# MANAGEMENT AND SCHEDULING OF RESOURCES IN A CLOUD COMPUTING ENVIRONMENT USING OPTIMIZATION ALGORITHM

**R. Parthiban[1,] Dr.K.Santhosh Kumar[2]**
[1]Research Scholar, Department of CSE, Faculty of Engineering and Technology, Annamalai University, Chidhambaram
[2]Assistant Professor, Department of IT, Annamalai University, Chidambaram,
parthineyveli@gmail.com[1], santhosh09539@gmail.com[2]

## Abstract:

The term "cloud computing" refers to a style of delivering computing resources over the Internet that is abstracted, virtualized, managed, and dynamically demand-driven. Notable capabilities include virtualization, heterogeneity, measured service, pricing, resource pooling, and elasticity. The purpose of this paper is to propose a model for task-based resource allocation in the context of cloud computing. The Process is responsible for allocating resources based on the available resources and user preferences. This paper provides a framework for the analysis of resource scheduling algorithms, which allows computing resources to be allocated based on the priority of each job. Three algorithms' time constraints are compared and contrasted. Many different scheduling algorithms, including Round Robin, Preemptive Priority, and Shortest Remaining Time First, as well as the Resource-Aware Hybrid Scheduling Algorithm and the Hybrid Job Scheduling Algorithm, have been considered. The results of running the proposed method in a cloud data centre simulator, or "cloudsim," demonstrate that it is possible to increased effectiveness in terms of response time, resource utilisation, and overall success rate time. In a simulation study, the method was found to increase the efficiency of resource scheduling by When compared to state-of-the-art works, it improves performance by 7.1% and decreases response time by 35.5%.

## Introduction:

Computing in the cloud is a cutting-edge method that can be effortlessly applied to high-performance computational tasks. Cloud services are highly managed, allowing users to access their data and applications on-demand and only pay for the resources they actually use. Cloud computing is concerned with remote services; it can save money and reduce the need for on-site server facilities. Self-service, network accessibility, shared resources from different servers, measurement services, and elastic services are just some of the features that make cloud computing so attractive. Other deployment models include private, public, hybrid, and community clouds. Managing how and when resources are used is a major difficulty in the field of cloud computing. It is important to ensure at least a minimum QoS when scheduling resources. sustained by employing suitable hardware architecture and algorithms. Part of what makes up the cloud. Typically implemented as a set of virtual machines, infrastructure is responsible for allocating resources and assigning tasks based on what end users have requested (VMs). The broker maps resources by running a scheduling algorithm.

In a distributed system, the scheduling algorithm can take many forms. The majority of them, after going through the proper checks, can be used in a cloud setting. Job scheduling algorithms' primary benefit is that they help increase computing speed and throughput in a system. Scheduling in the cloud is not something that can be done with traditional job scheduling algorithms. As both the number of submitted tasks and the number of available resources increase, it becomes exceedingly challenging to properly assign tasks to the appropriate virtual machines (VMs). Some virtual machines (VMs) may be over-utilized or under-utilized if an improper scheduling algorithm is used, which can have a negative impact on the performance of the cloud system as a whole. The problem of allocating scarce resources is a challenging optimization problem classified as NP-hard (nondeterministic polynomial time). Cloud infrastructure and a scheduler that uses a policy selector to apply one of several scheduling strategies. The cloud workloads have their allotted resources determined by the scheduling policy. Incoming cloud workloads are scheduled by the resource scheduler according to the specifics of each workload. Achieve scheduling for cloud workloads first, then efficiently map cloud workloads to available resources according to scheduling policies. Workloads are sent to be executed via a dispatcher. Workloads are only sent out for execution if they meet the SLA's quality of service requirements. A resource monitor is a tool used to verify the availability and utilisation of scheduled resources. Information on QoS parameters is displayed on the QoS monitor, which is used to ensure that all of the tasks are running within their allotted time frame. Allow us to pretend that time constraint is a Quality of Service

The Quality of Service (QoS) monitor's job is to ensure that tasks are completed on time. If the work schedule doesn't finish by the agreed upon time, the SLA has been broken. The job of the task scheduler in a cloud computing environment is to allocate available resources to the various jobs that need doing. In cloud computing systems, resources can be allocated using a wide variety of job scheduling heuristics.

## Literature Review:

In recent years, "Cloud computing" systems have been the subject of extensive study. Cloud computing is a model for providing networked, on-demand access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or interaction from the service provider.

Singh et al. [5] surveyed extensively on resource management, covering both resource provisioning and resource scheduling. QoS parameters like cost, time, profit, priority, SLA, energy, etc. were used to classify the various RSP methods described. Wadhonkar described the cloud computing architecture and then described the existing schemes to RSP in yet another survey. Plan for future research work proposed by Anton Beloglazov and Rajkumar Buyya, which includes several steps presented in Table. Once all of the proposed optimization stages have their corresponding algorithms developed, they can be combined into a single solution and deployed as part of a production Cloud service like Aneka. The widespread use of the Internet today has opened up a fantastic opportunity for delivering real-time services online.

Using a stochastic integer programming model, Qiang Li and Yike Guo have proposed a framework for optimising cloud computing's resource schedules based on service level agreements. Numerical studies and simulations have been used to evaluate performance. The results of the experiments demonstrate how quickly an optimal solution can be attained.

To shorten the duration of these tasks, Sindhu et al. developed two straightforward algorithms. Shortest Cloudlet to Fastest Processor (SCFP) and Longest Cloudlet to Fastest Processor (LCFP) were the two algorithms used (LCFP). In a cloud computing system, submitted work is referred to as "cloudlets." Specifically, symmetric complementarity flow partitioning (SCFP) sorts jobs by duration and processors by speed. It takes the ordered list of tasks and translates them into a list of processors in the same order. But in LCFP, the tasks are ordered from shortest to longest. As a result of the experiment, it was determined that LCFP outperformed SCFP and FCFS. As Ghanbari et al.

Zhenhua presented a method for distributing work among available resources in Swift.

Wang et al. (2015) present a cloud-based distributed storage system. Keeping an eye on the workload and analysis algorithms were developed to assess whether or not a node was overloaded or underloaded. Resource allocation Algorithm was developed for managing cloud-based virtual machines. Timely Allocation of Resources. Yan intended for his algorithm to be one that could be used in the cloud and would be based on an enhanced version of the particle swarm.

"Wang et al" (2013). It is time and resource dependent, and depends on the characteristics of cloud computing. User budget constraints, a resource scheduling model optimised by particle swarm a coding method was developed.

A new virtual machine load balancing algorithm is proposed and implemented in a Cloud Computing environment using the CloudSim toolkit and the Java programming language, as discussed in a paper by Liang Luo et al.[10]. To implement this algorithm, the VM allocates a different percentage of CPU time to each service in the application. Tasks and requests (application services) are distributed among these VMs in descending order of processing power. We have improved upon the existing VM Load Balancing algorithms by optimising the given performance parameters like response time and data processing time. Cloud-based weighted active load balancing algorithm.

One of the ways to achieve this is through load balancing, while power optimization is achieved. The issues of optimal power allocation and load distribution were discussed by Junwei Cao et al. for a queueing system across multiple cloud servers (2014). Controlling the workload of multiple servers in real time developed by Chun-Cheng Lin et al. using a powerful load balancing algorithm (2014). The The research conducted by Joao Ferreira and colleagues (2013) and the efforts of E. Pinto Neto. 2017 by et al. created a load-balanced design.

The best data centre algorithms for distribution. A data-sharing architecture that is built on an object-oriented a new method based on a highly distributed framework for accounting for information has been developed efficiency in scheduling was increased by Smitha Sundareswaran et al. (2012). Setting priorities for a system'sgreedy heuristic in a dispersed manner, as Olivier Beaumont et al. (2012) discussed technique, guaranteeing efficiency and l ow prices. Specifically, we create an algorithm for scheduling that we call Linear Scheduling for Tasks and Resources (LSTR). This algorithm is responsible for scheduling activities and assets. Nimbus and Cumulus services are imported to a server node to set up the infrastructure as a service cloud. The When combined with a scheduling algorithm, virtualization can increase resource utilisation and boost cloud performance.

Thomas et al. proposed a credit based task scheduling method, outlining algorithms for computing credits of tasks based on their length and priority. The length credit was calculated by determining the variance between the actual duration of tasks and the mean duration. In this context, "priority credit" means the weighted sum of each task's priority value. Two factors, "length credit" and "priority credit," combined to form the total credit. Credit systems were evaluated separately and then in combination, with the results compared for factors such as length and priority. It was determined that the group effort was superior in terms of achieving minimum makespan.

## Proposed work:

When something is scheduled, it means that it will occur at a predetermined time. When it comes to allocating resources, distributed computing offers a wide variety of scheduling algorithms to choose from. With proper authorization, the distributed system can make use of a wide variety of algorithms. Maximum throughput is the goal of the scheduling algorithm. Regular methods fall short of providing the desired efficiency in a cloud setting. Scheduling algorithms in the cloud were categorised as either Batch/sequential or Online/random. When data enters the system in batch/sequential mode, all available resources are lined up in a row. As such, the algorithm will activate at the predetermined interval. Fcfs, RR, min-min, and max-min are all algorithms that can be implemented in batch mode.

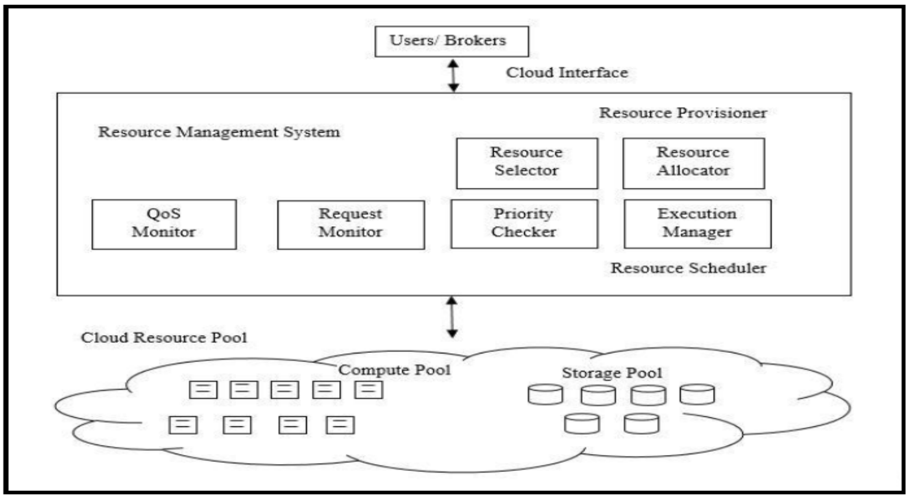

**Figure Cloud Computing Resource Scheduling**

**Scheduling Resources in the Cloud**

## ALGORITHMs FOR RESOURCE SCHEDULE PLANNING

When cloud computing was first introduced, it borrowed heavily from grid and cluster scheduling practises. Such methods include "First Come, First Served" (FCFS), in which services are provided in the order in which they were received by the system, "Round Robin" (RR) scheduling, in which a specified time quanta decides the period for which tasks will be executed in one go, "Min-Min" algorithm, in which the task with the shortest completion time is allocated to the resource where it takes the shortest execution time, and "Max-Min" algorithm, in which the task with the longest completion time These algorithms fared well, but they ignored cloud parameters and other QoS considerations.

An effective mechanism for scheduling tasks can help organisations better serve their customers and make better use of their resources. Providers of cloud computing services frequently field numerous requests from users with widely varying specifications and preferences. Some jobs call for more computing power and bandwidth, while others must be completed at a lower cost and with fewer resources. Tasks submitted by users can be compared using the comparison matrix method once the cloud computing service providers have received them. To ensure that all parties are satisfied, cloud service providers discuss and agree upon task requirements with their customers.

When allocating work to virtual machines (VMs) in the cloud, it is important to keep in mind some of the underlying assumptions.

- To ensure that more work gets done, there should be more virtual machines (VMs) than tasks.
- Only one virtual machine (VM) resource is used for each task.
- Tasks of varying sizes (from quick to lengthy).
- After a task's execution has begun, it will continue without interference.
- Virtual machines are capable of operating with their own set of resources and under their own management.
- The available virtual machines (VMs) are single-purpose and cannot be used for multiple purposes at once. What this means is that the virtual machines (VMs) will only focus on the tasks at hand until they are finished.

Because of the cloud's long execution times and limited resources, resource scheduling is a rapidly developing field of study. Categories of resources are assigned varying sets of criteria and parameters for resource scheduling.

**Evolutionary Approaches for Resource Scheduling**

Since the scheduling problem is an NP-hard problem, it is difficult, if not impossible, to find a good solution by employing purely linear methods. Consequently, there has been a lot of focus on algorithms that mimic natural processes in the scientific community. In contrast to deterministic methods, evolutionary ones have an algorithmic complexity that scales polynomially with the size of the problem. In this section, we will discuss the most widely used evolutionary algorithms, such as Autonomous Agent-Based Load Balancing, Hybrid Job Scheduling Algorithm, Hybrid Job Scheduling Algorithm, Firefly Algorithm, and Optimal Resource Allocation Technique, and the research that has been conducted utilising these methods in a Cloud Computing setting.

Load balancing in a cloud infrastructure using autonomous agents. Workload balancing is essential for effective scheduling, as evidenced by the growing number of daily cloud users. The authors of this paper propose an agent-based automatic load balancing method for managing fluctuating cloud workloads. If there is a mismatch between the calculated load in the VM and the DC, the agent will look for another possible VM and DC host. It is clear from the authors' experiments that the proposed algorithm performs admirably.

They use a genetic algorithm as their starting point and then tweak it using fuzzy logic to cut down on the number of times they have to generate a population, making it a hybrid job scheduling algorithm. To determine the fitness value of each chromosome, two types were designed using different QoS parameters and a fuzzy logic approach. The new method reduces the system's overall execution time by about half as much as well as its execution cost by about 45%.

The Firefly Algorithm is an approach to the load balancing problem. It manages the group of requests and hosts them on the appropriate machines. Due to its alluring qualities, the firefly algorithm serves as an inspiration for this. The method proposed is built up from three parts: index calculation, schedule list, and implementation. As demonstrated by the experiments, the proposed method works. The 0.934 ms target time has been met.

Method for Efficient Use of Available Resources:

The proposed algorithm took into account both data transfer rates and computing power. Here's how ORAT operates: The proposed architecture uses optimization techniques to allocate resources among a set of servers. With every user request, these servers apply their methods and resources. When an allocation is complete, the server will record the current usage and update the status accordingly. Whenever there is a step where improvement is necessary, it is implemented immediately.

**Result Analysis:**

| **Algorithm** | **Objective Criteria** | **Description** | **Experimental Environment** | **Experimental Scale** | **Results Compared** |
|---|---|---|---|---|---|
| Resource Aware Scheduling Algorithm | Execution Time | Incorporates the process of simulated annealing in the PSO algorithm to improve convergence. | Cloudsim | 70- 440 tasks | GA, SA, ACO, PSO |
| Resource-Aware Hybrid Scheduling Algorithm | Cost | Improves PSO by adding crossover and mutation and SPV. | Cloudsim | 55-70 tasks<br>12-27 resources | Basic PSO |
| Hybrid Job scheduling Algorithm | Cost | Considers deadline satisfaction as the constraint. | Cloudsim | Workflow with 9 tasks<br>3 resources | SCS and IC-PCP |

| | | | | | |
|---|---|---|---|---|---|
| | | Uses basic Hybrid Algorithm to optimize cost and time. | | | |
| Optimal Resource Allocation Technique | Throughtput | Uses a new metaheuristic algorithm called as Optimal Resource Allocation Technique to increase speed of convergence | Cloudsim | Workflow with 9 tasks<br>3 resources | Basic PSO |

**Conclusion**:

The operational cost of the service provider and the cloud user can both be impacted by resource scheduling in the cloud. Resource scheduling is an active area of study, with numerous studies focusing on various topics such as load balancing, makespan, workload priority, resource availability, and cost. To optimise resource utilisation in cloud computing environments and reduce the overall scheduling execution time (makespan), we employed a heuristic. The problem formulation and modelling proposed solution take into account a heterogeneous environment in terms of the number and types of servers used in each cluster. A new algorithm for resource scheduling and a comparison to existing algorithms will be proposed in a later improvement. Processor performance can be optimised for user requests in order of priority. The proposed new algorithm for resource scheduling and comparison to currently used algorithms will be part of a later improvement. When a user makes a request, the processor can be optimised for maximum efficiency by focusing first on completing the request.

**References:**

1. Bhupesh Kumar Dewangan, and Amit Agarwal. "Credential and Security Issues on Cloud Service Models" in the proceeding of 2nd IEEE International Conference on Next Generation Computing Technology India pp 1-8. 2017 .
2. D. Minarolli and B. Freisleben, "Uitlity–based Resource Allocations for virtual machines in cloud computing", (IEEE, 2011), pp.410-417.
3. Tychalas, Dimitrios, and Helen Karatza, "A Scheduling Algorithm for A Fog Computing System with Bag-of-Tasks Jobs: Simulation and Performance Evaluation," Simulation Modelling Practice and theory, Vol. 98 , Pp.101982, 2020.
4. Jamil, Bushra, Mohammad Shojafar, Israr Ahmed, Atta Ullah, Kashif Munir, and Humaira Ijaz, "A Job Scheduling Algorithm for Delay and Performance Optimization in Fog Computing," Concurrency and Computation: Practice and Experience, Vol. 32, No. 7 , Pp. E5581, 2020.
5. Hitoshi Matsumoto, Yutaka Ezaki," Dynamic Resource Management in Cloud Environment", July 2011, FUJITSU science & Tech journal, Volume 47, No: 3, page no: 270-276.
6. Ghalem Belalem, Samah Bouamama and Larbi Sekhri, "An Effective Economic Management of Resources in Cloud Computing", March 2011, JOURNAL OF COMPUTERS, Vol. 6, No. 3, page no: 404-411.
7. Anton Beloglazov and Rajkumar Buyya," Energy Efficient Resource Management in Virtualized Cloud Data Centers", 2010 10th IEEE/ACM International Conference on

Cluster, Cloud and Grid Computing, 978-0-7695-4039-9/10,IEEE, DOI 10.1109/CCGRID.2010.46, page no: 826-831.

8. Venkatesa Kumar. V and S. Palaniswami," A Dynamic Resource Allocation Method for Parallel Data Processing in Cloud Computing", 2012, Journal of Computer Science 8 (5), ISSN 1549-3636, Science Publications, page no: 780-788.
9. Weiwei Lina, James Z. Wangb, Chen Liangc and Deyu Qia, "A Threshold-based Dynamic Resource Allocation Scheme for Cloud Computing", 2011, 1877-7058, Elsevier Ltd, PEEA 2011 Doi:10.1016/j.proeng.2011.11.2568, page no: 695 – 703.
10. Salot, P. (2013). A survey of various scheduling algorithm in cloud computing environment. International Journal of Research in Engineering and Technology, 2(2), 131-135.
11. Singh, S., & Chana, I. (2016). A survey on resource scheduling in cloud computing: Issues and challenges. Journal of grid computing, *14*(2), 217-264.
12. Bacigalupo DA, Hemert J, Chen X, Usmani A, He L, Donna N, Wills G, Gilbert L, Jarvis S (2011). Managing dynamic enterprise and urgent workloads on clouds using layered queuing and historical performance models. Published in proceedings of Simulation Modelling Practice and Theory, Volume 19, pp 14 79-1495.
13. Banerjee C, kundu A, Bhaumik S, Babu RS, Gupta RD(2012). Framework on Service based Resource Selection in Cloud Computing. Published in International Journal of Information Processing and Management,Volume 3, Number 1, pp. 17-25.
14. Baomin X, Chunyan Z, Enzhao H, Bin H(2011). Job scheduling algorithm based on Berger model in cloud environment. Published in Elseivier Advances in Engineering Software, Volume 42, pp 419–425.
15. Rasooli, A., Down, D.: An adaptive scheduling algorithm for dynamic heterogeneous Hadoop systems. In: Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research, pp. 30–44. IBM Corp (2011)
16. Lee, Z., Wang, Y., Zhou, W.: A dynamic priority scheduling algorithm on service request scheduling in cloud computing. In: 2011 International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT), vol. 9, pp. 4665–4669. IEEE (2011)
17. Hwang, J., Wood, T.: Adaptive dynamic priority scheduling for virtual desktop infrastructures. In: Proceedings of the 2012 IEEE 20th International Workshop on Quality of Service, p. 16. IEEE Press (2012)
18. Fang, Y., Wang, F., & Ge, J. (2010, October). A task scheduling algorithm based on load balancing in cloud computing. In International Conference on Web Information Systems and Mining (pp. 271-277). Springer, Berlin, Heidelberg.
19. Sindhu, S., & Mukherjee, S. (2011). Efficient task scheduling algorithms for cloud computing environment. In High Performance Architecture and Grid Computing (pp. 79-83). Springer, Berlin, Heidelberg.
20. Ghanbari, S., & Othman, M. (2012). A priority based job scheduling algorithm in cloud computing. Procedia Engineering, 50, 778-785.
21. Li, Q. (2012). Applying Integer Programming to Optimization of Resource Scheduling in Cloud Computing. JNW, 7(7), 1078-1084.
22. Wu, X., Deng, M., Zhang, R., Zeng, B., & Zhou, S. (2013). A task scheduling algorithm based on QoS-driven in cloud computing. Procedia Computer Science, 17, 1162-1169.
23. Abdullah, M., & Othman, M. (2013). Cost-based multi-QoS job scheduling using divisible load theory in cloud computing. Procedia computer science, 18, 928-935.
24. K. C. Gouda, T. V. Radhika and M. Akshatha, "Priority based resource allocation model for cloud computing", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 2, 2013.

25. S. Irugurala and K.S. Chatrapati, “Various Scheduling Algorithms for Resource Allocation In Cloud Computing”, the International Journal of Engineering And Science (IJES), Volume 2, Issue 5, Pages 16-24, 2013.
26. M. A. Sharkh, M. Jammal, A.Shami, and A. Ouda, “Resource Allocation in a Network-Based Cloud Computing Environment: Design Challenges”, IEEE Communications Magazine • November 2013.
27. P. Mell and T. Grance, “Useful Information for Cloud Adopters”, Special Publication 800-145, NIST US Government Cloud Computing Technology Roadmap, Release 1.0 (Draft) November 2011.
28. H. Goudarzi and M. Pedram, “Maximizing Profit in Cloud Computing System Via Resource Allocation”, IEEE 31st International Conference on Distributed Computing Systems Workshops 2011: pp, 1-6.
29. C. Santos, X. Zhu, and H. Crowder, “A mathematical optimization approach for resource allocation in large scale clusters”, Technical Report HPL-2002-64, HP Labs, March 2002.
30. L. Jiayin, Q. Meikang, N. Jian-Wei, C. Yu, “Adaptive resource allocation for preemptable jobs in cloud systems” Intelligent Systems Design and Applications (ISDA), IEEE, 2010, pp.31-36. [15] A. Kundu, C. Banerjee, S.K. Guha, A. Mitra, S. Chakraborty, C. Pal and R. Roy,” Memory Utilization in Cloud Computing using Transparency”, 5th International Conference on Comp