ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Comparision of a New Approach of Balancing the Load in Cloud Environment with the Existing Techniques

Ashima Narang, Dr. Vijay Laxmi
Computer Science and Engineering, Guru Kashi University, India;
Computer Science and Engineering,Guru Kashi University, India

Abstract

In cloud Computing, Load Balancing is essential for efficient operations in distributed environments. As Cloud Computing is growing rapidly and clients are demanding more services and better results, load balancing for the Cloud has become a very interesting and important research area. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. Our objective is to implement an effective load balancing algorithm for balancing the load on cloud. In this thesis, we have studied and implemented three algorithm using Java Programming and simulate the algorithms on CloudSim. The Main contribution of CloudSim is to provide a holistic software framework for modeling Cloud computing environments and performance testing application services.

And, I have proved that the proposed algorithm Active Clustering gives the better results than RASA and ACO. I analyzed the results on the basis of different parameters (Resource Arrival Time, Processing time of Tasks, Tasks Arrival Time, Utilizations of Tasks, and Response Time).

Keywords: Load Balancing, CloudSim, Utilization of Tasks, Response Time.

Introduction

Cloud computing is a new technology and it is becoming popular because of its great features. In this technology almost everything like hardware, software and platform are provided as a service. A cloud provider provides services on the basis of client's requests. An important issue in cloud is, scheduling of users requests, means how to allocate resources to these requests, so that the requested tasks can be completed in a minimum time and the cost incurred in the task should also be minimum. In case of Cloud computing services can be used from diverse and wide spread resources, rather than remote servers or local machines. There is no standard definition of Cloud computing. Generally it consists of a bunch of distributed servers known as masters, providing demanded services and resources to different clients known as clients in a network with scalability and reliability of datacenter. The distributed computers provide on-demand services. Services may be of software resources (e.g. Software as a Service, SaaS) or physical resources (e.g. Platform as a Service, PaaS) or hardware/infrastructure (e.g. Hardware as a Service, HaaS or Infrastructure as a Service, IaaS). AmazonEC2 (Amazon Elastic Compute Cloud) is an example of cloud computing services [2].

Cloud computing has recently become popular due to the maturity of related technologies such as network devices, software applications and hardware capacities. Resources in these systems can be widely distributed and the scale of resources involved can range from several servers to an entire data center. To integrate and make good use of resources at various scales, cloud computing needs efficient methods to manage them [4]. Consequently, the focus of much research in recent years has been on how to utilize resources and how to reduce power consumption. One of the key technologies in cloud computing is virtualization. The ability to create virtual machines (VMs) [14] dynamically on demand is a popular solution for managing resources on physical machines. Therefore, many methods [17,18] have been developed that enhance resource utilization such as memory compression, request discrimination, defining threshold for resource usage and task allocation among VMs. Improvements in power consumption, and the relationship between resource usage and energy consumption has also been widely studied [6,10]. Some research aims to improve

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

resource utilization while others aim to reduce energy consumption. The goals of both are to reduce costs for data centers. Due to the large size of many data centers, the financial savings are substantial.

Load balancing on cloud computing

With the increasing popularity of cloud computing, the amount of processing that is being done in the clouds is surging drastically. A cloud is constituted by various nodes which perform computation according to the requests of the clients. As the requests of the clients can be random to the nodes they can vary in quantity and thus the load on each node can also vary. Therefore, every node in a cloud can be unevenly loaded of tasks according to the amount of work requested by the clients. This phenomenon can drastically reduce the working efficiency of the cloud as some nodes which are overloaded will have a higher task completion time compared to the corresponding time taken on an under loaded node in the same cloud. This problem is not only confined only to cloud but is related with every large network like a grid, etc.

Load balancing in large distributed server systems is a complex optimization problem of critical importance in cloud systems and data centers. Load balancing algorithms are classified as static and dynamic algorithms. Static algorithms are mostly suitable for homogeneous and stable environments and can produce very good results in these environments. However, they are usually not flexible and cannot match the dynamic changes to the attributes during the execution time. Dynamic algorithms are more flexible and take into consideration different types of attributes in the system both prior to and during run-time [2]. These algorithms can adapt to changes and provide better results in heterogeneous and dynamic environments. However, as the distribution attributes become more complex and dynamic. As a result some of these algorithms could become inefficient and cause more overhead than necessary resulting in an overall degradation of the services performance.

Types of Load balancing algorithms

Depending on who initiated the process, load balancing algorithms can be of three categories as given in [4]:

- Sender Initiated: If the load balancing algorithm is initialized by the sender.
- Receiver Initiated: If the load balancing algorithm is initiated by the receive
- Symmetric: It is the combination of both sender initiated and receiver initiated

Depending on the current state of the system, load balancing algorithms can be divided into 2 categories as given in [4]:

- Static: It doesn't depend on the current state of the system. Prior knowledge of the system is needed.
- Dynamic: Decisions on load balancing are based on current state of the system. No prior knowledge is needed. So it is better than static approach.

Here we will discuss on various dynamic load balancing algorithms for the clouds of different sizes.

Problem formulation and objectives

In the past, a number of load balancing algorithms have been developed specifically to suit the dynamic cloud computing environments such as INS (Index Name Server) algorithm, WLC (Weighted Least Connection) algorithm, LBMM (Load Balancing Min-Min) algorithm, ACO(Ant Colony Optimization) algorithm and Bee-MMT(Artificial Bee Colony algorithm- Minimal Migration time). We are going to use the Active Clustering and Resource Aware Scheduling Algorithm(RASA) for load balancing in dynamic cloud environments and compare it with Ant Colony Optimization (ACO) algorithm. Performance of Active Clustering and RASA has also been approved better in distributed system than ACO. In this research work, we proposed an algorithm for load distribution of workloads on a cloud by the use of Active Clustering and RASA.

Metrics for Load Balancing In Clouds

Various metrics will be considered in load balancing techniques in cloud computing are discussed below

1. **Throughput** is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system.
2. **Overhead Associated** determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and inter-process communication. This should be minimized so that a load balancing technique can work efficiently.
3. **Fault Tolerance** is the ability of an algorithm to perform uniform load balancing in spite of arbitrary node or link failure. The load balancing should be a good fault-tolerant technique.
4. **Response Time** is the amount of time taken to respond by a particular load balancing

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

algorithm in a distributed system. This parameter should be minimized.

5. **Resource Utilization** is used to check the utilization of re-sources. It should be optimized for an efficient load balancing.
6. **Scalability** is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.
7. **Performance** is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays.

Objectives for this research work are:

1. To optimize the performance of cloud architecture.
2.To implement Active Clustering, Resource Aware Scheduling Algorithm(RASA) and Ant Colony Optimization(ACO) algorithm on local environment using Java Programming and simulate on cloud computing environment using cloudsim toolkit.
3.To analyse the behavior of the load balancing algorithms using following parameters:
(a) Execution Time
(b) Response Time
(c) Resource Utilization
(d) Fault Tolerance
(e) Scalability
(f) Throughput
4.To evaluate the performance using Active Clustering,RASA algorithm and compare them with ACO performance evaluation.

Research methodology

In cloud computing, load balancing is required to distribute the dynamic local workload evenly across all the nodes. It helps to achieve a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource. Proper load balancing aids in minimizing resource consumption, implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning etc.

Input: - Required parameters for cloudlets and VM's are taken from user.

Output: - Improves load balancing at cloud with better response time, data processing time and throughput.

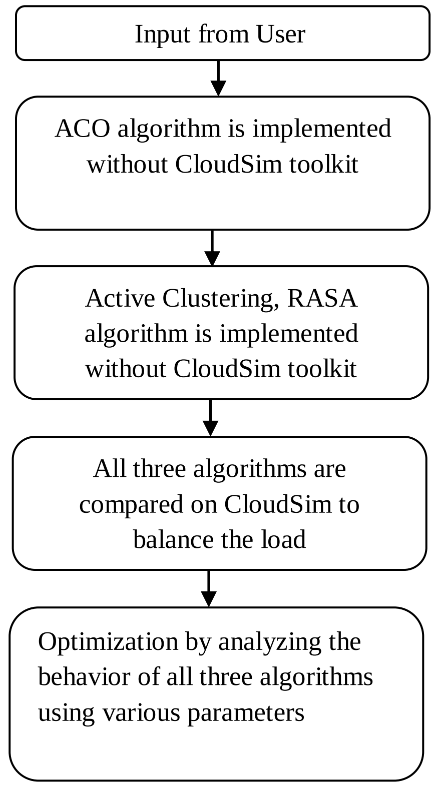

Figure1. Shows the Methodology to be followed

In this thesis, we are implementing ACO, Active Clustering and RASA using java and simulate on CloudSim for balancing the load on cloud.

ACO is applied from the perspective of cloud or grid network systems with the main aim of load balancing of nodes. The approach aims at efficiently distribution of the load among the nodes and such that the ants never encounter a dead end for movements to nodes for building an optimum solution set.

The purpose of **Active Clustering is to** investigate a self-aggregation load balancing technique that is a self-aggregation algorithm to optimize job assignments by connecting similar services using local re-wiring. The performance of the system is enhanced with high resources thereby increasing the throughput by using these resources effectively. It is degraded with an increase in system diversity.

And an RASA offers an improved task scheduling algorithm based on Max-min to resolve the load balancing problems with both Max-min and Min-min. The basic idea of an improved version of Max-min assign task with maximum execution time to resource produces minimum complete time rather than original Max-min assign task with maximum

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

completion time to resource with minimum execution time.

Results and discussions

Ant Colony Optimization (ACO) is inspired from the ant colonies that work together in foraging behavior. In fact the real ants have inspired many researchers for their work [6,8], and the ants approach has been used by many researchers for problem solving in various areas. This approach is called on the name of its inspiration ACO. The ants work together in search of new sources of food and simultaneously use the existing food sources to shift the food back to the nest. We also proposed the movement of ants in two ways, which are as follows:

1. Forward movement-The ants continuously move in the forward direction in the cloud encountering overloaded node or under loaded node.
2. Backward movement-If an ant encounters an overloaded node in its movement when it has previously encountered an under loaded node then it will go backward to the under loaded node to check if the node is still under loaded or not and if it finds it still under loaded then it will redistribute the work to the under loaded node. The vice-versa is also feasible and possible.

In this research, the ACO is used for load balancing. The approach aims at efficiently distribution of the load among the nodes and such that the ants never encounter a dead end for movements to nodes for building an optimum solution set.

Resource Aware Scheduling Algorithm

An RASA offers an improved task scheduling algorithm based on Max-min to resolve the load balancing problems with both Max-min and Min-min. The basic idea of an improved version of Max-min assign task with maximum execution time to resource produces minimum complete time rather than original Max-min assign task with maximum completion time to resource with minimum execution time.

Active Clustering works on the principle of grouping similar nodes together and working on these groups. The process involved is:

- A node initiates the process and selects another node called the matchmaker node from its neighbors satisfying the criteria that it should be of a different type than the former one.
- The so called matchmaker node then forms a connection between neighbors of it which is of the same type as the initial node.
- The matchmaker node then detaches the connection between itself and the initial node.

The purpose of **Active Clustering** is to investigate a self-aggregation load balancing technique that is a self-aggregation algorithm to optimize job assignments by connecting similar services using local re-wiring. The performance of the system is enhanced with high resources thereby increasing the throughput by using these resources effectively. It is degraded with an increase in system diversity.

Simulations Results

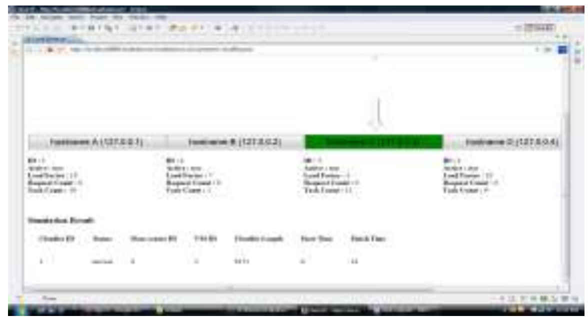

Figure 2. Shows the result of local server startup process for load balancer

Figure 3. Shows the console output screen for ACO algorithm on CloudSim

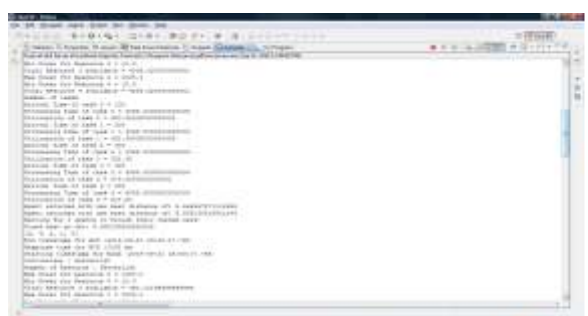

Figure 4 Shows the Task completion of ACO and startup of RASA

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Figure 5. Shows the Completion process of RASA and Startup of Active Clustering

Figure 6. Shows the Final output of Active Clustering Algorithm on CloudSim

Output Tables

Table 1 Shows the output parameters of ACO Algorithm

Number of Resources			Tasks				Final Response Time of ACO
Resources available at Time	Max Power of Resources	Min Power of Resources	Number of Tasks	Arrival Time of Tasks	Processing Time of Task	Utilization of Task	3142 ms In case of 5 Resources
#0 4262.895	1025.0	10.0	#0	100	4095.00	450.45	
#1 4230.135	1025.0	10.0	#1	200	4095.00	491.40	
#2 4211.755	1025.0	10.0	#2	300	4095.00	532.35	
#3 4238.32	1025.0	10.0	#3	400	4095.00	573.30	
#4 4238.32	1025.0	10.0	#4	500	4095.00	614.25	

Table 2 Shows the output parameters of RASA

Number of Resources			Tasks				Final Response Time of RASA
Resources available at Time	Max Power of Resources	Min Power of Resources	Number of Tasks	Arrival Time of Tasks	Processing Time of Task	Utilization of Task	29 ms In case of 5 Resources
#0 460.12	1025.0	10.0	#0	100	442	48.62	
#1 456.58	1025.0	10.0	#1	200	442	53.04	
#2 454.81	1025.0	10.0	#2	300	442	57.40	
#3 457.46	1025.0	10.0	#3	400	442	61.87	
#4 457.46	1025.0	10.0	#4	500	442	66.3	

Table 3 Shows the output parameters of Active Clustering

Number of Resources			Tasks				Final Response Time of Active Clustering
Resources available at Time	Max Power of Resources	Min Power of Resources	Number of Tasks	Arrival Time of Tasks	Processing Time of Task	Utilization of Task	17 ms In case of 5 Resources
#0 104.1	1025.0	10.0	#0	100	100	11.0	
#1 103.3	1025.0	10.0	#1	200	100	12.0	
#2 102.9	1025.0	10.0	#2	300	100	13.0	
#3 103.5	1025.0	10.0	#3	400	100	14.0	
#4 103.5	1025.0	10.0	#4	500	100	15.0	

Table 4 Shows the comparison of ACO, RASA and Active Clustering for Resource/Task Id: 1

Algorithms	Number of Resources/Tasks	Resource Arrival Time	Utilization of Tasks	Processing Time
ACO	For #1	4230.135	491.40	4095.00
RASA	For#1	456.58	53.04	442.0
Active Clustering	For#1	103.3	12.0	100

Similarly we can also draw the table for different resource and task id, every case we can see tht the results of Active clustering are better than the RASA and ACO.

Table 5 Shows the comparison of ACO, RASA with Active clustering for final Response Time.

Algorithms ⟶	ACO	RASA	Active Clustering
Response Time ⟶	3142 ms	29 ms	17 ms

The value of final response time shows that the Active clustering are better than the RASA and ACO for five resources and tasks.

Graphical Charts for Comparison of Results

<u>For Resource id:1</u>

The Resource arrival Time of **ACO: 4213.75, RASA: 454.81, and Active Clustering: 102.9** And The Response Time for **ACO: 3142ms, RASA: 29ms and for Active Clustering: 17ms**. The graphical chart for above values is given below:

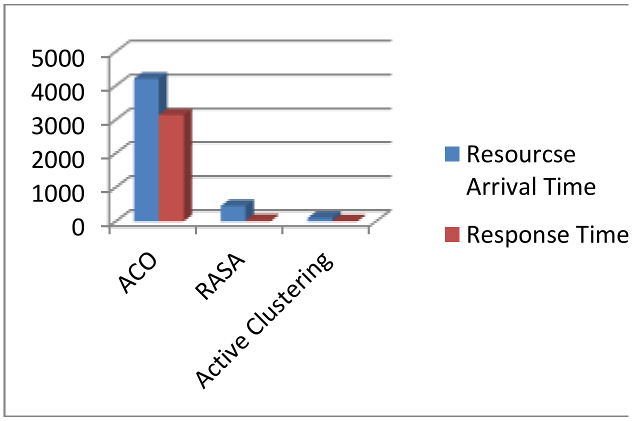

Figure 7 Shows the comparison chart between the ACO, RASA and Active clustering with respect to Resource arrival time and Response time.

<u>For Task id:1</u>

The processing time of Tasks and the Utilizations of Tasks

The Processing time of **ACO: 4095, RASA: 442 and for Active Clustering are 100**. And the Utilizations of Tasks for **ACO: 491.4, RASA: 53.04, and for Active Clustering are 12.**

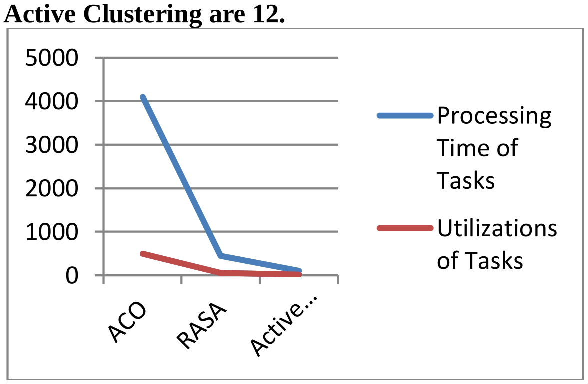

Figure 8 Shows the Line chart for the comparison of ACO, RASA and Active Clustering with respect to processing time and utilizations of tasks.

<u>For Resource /Task ID: 4</u>

The Processing Time of ACO: 4095, RASA: 442, and for Active Clustering: 100.

Utilizations of Tasks of ACO: 614.25, RASA: 66.3 and for Active Clustering: 15

Resource Arrival time of ACO: 4238.32, RASA: 457.46, and for Active Clustering: 103.5

And Response time of ACO: 3142, RASA: 29, and for Active Clustering: 17

Based upon all parameters values, we have proved that the Active clustering algorithm gives better results than RASA and ACO.

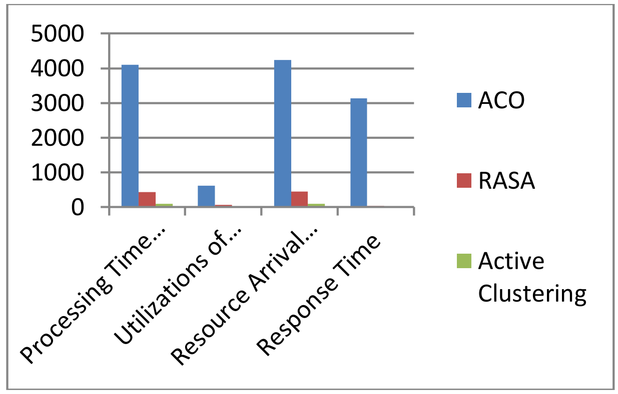

Figure 9 Shows the comparisons of results for ACO, RASA and Active Clustering

Conclusions and future work

I implemented the algorithms in java described in methodology and then used a CloudSim Toolkit for simulations. The results show that the Active Clustering algorithm are much better than the RASA and ACO algorithms in **Response Time**, **Resource Arrival Time**, **Processing Time** and **for Utilizations of Tasks**. Although, the results of RASA are much better than the ACO. But still the results of Active Clustering are overall better than the RASA and ACO.

Future Work

Cloud Computing is a vast concept and load balancing plays a very important role in case of Clouds. There is a huge scope of improvement in this area. I have implemented only three dynamic load balancing algorithms, but there are still other approaches that can be applied to balance the load in clouds. The performance of the given algorithms can also be increased by varying different parameters. Also we can also shift our work on any Private Cloud for the Security and further enhancements.

References

1. *Kumar Nishant, Pratik Sharma, Vishal Krishna, Nitin and Ravi Rastogi "Load Balancing of Nodes in Cloud Using Ant Colony Optimization" in the 14th International Conference on Modeling and Simulation, IEEE 2012*
2. *Klaithem Al Nuaimi, Nader Mohamed,*

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Mariam Al Nuaimi and Jameela Al-Jaroodi "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms" in the proceeding IEEE 2012.

3. *Jianzhe Tai Juemin Zhang Jun Li Waleed Meleis Ningfang Mi "ARA: Adaptive Resource Allocation for Cloud" in the proceeding IEEE 2011.*
4. *Amir Nahir , Ariel Orda, Danny Raz "Schedule First, Manage Later: Network-Aware Load Balancing" in the Proceedings IEEE INFOCOM, 2013.*
5. *Parveen Patel ,Deepak Bansal, Lihua Yuan, Ashwin Murthy, Albert Greenberg " Ananta: Cloud Scale Load Balancing" in SIGCOMM, August12–16, 2013, HongKong,China.*
6. *Isam Azawi Mohialdeen,2013, Comparative Study Of Scheduling Algorithms In Cloud Computing Environment .Journal of Computer Science, 9 (2): 252-263, 2013 ISSN 1549-3636 © 2013 Science Publications.*
7. *Isam Azawi Mohialdeen,2013, Comparative Study Of Scheduling Algorithms In Cloud Computing Environment .Journal of Computer Science, 9 (2): 252-263, 2013 ISSN 1549-3636 © 2013 Science Publications.*
8. *Mohamed Abu Sharkh, Abdelkader Ouda, and Abdallah Shami, "AResource Scheduling Model for Cloud Computing Data centers" in the proceeding IEEE 2013.*
9. *Andre Martin and Christof Fetzer "Active replication at (almost) no cost" in the 30th IEEE International Symposium on Reliable Distributed Systems, 2011.*
10. *Hwa Min Lee · Young-Sik Jeong · Haeng Jin Jang "Performance analysis based resource allocation for green cloud computing" in the Springer Science+Business Media, New York 2013.*
11. *Wei Chen, Junwei Cao_, and Yuxin Wan "QoS-Aware Virtual Machine Scheduling for Video Streaming Services in Multi-Cloud" in the Tsinghua Science and Technology,ISSNll10070214ll10/11llpp308-317 Volume 18, Number 3, June 2013.*
12. *Dhinesh Babu L.D., P. Venkata Krishna "Honey bee behavior inspired load balancing of tasks in cloud computing environments" in the Applied Soft Computing , at SciVerse ScienceDirect 2013.*
13. *Zhen Xiao, Weijia Song, and Qi Chen "Dynamic Resource Allocation Using Virtual Machines For Cloud Computing Environment" In the IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 6, June 2013.*
14. *Navendu Jain · Ishai Menache "A Truthful Mechanism for Value-Based Scheduling in Cloud Computing" published in the Springer Science+Business Media New York 2013.*
15. *Zheng Hu, Kaijun Wu, Jinsong Huang "An Utility-Based Job Scheduling Algorithm for Current Computing Cloud Considering Reliability Factor" in the proceeding IEEE 2012.*
16. *Mrs.S.Selvarani1,Dr.G.Sudha Sadhasivam "Improved Cost-Based Algorithm For Task Scheduling In Cloud Computing" in the proceeding IEEE 2010.*
17. *Ahmed S. Ghiduk " A New Software Data-Flow Testing Approach via Ant Colony Algorithms" in the Universal Journal of Computer Science and Engineering Technology, Oct. 2010.*
18. *Saeed Parsa and Reza Entezari-Maleki "RASA: A New Task Scheduling Algorithm in Grid Environment" in the World Applied Sciences Journal, 2009.*
19. *R.R. Kotkondawar,P.A. Khaire " A Study of Effective Load Balancing Approaches in Cloud Computing" in the International Journal of Computer Applications (0975 – 8887) Volume 87 – No.8, February 2014.*
20. *Dr. Hemant S. Mahalle Prof. Parag R. Kaveri Dr.Vinay Chavan " Load Balancing On Cloud Data Centres" in the International Journal of Advanced Research in Computer Science and Software Engineering , Volume 3, Issue 1, January 2013.*

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Author Biblography

	**Ashima Narang** Working as an Asst. Prof. in an Engineering College in Computer Science Department.Area of interest are Cloud Computing, Big Data, Virtualization.
	**Dr. Vijay Laxmi** Working at Guru Kashi University in Computer Science Department as an Associate Professor and Head of Department.