# A Resource Scheduling with Load Balancing in Cloud Environment Using Particle Swarm Optimization

R. Sandhiya and D. Radhika

**Abstract---** Cloud computing is a perfect platform for executing complex applications in any network. Dynamic resource scheduling for multi-objective schedules has an economic background process for fluctuating workloads. Execution assessment of Cloud Computing foundations is required to anticipate and evaluate the money saving advantage of a procedure portfolio and the relating Quality of Service (QoS) experienced by clients. Thus, focusing on Load balancing in the cloud computing environment has a significant impact on the performance.

This paper takes care of the multi-objective resource provisioning scheme for handling multiple task classes for various workload facility. In this proposal, the project is using the Best Partition Searching for distributing a file system to another cloud environment. In this data's are split based on domain and stored in cloud storage. From this, the data's are shown based on the field which user search. We propose a particle swarm optimization is an artificial intelligence to perform continues workload scheduling for various attributes.

So that balancing of cloud (main server) is reduced with low of cost while the cloud user gets the file from the cloud server and then User also getting an efficient and improves user satisfaction using our proposed method. These are implemented using particle swarm optimization for allocating a user downloading a file in the cloud environment. The proposed plan has produced more scheduling performance and low time complexity.

*R. Sandhiya, PG Student, Department of CSE, Vivekanandha College of Engineering for Women (Autonomous), Tiruchengode, India.*
*D. Radhika, Assistant Professor Department of CSE, Vivekanandha College of Engineering for Women (Autonomous), Tiruchengode, India.*

**Keywords---** Cloud Computing, Best Partition Searching for Distributing a File System, Particle Swarm Optimization, and Workload Scheduling.

## I. INTRODUCTION

Distributed storage has risen as a promising answer for giving universal, helpful, and on-request gets to a lot of information shared over the Internet. Today, a large number of clients are sharing individual information, for example, photographs and recordings, with their companions through interpersonal organization applications dependent on distributed storage every day. Business clients are likewise being pulled in by distributed storage because of its various advantages, including lower cost, more noteworthy spryness, and better asset usage.

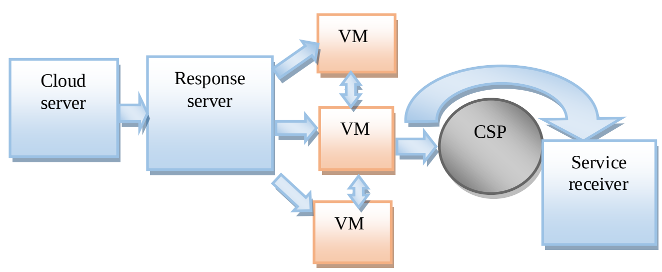


Figure 1: Cloud Virtual Service

We can utilize scalable, distributed computing environments within the confines of the internet, a practice known as cloud computing. In this new world of computing, users are universally required to accept the underlying premise of trust. Within the cloud computing world, the virtual environment lets users' access computing power that exceeds that contained within their physical worlds. Cloud computing is the process of providing computer facilities via the internet. And it's provided us with a better and efficient way to access information promptly and also increases storage of capacity for the user in.

### *A. Securing Transportation as a Service*

The Iaas model is working to computer networking and data storage, other resources in a virtualized environment.

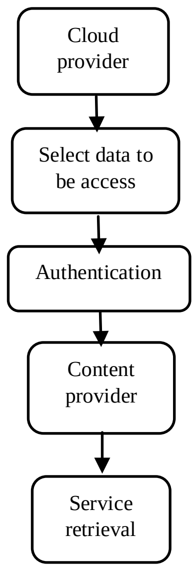


Figure 2: Transportation Cloud Service

Amazon's Elastic Compute cloud is one of an excellent example of Iaas, at the cloud infrastructure level, CSP can implement network security with intrusion-detection systems, firewalls, antivirus programs, distributed denial-of-service suspicion and so on. Securing policy as a service. Cloud platforms built in Iaas with system integration and virtualization middleware. And these platforms can be used to users for implementing user-built software applications onto the cloud infrastructure using provider-supported programming languages.

### *B. Trusted Cloud Computing Over Data Centers*

Security aware cloud architecture and this used to identify the protection mechanisms needed. Intruder detection action should be implemented by using this architecture.

As far as the cloud is concerned, it has the most effective impact on internet-based access for the data storage. So to maintain sufficient integrity and security from any internal and external attack, we must design a cloud storage area in a well-structured manner with reliable data protection against the vulnerability. We can specify some unique form from cloud such as SaaS, PaaS, IaaS, and AaaS explains this environment.

### *C. Scheduling*

Advances in equipment and programming innovations have prompted expanded enthusiasm for the utilization of substantial scale parallel and appropriated frameworks for database, continuous, guard, and extensive scale business applications. The working framework and administration of the simultaneous procedures comprise essential parts of the parallel and appropriated situations. A standout amongst the most noteworthy issues in such frameworks is the improvement of pragmatic systems for the dissemination of the procedures of a parallel program on various processors. The issue is the means by which to disseminate (or plan) the procedures among preparing components to accomplish some execution goal(s, for example, limiting execution time, limiting correspondence delays, and boosting asset use. From a framework's perspective, this conveyance decision turns into an asset administration issue and ought to be viewed as a basic factor amid the outline periods of multiprocessor frameworks.

### *D. Static Scheduling*

In static scheduling, the assignment of tasks to processors is done before program execution begins. Information regarding task execution times and processing resources is assumed to be known at compile time. A job is always executed on the processor to which it is assigned; that is, static scheduling methods are processor non-preemptive. Typically, the goal of static scheduling methods is to minimize the overall execution time of a concurrent program while reducing communication delays.

### *E. Dynamic Scheduling*

Dynamic planning depends on the redistribution of procedures among the processors amid execution time. This redistribution has performed by exchanging errands from the intensely stacked processors to the softly stacked processors (called stack adjusting) with the point of enhancing the execution of the application.

### *F. Distributed Scheduling*

In most cases, work in distributed scheduling concentrates on global schedule because of the architecture of the underlying system. It defines a taxonomy of task placement algorithms for distributed systems. The two major categories of global algorithms are static and dynamic.

Static algorithms make scheduling decisions based purely on information available at compilation time. For example, the typical input to a static algorithm would include the machine configuration and the number of tasks and estimates of their running time.

### *G. Load Balancing*

The load balancing operations may be centralized in a single processor or distributed among all the processing elements that participate in the load balancing process. Many combined policies may also exist. For example, the information policy may be centralized, but the transfer and placement policies may be distributed. In that case, all processors send their load information to a central processor and receive system load information from that processor. However, the decisions regarding when and where a job should be transfer are made locally by each processor.

## II. RELATED WORK

A broad introduction to the resource allocation problems in cloud systems including Inter-Clouds and Mobile Clouds as well as proposed solutions to these problems. Allocation of computing and network resources to cloud tasks requires innovative approaches in each case of cloud data centers. The problem is highly challenging especially in the affairs of distributed and federated clouds [1].Fine-grained constraints on the computation, storage, and networking resources are required to support mission-critical enterprise use-cases at a reasonable cost. They are spelled out by service level agreements (SLAs) between the application and the cloud platform. In the technique, they motivate the need for more complex performance requirement support with two use cases, electric utility metering and control, and public safety. They describe an application management tool, called the Abstract Service Manager (ASM), which is designed to allow the expression of performance requirements in the automated deployment of distributed cloud-native applications. [2].

Adoption of a distributed shared memory (DSM) programming paradigm will be one approach to ease the transition, through the use of Partitioned Global Address Space (PGAS) languages. This paper explores initial results from the adoption of a PGAS language, Unified Parallel C, in programming a representative private cloud based on Eucalyptus [3].Cloud computing enables users (clients) to outsource large volume of their data to cloud servers. Securely distributed cloud storage schemes ensure that multiple servers store these data in a reliable and unhampered fashion. They identify some challenges while extending this idea to accommodate append-only data. Then, they propose a secure distributed cloud storage scheme for append-only data that addresses the problems efficiently. Moreover, the client need not download any data (or parity) block to update the tags of the modified parity blocks residing on the servers [4].

To make the most efficient use of the resources, an optimized scheduling algorithm to achieve the optimization or sub-optimization for cloud scheduling problems. There are three main steps in the scheduling: (1), the system set an idle resource list and VM request list, update them at the initial time, and each time new VM requests come, or VMs are shut down, or physical resources change are being detected; (2), use the PGA to find the optimal allocation sequence; (3), launch the specified physical machines to the VM requests [5].Map-Reduce has become a prominent Parallel processing model used for analyzing large-scale data. Map-Reduce applications are increasingly being deployed in the cloud along with other applications sharing the same physical resources. Map-Reduce has to consider various other parameters like energy efficiency and meeting SLA goals besides achieving performance when executing jobs in cloud environments. In the technique, they have classified Map-Reduce Scheduling as Cluster based Scheduling and Objective based Scheduling [6].

They consider the problem of opportunistically scheduling low-priority tasks onto underutilized computation resources in the cloud left by high-priority tasks. To avoid conflicts with high-priority tasks, the scheduler must suspend the low-priority functions (causing waiting), or move them to other underutilized servers (causing migration), if the high-priority functions resume. They propose an efficient heuristic scheduling policy by formulating the problem as restless Multi-Armed Bandits (MAB) under relaxed synchronization [7].

Tasks scheduling problem is the key challenge in cloud computing system. For reducing the execution cost of workflow tasks scheduling under the deadline and the budget constraint, a workflow tasks scheduling algorithm based on a genetic algorithm in cloud computing has proposed. In our algorithm, each task is assigned priority by a top-down leveling method. By this top-down leveling method, all workflow tasks divided into different levels, which can promote the parallel execution of workflow tasks [8].The cloud platform is the model of parallel and distributed computing. It offers the facilities by pay-as per-usage policy. Also provides the platform for a high-performance application like scientific applications. Virtualization, elasticity, and pay by use are the essential features of cloud computing. Since the cloud provides the facility of running multiple tasks simultaneously, an eminent scheduling algorithm is needed for better performance [9].

Online energy-efficient scheduling of real-time virtual machines (VMs) for Cloud data centers. Each request had associated with a start time, an end-time, processing time and demand for a Physical Machine (PM) capacity. Offline scheduling to minimize busy time is NP-hard already in the particular case where all jobs have the same processing time and can be scheduled at a fixed time interval [10]. Clouds are becoming an increasingly popular infrastructure for enabling large-scale data-intensive scientific and business applications. Cloud provides a suitable environment for processing Big Data applications to handle large volumes of data and parallel processing of data. Workflows are used to allocate and schedule execution of Big Data applications in an optimized manner [11].The current method is to implement cloud computing for complex applications and refined in remote data centers are supported on parallel processing capability.

The method of parallel job scheduling was mainly based on information processed by the cloud center in a specified period. But at the same time, it not only maximize the communication cost but also results in the improper utilization of nodes in a data center with inadequate responses to the parallel workload with many data center in the cloud [12].

## III. PARTICLE SWARM OPTIMIZATION

Focusing on Load balancing and work schedules in the cloud computing environment has a significant impact on the performance. Proper load balancing makes cloud computing more efficient and improves user satisfaction. This article introduces a better load balance model for the public cloud based on the cloud partitioning concept with a switch mechanism to choose different strategies for different situations. In this proposal, the project is using the Best Partition Searching for distributing a file system to another virtual cloud environment.

This project helps to provide the file to the user in the cloud so that balancing of cloud (central server) is reduced with low of cost while the cloud user gets the file from the cloud server and then User also getting an efficient and improves user satisfaction using our proposed method. These are implemented using particle swarm optimization (PSO) for allocating a user downloading a file in a virtual cloud environment.

### A. Block Diagram

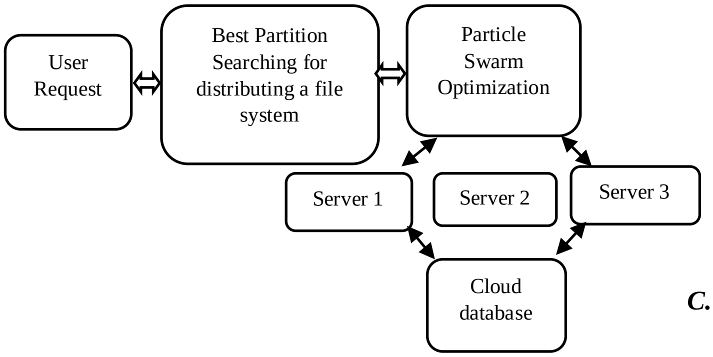


Figure 3: Work Scheduling Process

In the above figure describes the Best partition searching for distributing a file system in this the search is based on the user requested domain. Particle Swarm Optimization perform based on the workload for the server. Multiple servers perform the searching process. This server will get the data from the central server a send it to the user.

PSO is artificial intelligence, based on the collective behavior of decentralized, self- organized systems. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized.

### B. Algorithm Steps

**Step 1:** The TMA Load Balancer performs load balancing by updating, maintaining two index tables.

- Available Index: Status of VMs is available is '0.'
- Busy Index: Status of VMs is not available '1'.

In the beginning, all VMs are updated in the "Available Index" table, and the "Busy Index" table is empty.

**Step 2.** The Data Center Controller receives a new request.

**Step 3:** Data Center Controller queries to the TMA Load Balancer for next allocations.

**Step 4:** TMA Load Balancer detects and sends VM ID (VM) from the top down in the "Available Index" table of the Data Center.

The Data Center Controller sends the request to the specified VM by that ID.

- The Data Center Controller informs the TMA Load Balancer for a new allocation.
- The TMA Load Balancer will update the VM into the Busy Index and wait for the new request from the Data Center Controller.
- In case, if the table "Available Index" is empty (all VMs unavailable).

### C. Data Flow Diagram

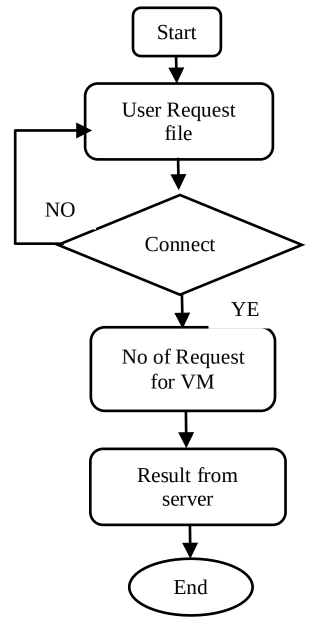


Figure 4: Data Flow Diagram of Scheduling

Above figure shows the processing in the server to access the particular data. At first user request is collected and connect is checked, if a connection is available then request is sent to the scheduling process, if no connection again applications are received from the user. In the scheduling process, the available virtual machine is used to take data from the server.

### D. Multilevel Priority

The scheduling on the VM's cloud data centers which will configure the switching on bin packing mechanism. In the proposed scheme, each VM's, except those at the last level of the virtual hierarchy in the zone-based dynamic topology consideration in cloud data centers, has three levels of priority queues based on scheduling, job allocation, bin packing. We develop a set of heuristics that prevent burden in the system effectively while saving energy used.

### *E. Centralized Partitioning*

Data partitioning splits the data input into multiple chunks. Partitioning can occur at two Points in the provisioning of the significant data service. It could be done initially at a central location with a subdividing service. At this stage, data is partitioned logically no actual movement of information is initially done, but an index is kept that limits the beginning of a partition and end of the previous one.

### *F. Data Distribution*

Data distribution is a critical task distribution should maximize transfer rates and minimize information redundancy. VMs algorithm to reduce the burden in the virtual machine. Set of heuristics that prevent trouble in the system effectively while saving energy used. The second component is a central data repository which has the files to be deployed onto all the VMs.

### *G. Particle Swarm Optimization*

The specified number of VMs are created and started within the selected cloud provider. This function performs the actual deployment of the virtual infrastructure, installation, and configuration of the software installed in the VMs in our specific case it is our VMs and the peers for distributing the partitioned file. Automatic Deployment Layer using the configuration parameters taken from the user. In this scenario, one of the VMs is randomly chosen to be the master, and the others become slaves of the application.

### *H. Virtualization*

The highest-priority of visualization queue and can preempt data in other lines based on the selection of cloud data providers Server virtualization is the partitioning of a physical server into smaller virtual servers to help maximize your server resources. In server virtualization, the resources of the server itself are hidden, or masked, from users, and software is used to divide the physical server into multiple virtual environments, called virtual or private servers. Applications run on virtual servers that are constructed using virtual machines, and one or more virtual servers are mapped onto each physical server in the system.

## IV. RESULT AND DISCUSSION

In this simulated result is shown the comparison of processing time and response time between ant colony technique and Particle Swarm Optimization technique. From this, the proposed method shows the improved period for accessing the data from the server and reduce time delay.

Table 1: Processing time between Ant Colony and PSO

| | *Resource 1 Time (ms)* | *Resource 2 Time (ms)* | *Resource 3 Time (ms)* |
|---|---|---|---|
| Ant Colony | 6 | 4.4 | 5 |
| PSO | 4 | 3.8 | 3.9 |

In the above table shows the values of processing time difference between an ant colony and PSO.

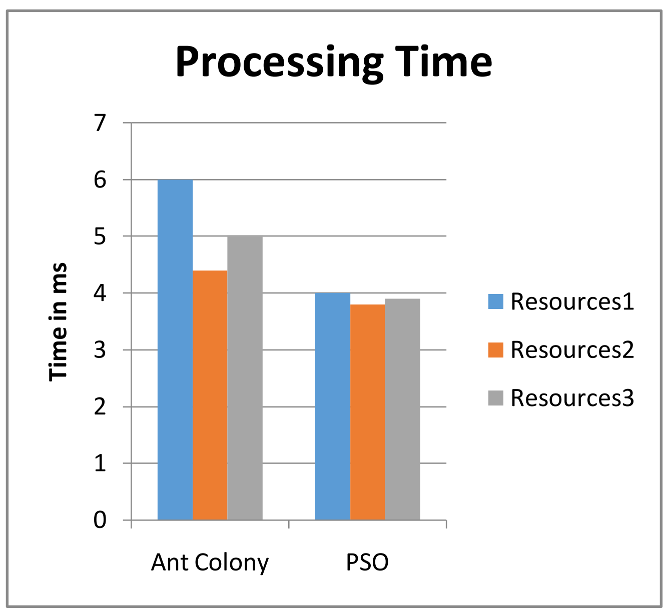


Figure 5: Processing Time between existing and proposed system

Above figure shows the processing time comparison between the ant colony and particle swarm optimization for resource 1, resource 2, and resource 3.

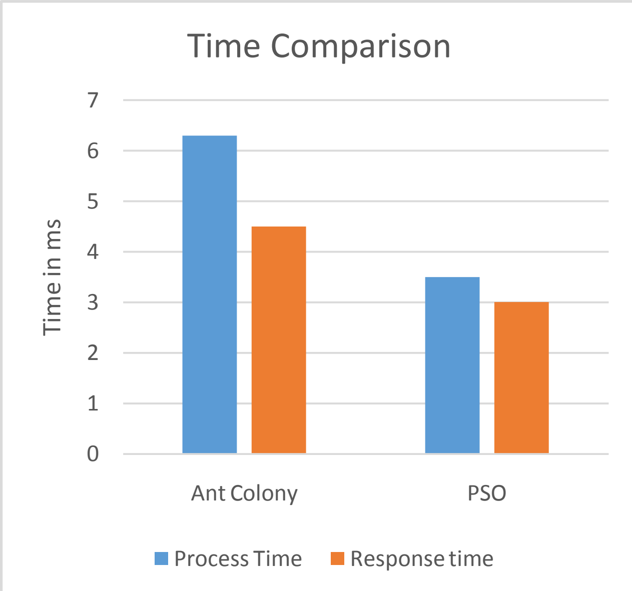


Figure 6: Time Comparison between Ant Colony and PSO

Above figure shows the response time and processing time between an ant colony and particle swarm optimization. From this time delay is reduced and accessing data is improved.

## V. Conclusion

We proposed Best Partition Searching for distributing a file system to reduce the scheduling time in a cloud environment. From this technique, the file search is performed based on the domain which user is search. In this, the file is stored in the cloud based on their domain. We propose a Particle Swarm Optimization approach considers various attributes to schedule the workloads. This artificial intelligent perform an automatic work scheduling. In this workload is scheduled for the server based on the user's approach. The cloud servers work schedule is based on the workload for the particular server. Scheduling is performed to reduce the time delay for the user request to search and provide a particular data. If many users are searching data in the server, then the server workload is split into multiple to provide an efficient result. Our result shows more scheduling performance and low time complexity. In the future, we can improve the period for accessing the data and implement the security to access the data securely. That transparently migrates only the working set of an idle VM and support green computing by optimizing the number of servers in use. We use the maximum precedence algorithm to reduce the burden in a virtual machine.

## References

[1] A. Aral, “Modeling and optimization of resource allocation in distributed clouds”, In IEEE International Conference on Cloud Engineering Workshop (IC2EW), Pp. 210-212, 2016.

[2] D. Mazmanov, C. Curescu, H. Olsson, A. Ton and J. Kempf, “Handling performance sensitive native cloud applications with distributed cloud computing and SLA management”, IEEE/ACM 6th International Conference on Utility and Cloud Computing (UCC), Pp. 470-475, 2013.

[3] A. Anbar, V.K. Narayana and T. El-Ghazawi, “Distributed shared memory programming in the cloud”, In12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), Pp. 707-708, 2012.

[4] B. Sengupta, N. Nikam, S. Ruj, S. Narayanamurthy and S. Nandi, “An Efficient Secure Distributed Cloud Storage for Append-only Data”, arXiv preprint arXiv:1805.02970, pp. 1-11, 2018.

[5] Z. Zheng, R. Wang, H. Zhong and X. Zhang, “An approach for cloud resource scheduling based on Parallel Genetic Algorithm”, IEEE 3rd International Conference on Computer Research and Development (ICCRD), Vol. 2, Pp. 444-447, 2011.

[6] S. D'Souza and K. Chandrasekaran, “Analysis of Map Reduce scheduling and its improvements in cloud environment”, IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), Pp. 1-5, 2015.

[7] T. He, S. Chen, H. Kim, L. Tong and K.W. Lee, “Scheduling parallel tasks onto opportunistically available cloud resources”, IEEE 5th International Conference on Cloud Computing (CLOUD), Pp. 180-187, 2012.

[8] Y. Cui and Z. Xiaoqing, “Workflow tasks scheduling optimization based on genetic algorithm in clouds”, IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Pp. 6-10, 2018.

[9] M. Shanmugasundaram, R. Kumar, D. Shinde and H.M. Kittur, “Comparative analysis of scientific workflow scheduling in cloud environment”, In Innovations in Power and Advanced Computing Technologies (i-PACT), Pp. 1-7, 2017.

[10] W. Tian, R. Xue, J. Cao, Q. Xiong and Y. Hu, “An energy-efficient online parallel scheduling algorithm for cloud data centers”, In IEEE Ninth

World Congress on Services (SERVICES), Pp. 397-402, 2013.

[11] B.K. Rani and A.V. Babu, "Scheduling of Big Data application workflows in cloud and inter-cloud environments", IEEE International Conference on Big Data (Big Data), Pp. 2862-2864, 2015.

[12] C. Antony and C. Chandrasekar, "Performance study of parallel job scheduling in multiple cloud centers", IEEE International Conference on Advances in Computer Applications (ICACA), Pp. 298-303, 2016.