Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 5, 2021, Pages. 104-114
Received 15 April 2021; Accepted 05 May 2021.

An Efficient Framework for Resource Optimization and Task Scheduling in Cloud Computing

Naresh T[1], Dr.A.Jaya Lakshmi [2], Dr. Vuyyuru Krishna Reddy[3]
[1,3]K.L.University Vijayawada.
[2]PVP Siddhartha Institute of Technology,Vijayawada
*Corresponding Author : naresh1060@gmail.com

Abstract.
Cloud computing is the provision of computer services through the cloud (Internet), such as servers, storage, databases, networks, software, analytics, intelligence, etc.Cloud computing provides an alternative to local data centers. We can rent out any necessary services.Cloud optimization is the action of correctly selecting the right resources and assigning them to workloads or applications. Efficiency can be achieved when workload performance, compliance, and cost are balanced with the most suitable infrastructure in real time, correctly and continuously. Load balancing in the cloud is the distribution of workload across multiple computer sources. DNS load balancing uses software or hardware to perform functions, while cloud load balancing uses services provided by various computer network companies. Resource allocation (RA) is an important aspect of cloud computing. It can provide cloud resources to cloud consumers and provide services based on traffic. The cloud resource manager is in charge for allocating available resources to tasks for execution in an effective manner, thereby improving system performance, reducing response time, shortening the validity period, and effectively using resources. Therefore, resource mapping is predominant matter when considering task allocation and load balancing. This article aims to focus on solving problems related to task allocation by using the recommended optimization algorithm and applying the recommended load balancing algorithm, and then implement resource mapping.

Keywords: Cloud computing, Cloudlet, Data Center, Resource Optimization, Resource mapping, Load balancing, Task job scheduling, task scheduling, optimization.

INTRODUCTION:

Cloud computing is a revolutionary service method that uses a "pay-as-you-go" utility computing model to provide customers with on-demand applications, platforms, and infrastructure. Naturally, cloud providers need to deploy a large number of virtual machines (VMs) to meet the various needs of a broad spectrum of users. This is a typical economy of scale and helps reduce the cost of cloud providers and users. In addition, cloud providers will dynamically adjust the accessible virtual machines based on the workload of their platforms to further save costs. The exposure of cloud computing has conduct many advantages to the distribution of scientific workflows. In specific, the Infrastructure-as-a-Service (IaaS) cloud enables Workflow Management Systems (WMS) to access almost unlimited resource pools that can be received, setup, and applied as needed, and paid for through use[1].

Cloud computing is called a sort of Internet-based computing. In cloud computing, the cloud can be regarded as the Internet. It provides access to assets on the Internet for the entire user, and also provides computing functions on demand. This is a free or fee-based service. Customers use these services according to a "pay-as-you-go" model. Cloud computing objective is to share data, resources and services among its users. An efficient load balancing algorithm can maximize the use of resources. Therefore, the proposed algorithm can maximize the use of every virtual machine created in the data center in the cloud system[2].

Virtual machine (VM) placement algorithms play a vital role in cloud computing. In the traditional cloud computing environment, the general obstacles of virtual machine placement has been considerably studied.Once the volunteer computing resources are discovered and the resource pool of the volunteer cloud is upgraded, next to assign the resources to the workload of the cloud users. This is a two-stage process: VM configuration, assigning virtual machines to cloud application workloads, VM placement, and mapping VMs to volunteer nodes (VN)[3].

Workflow applications are commonly used in distributed computing environments, as in big data analysis applications, which comprise several tasks along with inter-task dependencies4, image processing applications[5–7], Scientific workflow (like CyberShake, LIGO, Epigenomics)[8]. Recently, due to widespread attention of workflow scheduling ,tremendous work has been proposed on workflow scheduling methods. These approaches can be unevenly categorized into meta-heuristic-based, list-based, cluster-based[9]. The HEFT algorithm to find a set of Pareto solutions for workflow scheduling, with the aim of building construction time and energy competence is projected by Durillo et al[10] .A random level scheduling procedure to plan various workflows with random running time presented by Li et al[1]. Nonetheless, shortcomings of all mentioned methods is that, the only emphasis is on scheduling a single workflow, rather than scheduling numerous workflows by timelines in the cloud platform is not competent.

The basic principle of Energy-saving-aware operating model is to state an energy-optimized operating mode and try to increase the number of operating servers in this state. The idle servers will shift to the sleep states for energy saving[12].

Scheduling and load balancing are vital to transfer tasks in the cloud computing for efficient resources sharing from under-utilized VMs . In cloud computing environment, the scheduling of non-preemptible tasks is an unrecoverable restriction, so it must be allocated to the most suitable VM in the initial placement position. Making cloud computing more efficient and improving user satisfaction is done by assigning tasks to applicable resources over static or dynamic scheduling to accomplish contributing heterogeneous resources[13].

At present the cloud achieves load balancing by transferring virtual machines (VMs) from heavier physical machines (PM) to lighter PMs,to provide a commanding infrastructure as a service (IaaS). The most looked-for features of the cloud is to attain thriving and efficient load balancing. Virtual machines in the cloud acquires different physical resources like CPU, bandwidth and memory for serving many facilities as in high-performance computing, web services, due too which it tends to over consumption of resources in diverse ways[14].

For efficiently handling the tremendous dynamic requests, the load essentials to be uniformly distributed amongst cloud nodes. For achieving this objective, numerous load balancing approaches have been presented and studied till date. This algorithm provides better results in terms of waiting time, execution time, turnaround time and throughput, experimental results will help to compare the result with improved throttling algorithms[15].

With the rapid increase of demand in cloud computing services, the workload on a single node will obviously increase and here comes the essential need for load balancing. In proposed theory, the weighted round-robin algorithm and the Honeybee algorithm are combined to attain the goal of lowest data center processing and response time[16].

Cloud delivers countless features because of its massive resources like, resources sharing for various purposes. Cloud computing comprises of various encounters for performance and efficiency. That is why to improve the proficiency of cloud computing environments, virtual machines (VM) have been used for resource planning. In this study, an algorithm is proposed to achieve workload distribution amongst diverse VMs built on precedence. Here, VMs are categorized according to their processing capabilities, and job requests are assigned to VMs according to their instruction numbers and priorities[17].

Unconventional cloud systems which relies on the standby resources are developing, as enhancement for the current cloud based on data centers. These systems are having a communal significance that they don’t rely on devoted data centers to deliver desired services. The implementation and integration of the model in the real volunteer cloud system cuCloud is demonstrated, for pointing at the design of this predictive model for the availability and reliability of nodes in a volunteer cloud computing system based on a multi-state semi-Markov process,[18].

The placement of virtual machines is still an open and challenging problem with several different characteristics ,even though broad study has been conducted on the assignment of virtual machines in cloud computing environments, for volunteer cloud computing. The virtual machine placement issue in voluntary cloud computing is a 0-1 multidimensional knapsack problem, and three heuristic-based algorithms have been developed to meet the specific goals and constrictions of voluntary cloud computing[19].

Volunteer computing is a sort of distributed computing that usages the total backup computing resources of voluntary devices. It offers a cheaper and more environmental friendly alternate computing infrastructure that can

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 5, 2021, Pages. 104-114
Received 15 April 2021; Accepted 05 May 2021.

accompaniment devoted, central and costly data centers. Though, the implication of this is dominated by scientific methods and only comprises a lesser possible volunteer nodes. Security, task allocation, resource management and incentive models are the main technical and operational cons. We need to improve existing technologies and new equipment mechanisms, to fully consume the potential of voluntary computing and prove it a reliable substitute computing infrastructure for general-purpose applications[20].

Objectives

- Implementation of Proposed Optimization algorithm to solve the optimizing issues in cloud environment.
- To explain task load balancing crisis in cloud environment to diminish cost and time.
- Attribute based resource mapping process to enhance system performance.
- To improve the utilization ratio of the resource as well as guarantee application quality of service

METHODS

Here the proposed architecture of system for the workflow planning and scheduling algorithm is presented and studied in brief for multi-tenant cloud platform. There are various notable workflow management systems, like Pyrus, SAP WfMS, Pegasus.

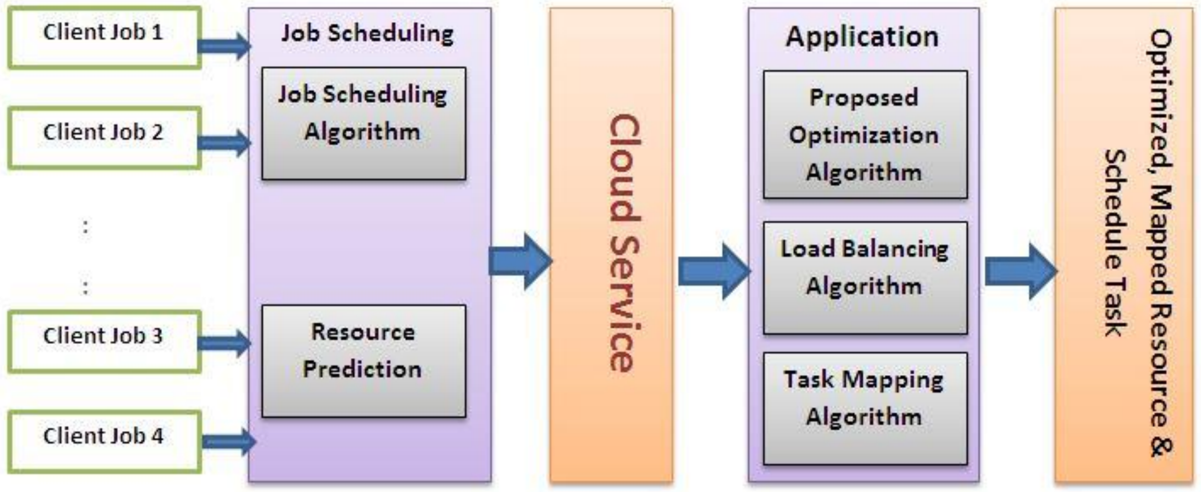


Fig No: 1 Reference model for Workflow Scheduling

Not with standing, a considerable lot of their highlights are advanced for ordinary framework and bunch figuring to execute single/various job(s) or workflow and in this manner will be unable to acquire the greater part of the key parts of distributed computing, while such frameworks experience the ill effects of restricted asset provisioning.

Despite the fact that there are few works tending to work flow booking on mists, e.g., they were not composed with regards to multi-occupancy. Given the rise of assorted arrangements of logical workflow applications each having a place with various areas, a multi-occupant mindful and adaptable workflow stage is expected to cost-adequately execute/convey the workflow uses of numerous tenants. The imagined design empowers such workflow applications to share a solitary foundation while exploiting the versatility and pay-as-you go charging model of distributed computing.

Figure1 delineates a four-layered plan of the stated workflow booking framework. The main layer (occupant) comprises of workflow maker/writer. The succeeding layer (middleware) consists of workflow dispatcher, benefit line, workflow scheduler, shared pool asset data, ten-insect data and execution store, QoS screen, Furthermore, agent. The third layer is the virtual framework layer, and the fourth layer is the physical foundation layer. Each layer is associated with succeeding layer.

Tenant Layer: Every occupant particular cloud workflow is arranged by obtaining the inhabitant inclination and QoS. Each inhabitant represents an individual workflow function to the work flow booking framework. The workflow events are booked by the accessible assets (virtual machines, data center, and so on.) at the given due date. The inhabitants present the work flow applications as per a uniform or arbitrary distribution. The workflow scheduler checks the accessibility of administrations and assets as per the clients QoS pre-conditions and then applies the given planning approach to implement planned workflow assignments.

Middleware Layer: With a fixed agenda to acknowledge multi-occupancy, an proper middleware is required to limit the fundamental many-sided quality as arrange, oversee, and recognize various occupants and in addition inhabitant particular customization of workflow applications. Actually, it separates the inhabitant and framework layers. Middleware layer consists of various segments, workflow scheduler, QoS observing part, and execution store segment.

1. Workflow Dispatcher: Its work is to calculate the workload of all the applications and send out to the administration line.
2. Service Queue: The administration line keeps up a necessary line intended for all approaching workflow responsibility and conveys them to the workflow scheduler.
   In the last layer we are applying our algorithms like Proposed optimization algorithm, Proposed Load balancing algorithm and Proposed task mapping algorithm.

RESULTS:

In this proposed work comparing various algorithms related to resource optimization ,Load balancing and Resource mapping algorithms. The existing optimization algorithms which are Particle swarm Optimization(PSO) and Improved Particle Swarm Optimization(IPSO)[21].The proposed Optimization algorithm is explained below contains better results as compare to the existing algorithms like PSO and IPSO.In the next step the proposed work focuses on Load balancing algorithms like Honey Bee and Dynamic Load Balancing[22] .The Proposed Load Balancing algorithm which is explained below has better results as compare to existing Honey Bee and Dynamic LB algorithms. Lastly here applying Resource Mapping algorithm. The existing resource mapping algorithms like Utilization-Reliability Aware Scheduling Algorithm(URASA) and Unobtrusive Utilization-Reliability Aware Scheduling Algorithm (UURASA)[3] are compared with proposed resource mapping algorithm.The Proposed Resource Mapping Algorithm is discussed below.

Proposed Optimization Algorithm:

```
1. Clustering ()
2. CheckLoad()
 3. After the job arrives, the scheduler will query the VM queue.
 4. If (VM queue)< >ø
 The scheduler removes 1stvm from the VM queue and directs the job to the VM.
 5. If (VM queue)=ø{
 If (GResourcesCan Host vm), then // GlobalResource
{Start a new VM instance
Add a virtual machine to the list of available virtual machines
 Deploy the service on the new vm}
 different {
 The scheduler uses Proposed() to direct the job to a randomly selected cluster
 LoadCheck()}
 6. If (OVMC) {
 Within Pm()
 Find the overloaded VM instance, and then distribute the load to the less loaded VM in the PM.
 different {
 International Standard Time ()
Move tasks to Pm where the average usage rate is below the threshold in the peer cluster.
```

Proposed Load balancing Algorithm

```
Input: VM number, task number, task size
Output: All tasks are allocated according to the division of labor
Step 1: Check the number of virtual machines
Step 2: Get the number and size of tasks.
```

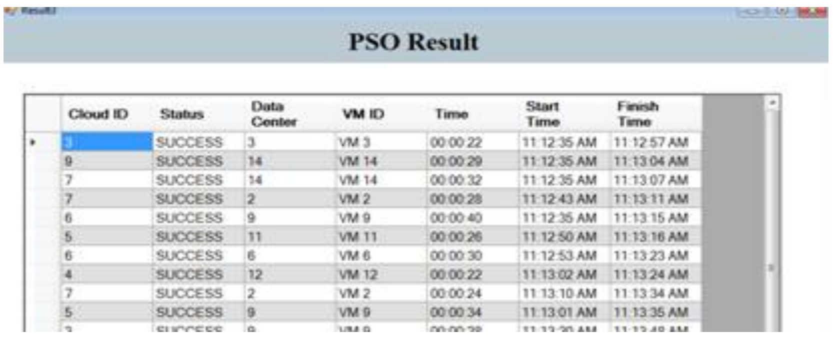
Step 3 Check the current load status in the VM
Step 4 Sort the load from the VM into the loaded computers in ascending order
Step 5: Assign the new task to be executed to the low-load virtual machine.
Step 6: Clear all assigned tasks from the list
Step 7: Calculate the average burst time value
Step 8 Assign tasks to the virtual machine with the least load
Step 9: Continue until the waiting list is free

Proposed Resource Mapping Algorithm

```
Input: VM_Req_Set
Primary_Pool←∅
Secondary_Pool_Queue←∅
Output: Assign VM_Req to VN1, VN2,..., VNn
Add VN to the main pool with Relibilityscore ≥ α;
for (VN = 0; toVN length in Primary_Pool)
Calculation using vector;
end
for (VM = 0; to the length of VM in VM_Request_Set)
For (VN = 0 to VN length in Primary_Pool)
VM_Request_Vec←Calculate the VM req vector;
Secondary_Pool_Queue.enqueue (VN);
end
Calculate the penalty vector (Penalty_Vec);
Obtain all hosts that have hosted virtual machines;
Remove Hosts_with_VMs from Secondary_Pool_Queue;
ifSecondary_Pool_Queue <>∅then
Secondary_Pool_Queue.dequeue(); //The smallest VN
//Excellent value
Is VM suitable for VN
Allocate VM <-VN;
updateUtil_Vec
rest;
different
Otherwise go outside
end
different
//VN with the lowest fine
Select minPenalty_Vec;
Then whether the VM can be put into the VN
Assign VM to VN;
updateUtil_Vec
rest;
different
The virtual machine cannot be allocated;
return;
end
end
end
```

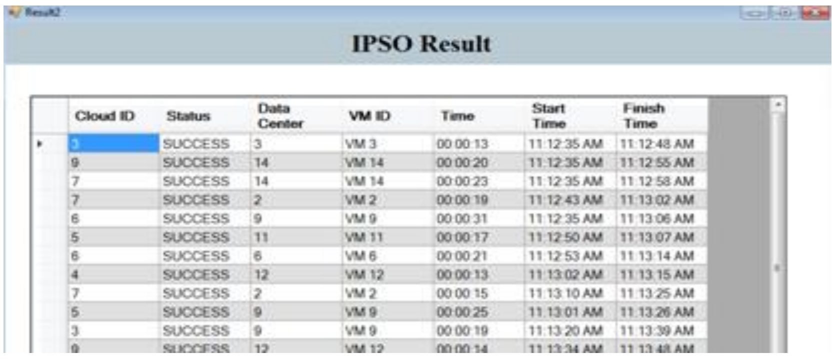
Result

The results are implemented using Java Net Beans tool.Here used cloud sim frame work.Firstly in Fig.2 Shows the implementation of Particle Swarm Optimization (PSO). Nextin Fig.3 shows the implementation of Improved Particle Swarm Optimization (IPSO). The efficiency of resources utilized in the Proposed Optimization which is shown in Fig.4 is better than existing PSO and IPSO. Here the benefit for the cloud service providers that is Cloud ID is mentioned. Various Data Centers and Virtual Machines are considered. The time taken to execute number of tasks which is make span time is minimized in the Proposed Optimization algorithm.


PSO Result

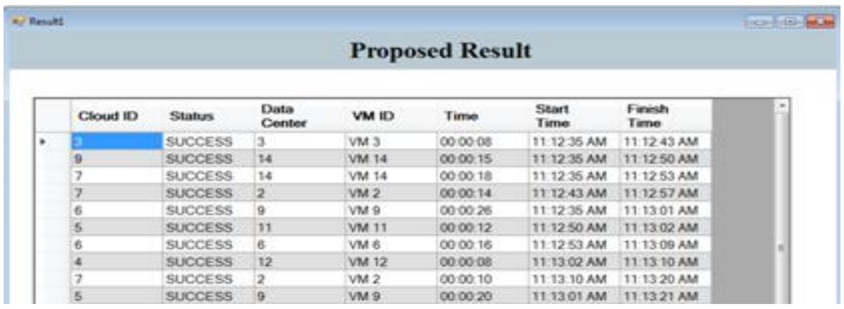
Cloud ID	Status	Data Center	VM ID	Time	Start Time	Finish Time
3	SUCCESS	3	VM 3	00:00:22	11:12:35 AM	11:12:57 AM
9	SUCCESS	14	VM 14	00:00:29	11:12:35 AM	11:13:04 AM
7	SUCCESS	14	VM 14	00:00:32	11:12:35 AM	11:13:07 AM
7	SUCCESS	2	VM 2	00:00:28	11:12:43 AM	11:13:11 AM
6	SUCCESS	9	VM 9	00:00:40	11:12:35 AM	11:13:15 AM
5	SUCCESS	11	VM 11	00:00:26	11:12:50 AM	11:13:16 AM
6	SUCCESS	6	VM 6	00:00:30	11:12:53 AM	11:13:23 AM
4	SUCCESS	12	VM 12	00:00:22	11:13:02 AM	11:13:24 AM
7	SUCCESS	2	VM 2	00:00:24	11:13:10 AM	11:13:34 AM
5	SUCCESS	9	VM 9	00:00:34	11:13:01 AM	11:13:35 AM
3	SUCCESS	9	VM 9	00:00:28	11:13:20 AM	11:13:48 AM

Fig No:2 Cloudlet execution using Particle Swarm Optimization algorithm (PSO)


IPSO Result

Cloud ID	Status	Data Center	VM ID	Time	Start Time	Finish Time
3	SUCCESS	3	VM 3	00:00:13	11:12:35 AM	11:12:48 AM
9	SUCCESS	14	VM 14	00:00:20	11:12:35 AM	11:12:55 AM
7	SUCCESS	14	VM 14	00:00:23	11:12:35 AM	11:12:58 AM
7	SUCCESS	2	VM 2	00:00:19	11:12:43 AM	11:13:02 AM
6	SUCCESS	9	VM 9	00:00:31	11:12:35 AM	11:13:06 AM
5	SUCCESS	11	VM 11	00:00:17	11:12:50 AM	11:13:07 AM
6	SUCCESS	6	VM 6	00:00:21	11:12:53 AM	11:13:14 AM
4	SUCCESS	12	VM 12	00:00:13	11:13:02 AM	11:13:15 AM
7	SUCCESS	2	VM 2	00:00:15	11:13:10 AM	11:13:25 AM
5	SUCCESS	9	VM 9	00:00:25	11:13:01 AM	11:13:26 AM
3	SUCCESS	9	VM 9	00:00:19	11:13:20 AM	11:13:39 AM
9	SUCCESS	12	VM 12	00:00:14	11:13:34 AM	11:13:48 AM

Fig No: 3 Cloudlet execution using Improved Particle Swarm Optimization algorithm (IPSO)


Proposed Result

Cloud ID	Status	Data Center	VM ID	Time	Start Time	Finish Time
3	SUCCESS	3	VM 3	00:00:08	11:12:35 AM	11:12:43 AM
9	SUCCESS	14	VM 14	00:00:15	11:12:35 AM	11:12:50 AM
7	SUCCESS	14	VM 14	00:00:18	11:12:35 AM	11:12:53 AM
7	SUCCESS	2	VM 2	00:00:14	11:12:43 AM	11:12:57 AM
6	SUCCESS	9	VM 9	00:00:26	11:12:35 AM	11:13:01 AM
5	SUCCESS	11	VM 11	00:00:12	11:12:50 AM	11:13:02 AM
6	SUCCESS	6	VM 6	00:00:16	11:12:53 AM	11:13:09 AM
4	SUCCESS	12	VM 12	00:00:08	11:13:02 AM	11:13:10 AM
7	SUCCESS	2	VM 2	00:00:10	11:13:10 AM	11:13:20 AM
5	SUCCESS	9	VM 9	00:00:20	11:13:01 AM	11:13:21 AM

Fig.4 Cloudlet execution using Proposed Optimization algorithm .

In the above figures the proposed optimization algorithm's execution of various tasks is minimized as compare to existing algorithms PSO and IPSO.

The results are checked by increasing number of tasks executed. The following fig.5 shows the graph of comparison between resource utilization versus number of tasks. As per the result it shows the proposed optimization algorithm results of resource utilization is minimized by increasing number of tasks. In the Table1 shows Resource Utilization Values compared to PSO,IPSO and Proposed optimization algorithms by increasing number of tasks.

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 5, 2021, Pages. 104-114
Received 15 April 2021; Accepted 05 May 2021.

Table No:1:Resource Utilization Values compared to PSO,IPSO and Proposed optimization algorithms by increasing number of tasks

Number of tasks	PSO	IPSO	Proposed Optimization
5	0.83	0.72	0.68
10	0.84	0.72	0.67
15	0.85	0.73	0.66
20	0.86	0.75	0.65
25	0.9	0.76	0.65
30	0.9	0.79	0.65

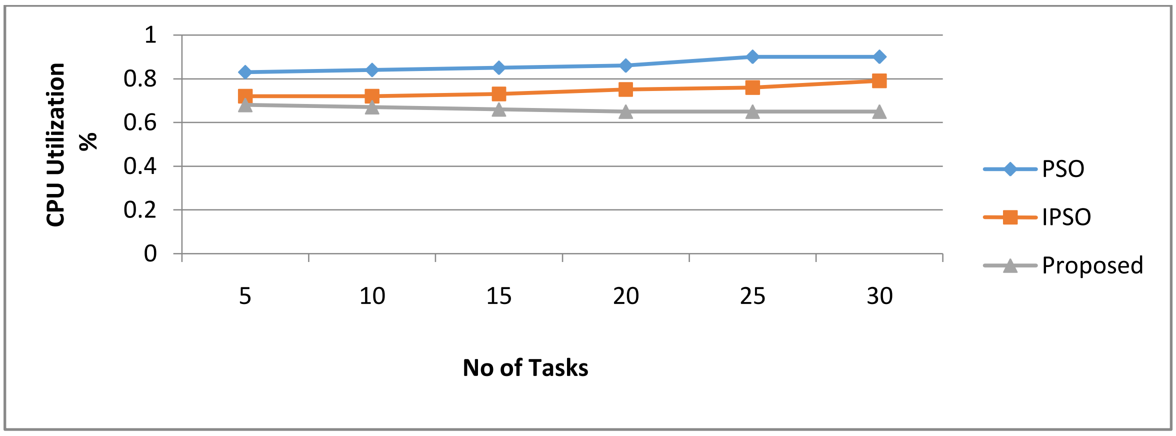


Fig No:5 Resource Utilization Versus Number of Tasks experimentation results.

ExecutionTime:

The following fig.6 shows the time utilization values in Secs in ratio to the number of Virtual Machines. In Table 2 shows the values of time in Secs with respect to number of Virtual Machines. From these experimental results it can be concluded that the proposed optimization algorithm's execution time is less as compared to existing PSO and IPSO algorithms.

Table No: 2 Time Utilization values in secs with respect to number of Virtual Machines for PSO,IPSO and Proposed Optimization Algorithms.

Number of Tasks	PSO	IPSO	Proposed Optimization
25	108	81	73
30	101	78	71
35	99	75	66
40	96	72	61
45	91	69	56
50	89	65	50

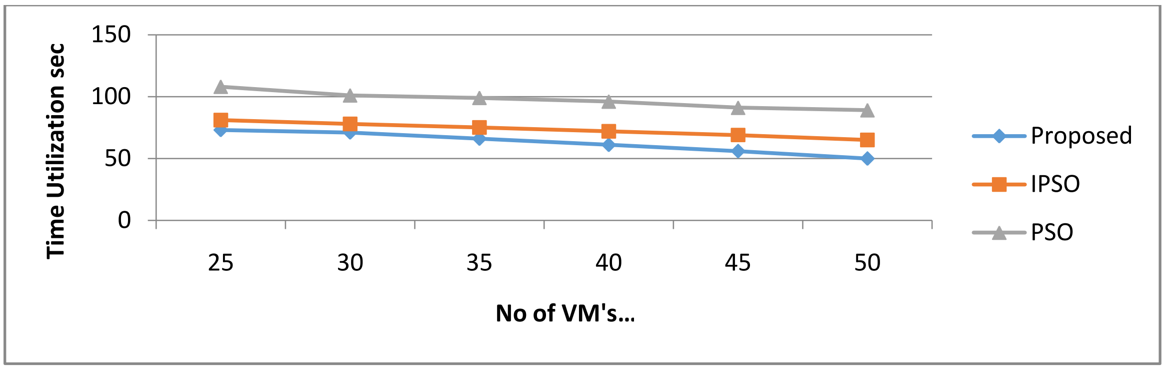
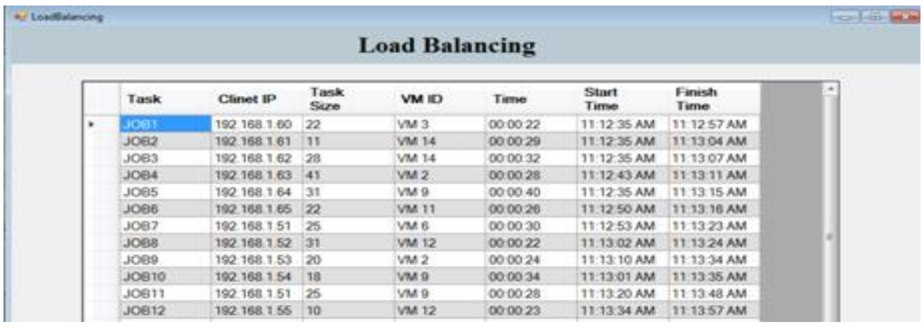


Fig no: 6 Number Of Virtual Machines versus Time Utilization in Secs for PSO,IPSO and Proposed Optimization Algorithm.

Considering the objective 2 the experimental results shows that the allocation of task size to the corresponding Virtual machine by adjusting the load. The time taken to execute such task is calculated and load balancing done.The fig.7 Shows the cloudlet execution by considering the proposed optimization algorithm.



Load Balancing

Task	Clinet IP	Task Size	VM ID	Time	Start Time	Finish Time
JOB1	192.168.1.60	22	VM 3	00:00:22	11:12:35 AM	11:12:57 AM
JOB2	192.168.1.61	11	VM 14	00:00:29	11:12:35 AM	11:13:04 AM
JOB3	192.168.1.62	28	VM 14	00:00:32	11:12:35 AM	11:13:07 AM
JOB4	192.168.1.63	41	VM 2	00:00:28	11:12:43 AM	11:13:11 AM
JOB5	192.168.1.64	31	VM 9	00:00:40	11:12:35 AM	11:13:15 AM
JOB6	192.168.1.65	22	VM 11	00:00:26	11:12:50 AM	11:13:16 AM
JOB7	192.168.1.51	25	VM 6	00:00:30	11:12:53 AM	11:13:23 AM
JOB8	192.168.1.52	31	VM 12	00:00:22	11:13:02 AM	11:13:24 AM
JOB9	192.168.1.53	20	VM 2	00:00:24	11:13:10 AM	11:13:34 AM
JOB10	192.168.1.54	18	VM 9	00:00:34	11:13:01 AM	11:13:35 AM
JOB11	192.168.1.51	25	VM 9	00:00:28	11:13:20 AM	11:13:48 AM
JOB12	192.168.1.55	10	VM 12	00:00:23	11:13:34 AM	11:13:57 AM

Fig no: 7 Cloudlet execution by considering proposed load balancing algorithm.

The following table 3 shows the CPU Processing and Completion Time in milli Secs with respect to Honey Bee ,Dynamic Load Balancing and Proposed Load Balancing algorithm. Fig.8 shows graph comparison of CPU Processing and Completion time between Honey Bee and Dynamic Load Balancing with respect to Proposed Load Balancing Algorithm.From this it can be concluded that Proposed Load Balancing takes less time as compare to existing Honey Bee and Dynamic LB algorithms.

Table no: 3 CPU Processing and Completion Time in milli Secs with respect to Honey Bee,Dynamic LB and Proposed Load Balancing algorithms.

	Honey Bee	Dynamic LB	Proposed LB
CPU Processing Time (milli sec)	280	140	120
Completion Time(milli sec)	270	180	160

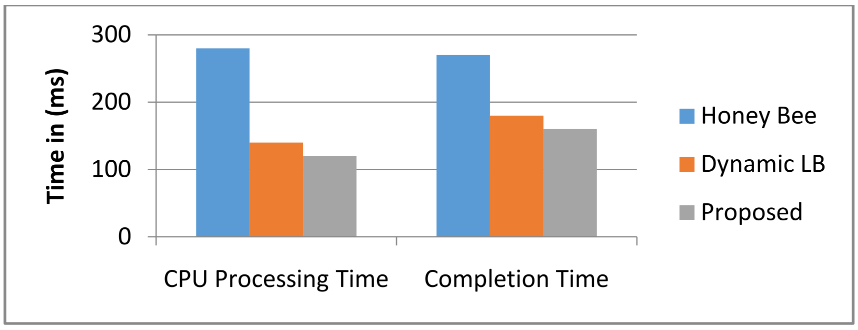
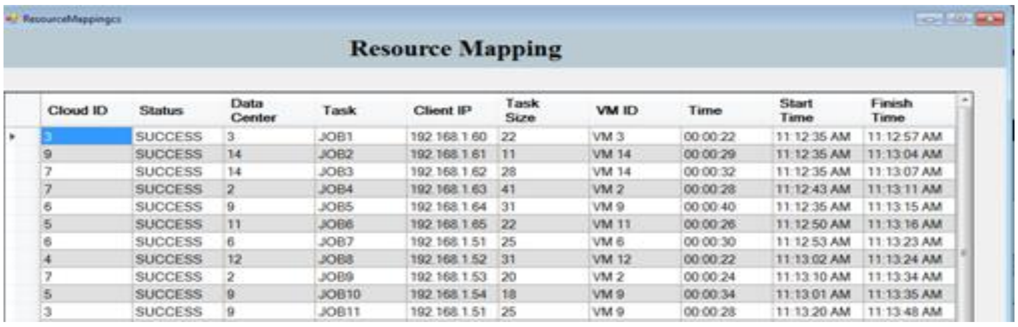


Fig No: 8 shows graph comparison of CPU Processing and Completion time between Honey Bee and Dynamic Load Balancing with respect to Proposed Load Balancing Algorithm.

As of Objective 3 Fig.9 shows Cloudlet execution by considering proposed Resource Mapping Algorithm. Here tasks are executed by considering Cloud let ID ,Client IP,task size and allocation of Virtual Node(VN) is done.There will be many virtual machines exists but for simulation purpose only 4 Virtual machines or Virtual nodes are considered.



Resource Mapping

Cloud ID	Status	Data Center	Task	Client IP	Task Size	VM ID	Time	Start Time	Finish Time
3	SUCCESS	3	JOB1	192.168.1.60	22	VM 3	00:00:22	11:12:35 AM	11:12:57 AM
9	SUCCESS	14	JOB2	192.168.1.61	11	VM 14	00:00:29	11:12:35 AM	11:13:04 AM
7	SUCCESS	14	JOB3	192.168.1.62	28	VM 14	00:00:32	11:12:35 AM	11:13:07 AM
7	SUCCESS	2	JOB4	192.168.1.63	41	VM 2	00:00:28	11:12:43 AM	11:13:11 AM
6	SUCCESS	9	JOB5	192.168.1.64	31	VM 9	00:00:40	11:12:35 AM	11:13:15 AM
5	SUCCESS	11	JOB6	192.168.1.65	22	VM 11	00:00:26	11:12:50 AM	11:13:16 AM
6	SUCCESS	6	JOB7	192.168.1.51	25	VM 6	00:00:30	11:12:53 AM	11:13:23 AM
4	SUCCESS	12	JOB8	192.168.1.52	31	VM 12	00:00:22	11:13:02 AM	11:13:24 AM
7	SUCCESS	2	JOB9	192.168.1.53	20	VM 2	00:00:24	11:13:10 AM	11:13:34 AM
5	SUCCESS	9	JOB10	192.168.1.54	18	VM 9	00:00:34	11:13:01 AM	11:13:35 AM
3	SUCCESS	9	JOB11	192.168.1.51	25	VM 9	00:00:28	11:13:20 AM	11:13:48 AM

Fig no: 9 Cloudlet execution by considering proposed Resource Mapping Algorithm.

Fig.10 Shows CPU Utilization versus number of Virtual nodes .Table 4 shows CPU Utilization Values with respect to Number of Virtual Nodes(VN) for URASA,UURASA and Proposed resource mapping algorithm.With these experimental results it can be concluded that the resource utilization of proposed resource mapping algorithm is minimized as per the existing algorithms URASA and UURASA.

Table no: 4 CPU Utilization Values with respect to Number of Virtual Nodes(VN) for URASA,UURASA and Proposed resource mapping algorithm

Virtual Nodes	URASA	UURASA	Proposed Resource Mapping
VN1	0.18	0.15	0.1
VN2	0.2	0.3	0.16
VN3	0.3	0.2	0.18
VN4	0.32	0.28	0.2

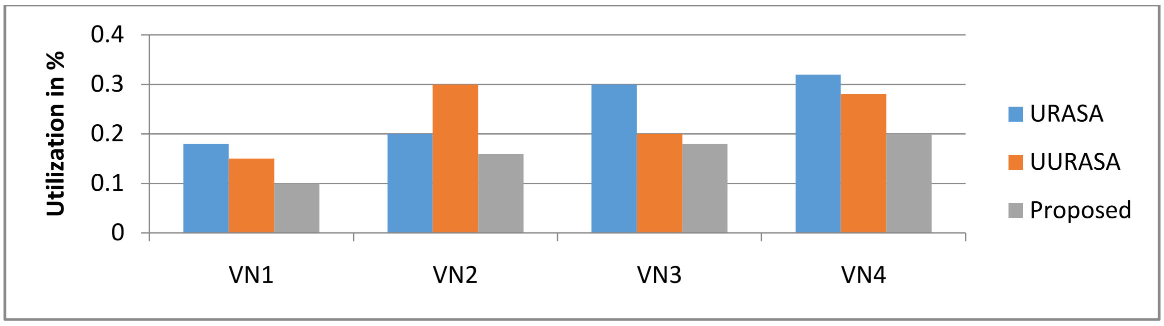


Fig No:10 CPU Utilization versus number of Virtual nodes.

CONCLUSION

This paper explains new approach for resource optimization and task scheduling advancement and applied Load Balancing and Resource mapping. The experimental results shows that the proposed algorithm are helpful to utilize resources efficiently and adjust the load in proper manner. It also enhances results with respect to resource mapping.

ACKNOWLEDGEMENT

It is my sincere gratitude towards my Project Guide Dr.A.Jaya Lakshmi for her significant proposals and direction throughout my course of study and auspicious assistance given to me in completion of research work. I happily accept this open door to express gratitude toward Dr. V .Krishna Reddy, for important direction of research work. I am appreciative to every one of the individuals who helped us straightforwardly or in a roundabout way for my examination work.

CONFLICTS OF INTEREST:

The author have declared no conflicts of interest

REFERENCES

[1] A.Prasanth, S.Jayachitra, 'A Novel Multi-Objective Optimization Strategy for Enhancing Quality of Service in IoT enabled WSN Applications', Peer-to-Peer Networking and Applications, Vol.13, 2020, pp.1905–1920.

[2] Shikha Garg, Rakesh Kumar Dwivedi and Himanshu Chauhan," Efficient Utilizationof Virtual Machines in CloudComputing using Synchronized Throttled LoadBalancing" , 1st IEEE International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India,September 2015.

[3] Tessema M. Mengistu, Dunren Che, Shiyong Lu," Multi-Objective Resource Mapping and Allocation forVolunteer Cloud Computing", IEEE 12th International Conference on Cloud Computing (CLOUD),2019.

[4] Boutin E, Ekanayake J, Lin W, Shi B, Zhou J, Qian Z, Wu M, Zhou L (2014) Apollo: scalable and coordinated scheduling for cloud-scale computing. In:Proceedings of the 11th USENIX conference onoperating systems design and implementation. USENIX Association, pp 285–300.

[5] Meneguzzo DM, Liknes GC, Nelson MD (2013) Mapping trees outside forests using high-resolutionaerialimagery:acomparisonofpixel-and object based classification approaches.Environ Monit Assess185(8):6261–6275.

[6] S.Jayachitra, A.Prasanth, 'Multi-Feature Analysis for Automated Brain Stroke Classification Using Weighted Gaussian Naïve Baye’s Classifier', Journal of Circuits, Systems, and Computers, 2021

[7] Abduljabbar ZA, Jin H, Ibrahim A, Hussien ZA, Hussain MA, Abbdal SH, Zou D (2016) Sepim:secure and efficient private image matching. Appl Sci 6(8):213.

[8] Juve G, Chervenak A, Deelman E, Bharathi S, Mehta G, Vahi K (2013) Characterizing and profiling scientific workflows. Future Gener Comput Syst 29(3):682–692.

[9] A.Prasanth, S.Pavalarajan, 'Implementation of Efficient Intra- and Inter-Zone Routing for Extending Network Consistency in Wireless Sensor Networks', Journal of Circuits, Systems, and Computers, Vol.29, 2020.

[10] Durillo JJ, Nae V, Prodan R (2014) Multi-objective energy-efficient workflow scheduling using list-based heuristics. Future Gener Comput Syst 36:221–236.

[11] Li K, Tang X, Veeravalli B, Li K (2015) Scheduling precedence constrained stochastic tasks on het-erogeneous cluster systems. IEEE Trans Comput 64(1):191–204.

[12]Ashkan Paya and Dan C. Marinescu," Energy-aware Load Balancing and Application Scaling for the Cloud Ecosystem", IEEE Transactions on Cloud Computing,Volume: 5, Issue: 1,March 2017.

[13]D. Chitra Devi and V. Rhymend Uthariaraj,"Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks",The Scientific World Journal,Volume 2016,Feb 2016.

[14] Liuhua Chen, Haiying Shen, Karan Sapra,"RIAL: Resource Intensity Aware Load Balancing in Clouds",IEEE INFOCOM-IEEE Conference on Computer Communications 2014.

[15]Surbhi Kapoor and Dr. Chetna Dabas, "Cluster Based Load Balancing in Cloud Computing",Eighth International Conference on Contemporary Computing (IC3),IEEE Aug 2015.

[16] Nithin K. C. Das,Melvin S. George and P. Jaya,"Incorporating weighted round robin in honeybee algorithm for enhanced load balancing in cloud environment" ,International Conference on Communication and Signal Processing (ICCSP),IEEE,April 2017.

[17]Farzana Sadia, Nusrat Jahan, Lamisha Rawshan, Madina Tul Jeba and Dr. Touhid Bhuiyan,"A Priority Based Dynamic Resource Mapping Algorithm For Load Balancing In Cloud", 4th International Conference on Advances in Electrical Engineering (ICAEE),IEEE,Sep 2017.

[18]Usha Kiruthika,Thamarai Selvi Somasundaram, S. Kanaga Suba Raja, 'Lifecycle Model of a Negotiation Agent: A Survey of Automated Negotiation Techniques', Group Decision and Negotiation, ISSN 0926-2644, Volume 29, Issue - 6, 2020

[19] Tessema Mengistu, Dunren Che and Shiyong Lu,"Multi-Objective Resource Mapping and Allocation forVolunteer Cloud Computing", IEEE 12th International Conference on Cloud Computing (CLOUD),July 2019.

[20]Tessema M. Mengistu and Dunren Che,"Survey and Taxonomy of Volunteer Computing",ACM Computing Surveys, July 2019.

[21]Naresh T,Dr.A.Jaya Lakshmi and Dr.V.Krishna Reddy,"Implementation of Resource Optimization approaches for Workflow Scheduling ",Jour of Adv Research in Dynamical & Control Systems, Vol. 10, 13-Special Issue, 2018.

[22]Karan D. Patel and Tosal M.Bhalodia," An Efficient Dynamic Load Balancing Algorithm for Virtual Machine in Cloud Computing ", Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2019)IEEE Xplore Part Number: CFP19K34-ART; ISBN: 978-1-5386-8113-8.

[23]Murugan, S., Jeyalaksshmi, S., Mahalakshmi, B., Suseendran, G., Jabeen, T. N., & Manikandan, R. (2020). Comparison of ACO and PSO algorithm using energy consumption and load balancing in emerging MANET and VANET infrastructure. Journal of Critical Reviews, 7(9), 2020.