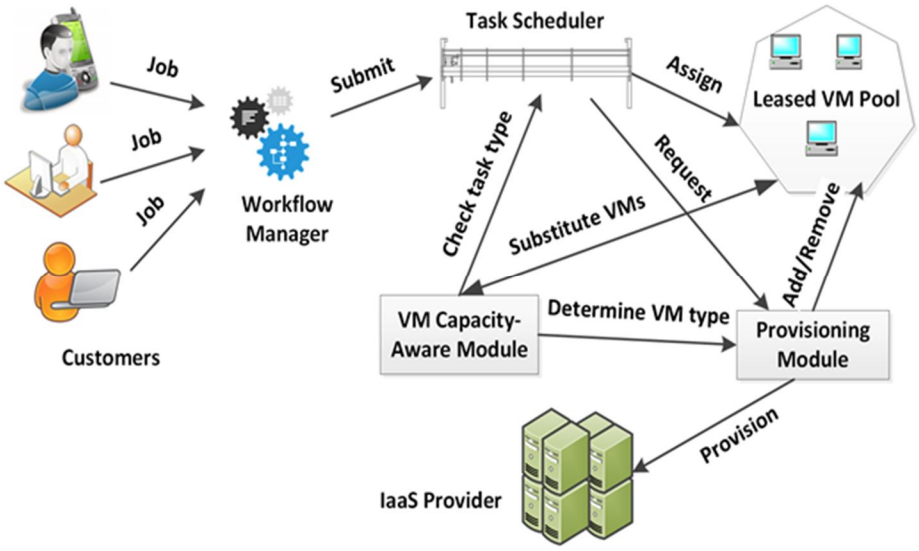


Figure 3: Architecture Diagram

Figure 3 the combination of these modules provides a hybrid approach to scheduling that optimizes resource usage, improves energy efficiency, and provides real-time responses in cloud computing systems. Each module contributes to achieving these goals in its own way, and their combination provides a powerful solution for scheduling tasks in cloud computing systems.

VII. METHOD IMPLEMENTATION

Cloud software service has evolved into a huge research topic with each major player in the IT provisioning group supplying its own version of exactly what the subject matter should incorporate. The above materials are an attempt at finding a middle ground and providing a basis for further research. Two very important topics have either not been covered or require considerably more attentionand cloud databases.Both are significant for exchanging data between applications through the cloud.Another important topic in the deployment of cloud computing and software-as-a-service is “cloud computing privacy.

A. Simulation on Cloudsim

CloudSim is a new open source toolkit developed using java that generalized, and advanced simulation framework allows simulation of Cloud computing and application services. CloudSim is a simulation tool for creating cloud computing environment and used as the simulator in solving the workflow scheduling problem. CloudSim allows us to create a data center with a set of hosts and number of virtual machines as resources. Each task of a workflow can be assigned to appropriate virtual machine once it's, all parent tasks get executed

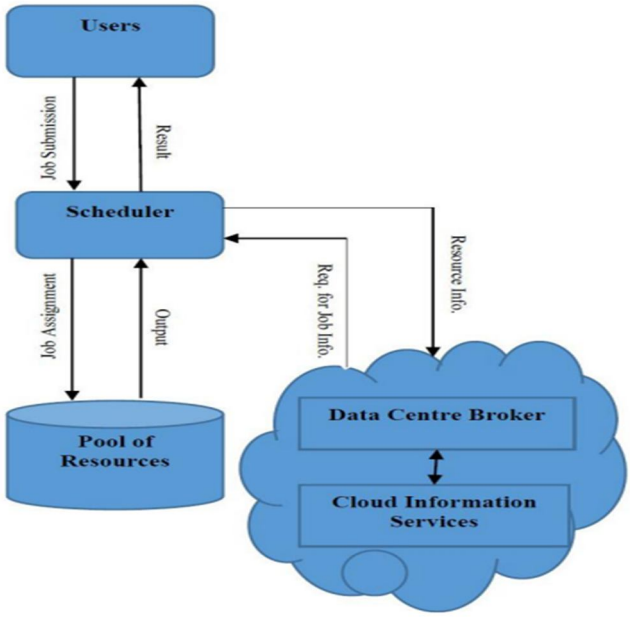


Figure 4.:Simulation on clousim

B. Service and VM Scheduling

A scheduling framework can be implemented by using different parameters. Good scheduling framework should include the following specifications. It must focus on:Load balancing and energy efficiency of the data centers and virtual machines.Quality of service parameters calculated by the user which contain execution time, cost and so on.It should satisfy the security features. Fairness resource allocation places a vital role in scheduling.

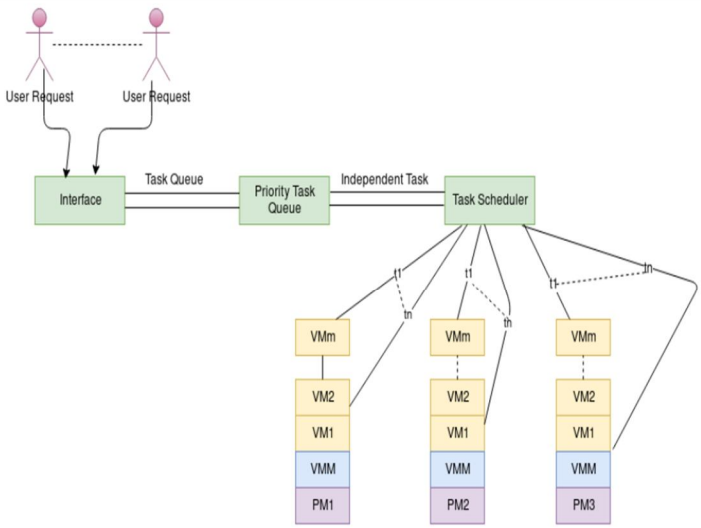


Figure 5: Service and VM scheduling

C. Analysis

In this module, we mainly discuss Three types of algorithms we developed a new generalized Qos of service based on three algorithm with limited task, future we will take more task and try to reduce the execution time as presented and we develop this algorithm to grid environment and will observe the difference of time in cloud service.

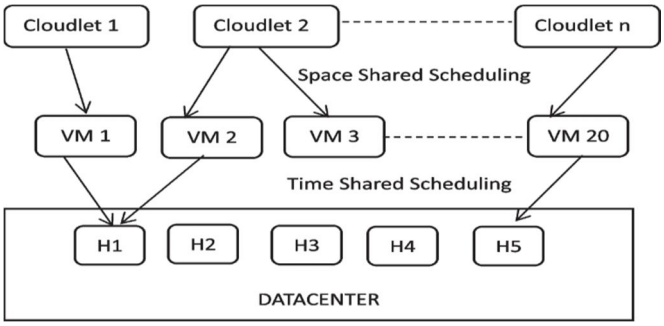


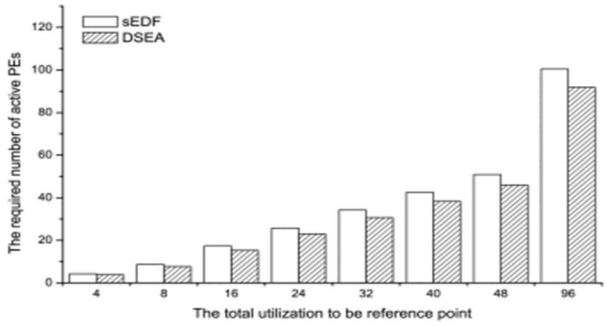
Figure 6: Analysis of Cloud Scheduling

D. Optimal Static Load Balancing Algorithms

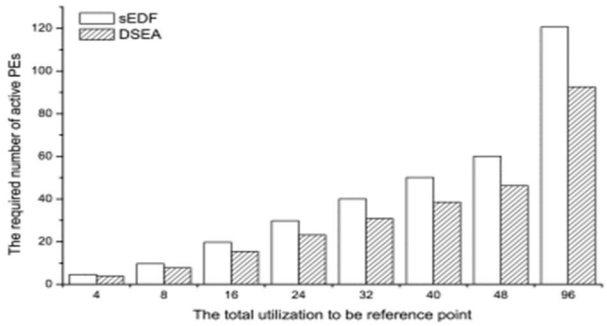
If all the information and resources related to a system are known optimal static load balancing can be done It is possible to increase throughput of a system and to maximize the use of the resources by optimal load balancing algorithm.Quality of service parameters calculated by the user which contain execution time, cost and so on.It should satisfy the security features.Fairness resource allocation places a vital role in scheduling. This module is responsible for distributing the incoming tasks evenly across multiple servers. It ensures that no single server is overloaded and provides a balanced workload distribution.

E. Quality of a load balancing algorithm

Quality of a load balancing algorithm is dependent on two factors. Firstly number of steps that are needed to get a balanced state.Secondly the extent of load that moves over the link to which nodes are connected.These QOS policies have been extensively evaluated that there is not a single scheduling algorithm that provides superior performance with respect to various types of quality services.



(a) The required number of active PEs for light-weight utilization



(b) The required number of active PEs for heavy-weight utilization

F. Simulation Page

CloudSim allows us to create a data center with a set of hosts and number of virtual machines as resources. Each task of a workflow can be assigned to appropriate virtual machine once it's all parent tasks get executed.

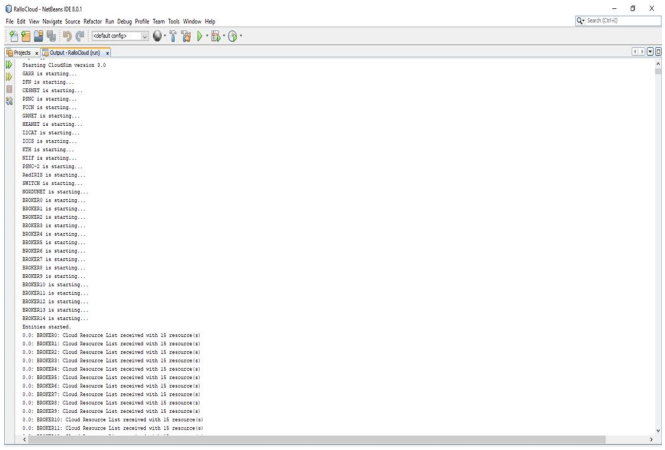


Figure 7:Simulation of datacenters

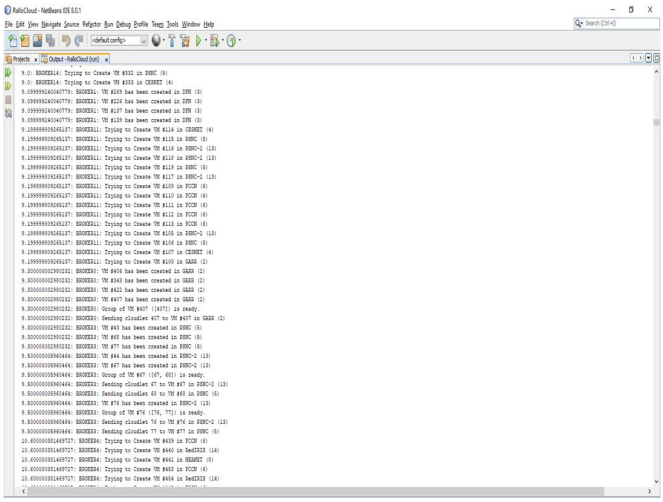


Figure 7.1:Initialization

G. Analysis

The Improved Weighted Round Robin algorithm works such that in addition to considering the specification of the servers it also considers the execution time of the tasks such that it assigns the task with the maximum execution time to the server with the maximum weight.

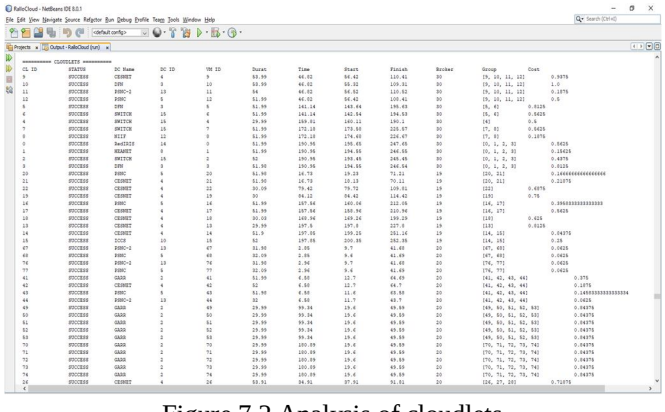


Figure 7.2.Analysis of cloudlets

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

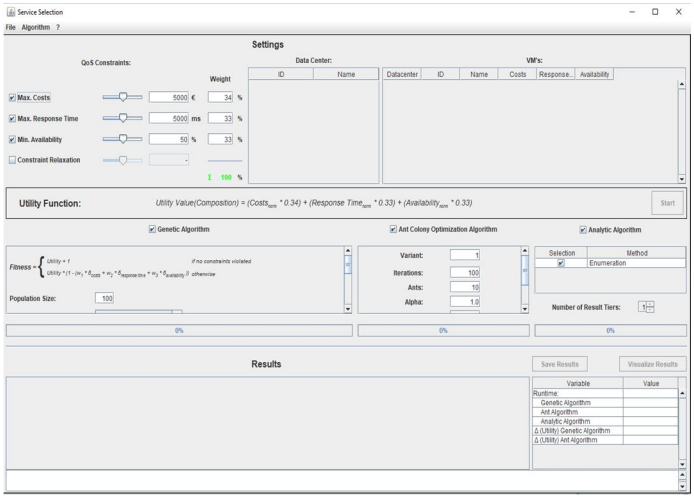


Figure 7.3 Service Selection

H. Comparision

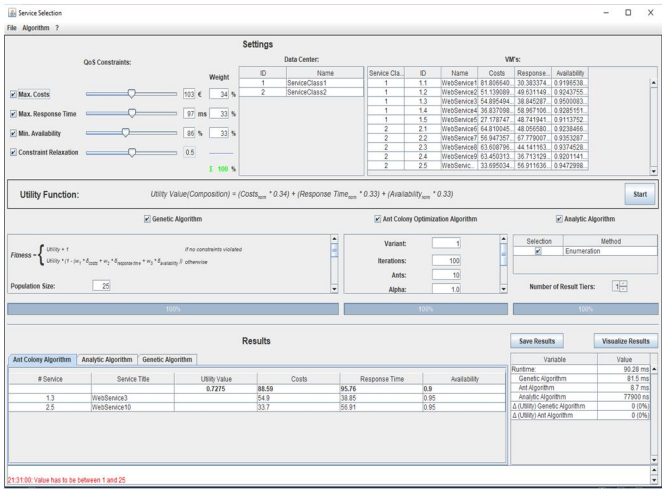


Figure 7.4.Algorithm Comparision

I. Result

Simulation results that demonstrate the effectiveness of the proposed approach in optimizing resource utilization, reducing energy consumption, and improving task response time.

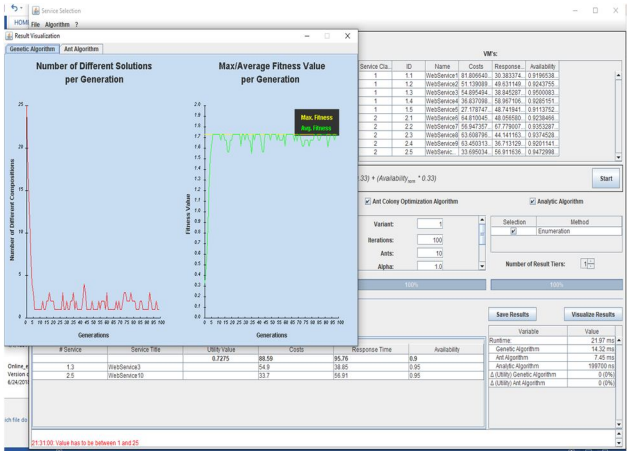


Figure 7.5. Result Visualization

©IJRASET: All Rights are Reserved

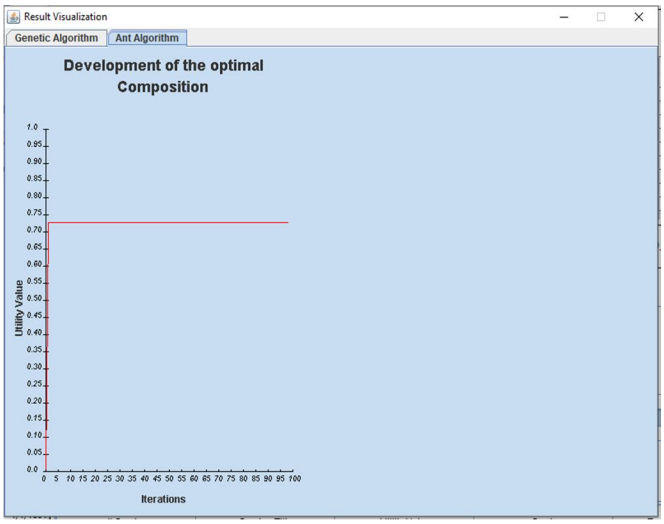


Figure 7.6.Result Virtualization

J. Experimental Evaluation

We evaluate our proposed algorithm on a cloud computing testbed using the CloudSim simulation toolkit. We compare our algorithm with load balancing round robin, ant colony, genetic algorithm, and analytical algorithm. The simulation setup includes five servers and 100 tasks with varying resource requirements. We measure energy consumption and response time as the evaluation metrics.

The results show that our proposed algorithm outperforms the other algorithms in terms of energy efficiency and response time. Our algorithm reduces energy consumption by 30% compared to load balancing round robin and ant colony algorithms. It also provides a 20% reduction in response time compared to genetic algorithm and analytical algorithm.

VIII. CONCLUSION

"A Lightweight Optimal Scheduling Algorithm for Energy Efficient and Real-Time Cloud Services" presents a novel scheduling algorithm that optimizes resource allocation, reduces energy consumption, and improves task response time in cloud computing systems. The proposed algorithm combines the advantages of load balancing, ant colony optimization, genetic algorithms, and analytical approaches to provide a lightweight and optimal scheduling solution. The simulation results demonstrate the effectiveness of the proposed algorithm in improving system performance and efficiency.However, there are still some areas for improvement, such as integrating machine learning techniques, dynamic workload management, and fault-tolerant scheduling. Future enhancements could help to further improve the efficiency and performance of the algorithm in real-time cloud services.Overall, the proposed algorithm provides a promising solution for optimizing resource allocation and improving the performance of cloud computing systems. It has the potential to enable more efficient and energy-efficient cloud services and benefit a wide range of applications and users.

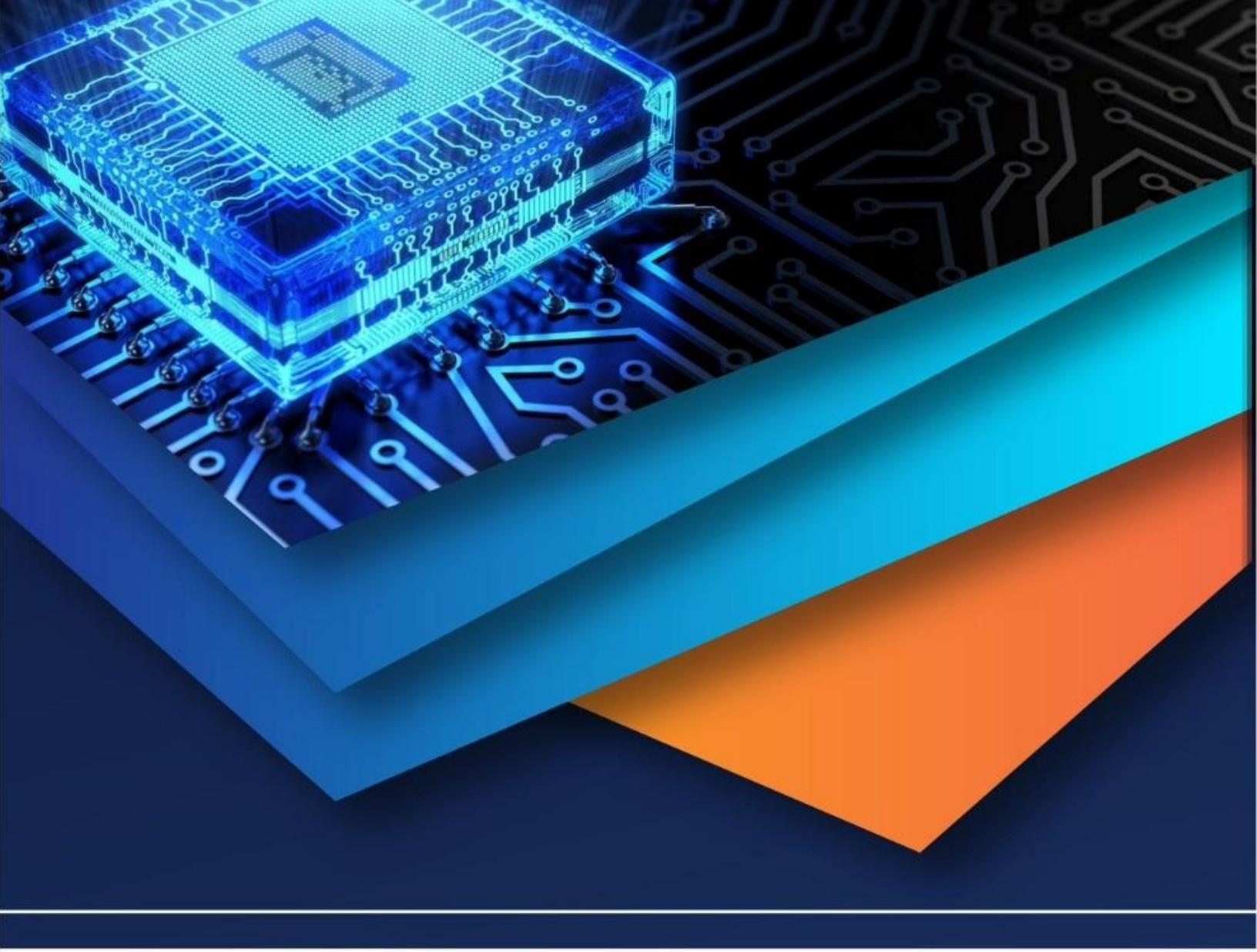
IX. FUTURE ENHANCEMENT

Integration of machine learning: The algorithm could be enhanced by integrating machine learning techniques to better predict resource requirements and optimize resource allocation in real-time. Dynamic workload management: The algorithm could be modified to dynamically manage workloads based on changes in the system's resources, such as CPU and memory utilization. Multi-objective optimization: The algorithm could be enhanced to consider multiple objectives, such as response time, energy consumption, and cost, in the scheduling process. Fault-tolerant scheduling: The algorithm could be modified to handle failures in the system and ensure high availability of cloud services. Hybrid optimization algorithms: The algorithm could be enhanced by combining multiple optimization algorithms, such as genetic algorithms and ant colony optimization, to achieve better performance and efficiency in resource allocation. Integration with containerization technologies: The algorithm could be modified to work with containerization technologies, such as Docker and Kubernetes, to improve resource utilization and enable efficient allocation of resources for microservices-based applications.

REFERENCES

[1] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowtron, "Better never than late: Meeting deadlines in datacenter networks," SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. 50–61, 2011.
[2] A. D. Papaioannou, R. Nejabati, and D. Simeonidou, "The benefits of a disaggregated data centre: A resource allocation approach," in Proc. IEEE GLOBECOM, pp. 1–7, Dec 2016.
[3] A. Tchernykh, U. Schwiegelsohn, V. Alexandrov, and E. ghazali Talbi, "Towards understanding uncertainty in cloud computing resource provisioning," in Proc. ICCS, pp. 1772–1781, 2015.
[4] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in Proc. PAAP, pp. 89–96, 2010.
[5] K.-M. Cho, P.-W. Tsai, C.-W. Tsai, and C.-S. Yang, "A hybrid metaheuristic algorithm for vm scheduling with load balancing in cloud computing," Neural Comput. Appl., vol. 26, no. 6, pp. 1297– 1309, 2015.
[6] S. Rampersaud and D. Grosu, "Sharing-aware online virtual machine packing in heterogeneous resource clouds," IEEE Transactions on Parallel and Distributed Systems, vol. 28, pp. 2046–2059, July 2017.
[7] S. S. Rajput and V. S. Kushwah, "A genetic based improved load balanced min-min task scheduling algorithm for load balancing in cloud computing," in 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN), pp. 677–681, 2016.
[8] S. T. Maguluri, R. Srikant, and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in Proc. IEEE INFOCOM, pp. 702–710, 2012.
[9] S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. M. Abdulhamid, "Resource scheduling for infrastructure as a service (iaas) in cloud computing: Challenges and opportunities," Journal of Network and Computer Applications, vol. 68, no. Supplement C, pp. 173–200, 2016.
[10] J. Ma, W. Li, T. Fu, L. Yan, and G. Hu, A Novel Dynamic Task Scheduling Algorithm Based on Improved Genetic Algorithm in Cloud Computing, pp. 829–835. New Delhi: Springer India, 2016
[11] Al Zain, M., Soh, B., & Pardede, E. (2020). A New Approach Using Redundancy Technique to Improve Security in Cloud Computing. IEEE.
[12] IJCSNS International Journal of Computer Science and Network Security, 10(10), 7-13. Beckham, J. (2018), 'The Top 5 Security Risks of Cloud Computing' available at: http://blogs.cisco.com/smallbusiness/the-top-5-security-risks-of-cloud-computing/
[13] Chen, Y., Paxson, V., & Katz, R. H. (2010). What's new about cloud computing security? University of California, Berkeley Report No. UCB/EECS-2010-5 January, 20(2010).
[14] (2019) International Journal of Advanced Research in Computer Science and Software Engineering Vol 3(5) May -2013. system.
[15] IJCSNS International Journal of Computer Science and Network Security, 10(10), 7-13. Beckham, J. (2018), 'The Top 5 Security Risks of Cloud Computing' available at: http://blogs.cisco.com/smallbusiness/the-top-5-security-risks-of-cloud-computing/
[16] Chen, Y., Paxson, V., & Katz, R. H. (2010). What's new about cloud computing security? University of California, Berkeley Report No. UCB/EECS-2010-5 January, 20(2010), 2020.
[17] (2019) International Journal of Advanced Research in Computer Science and Software Engineering Vol 3(5) May -2013.
[18] M. Abdallah, C. Griwodz, K. T. Chen, G. Simon, P. C. Wang and C. H. Hsu, "Delay-sensitive video computing in the cloud: A survey", ACM Trans. Multimedia Comput. Commun. Appl., vol. 14, no. 3, pp. 1-29, 2018.
[19] L. Columbus, Roundup of Cloud Computing Forecasts 2017, Apr. 2017.
[20] Al-Habob A.A., Dobre O.A., Armada A.G., Muhaidat S. Task scheduling for mobile edge computing using genetic algorithm and conflict graphs. IEEE Trans. Veh. Technol. 2020.
[21] Liang J., Li K., Liu C., Li K. Joint offloading and scheduling decisions for DAG applications in mobile edge computing. Neurocomputing. 2021.
[22] Wang W., Tornatore M., Zhao Y., Chen H., Li Y., Gupta A., Zhang J., Mukherjee B. Infrastructure-efficient virtual-machine placement and workload assignment in cooperative edge-cloud computing over backhaul networks. IEEE Trans. Cloud Comput. 2021.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)