International Journal of Engineering & Technology, 7 (4.7) (2018) 131-135

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Cloud Computing based on the Load Balancing Algorithm

NV Abhinav Chand[1], A Hemanth Kumar[2], Surya Teja Marella*[3]

[1] *Student, Department of Computer Science Engineering, Koneru lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India*
[2] *Student, Department of Computer Science Engineering, Koneru lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India*
[3] *Student, Department of Computer Science Engineering, Koneru lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.*
**Corresponding author E-mail: suryatejamarella@gmail.com*

Abstract

Emerging cloud computing technology is a big step in virtual computing. Cloud computing provides services to clients through the internet. Cloud computing enables easy access to resources distributed all over the world. Increase in the number of the population has further increased the challenge. The main challenge of cloud computing technology is to achieve efficient load balancing. Load balancing is a process of assigning load to available resources in such a way that it avoids overloading of resources. If load balancing is performed efficiently, it improves QoS metric including cost, throughput, response time, resource utilization and performance. Efficient load balancing techniques also provide better user satisfaction. Various load balancing algorithms are used in different scenarios for ensuring the same. In the current research, we will study different algorithms for load balancing and benefits and limitations caused to the system due to the algorithms. In this paper, we will compare static and dynamic load balancing algorithms for various measures of efficiency. These will be useful for future research in the concerned field.

Keywords: *Cloud Computing, Load Balancing Algorithms Virtualization, Task allocation, Static and Dynamic Load Balancing.*

1. Introduction

CC gives a flexible method to retain the data that involves virtualization, web services, and distributed computing. The main aim of cloud computing is to minimize cost and maximize services. Large numbers of devices (PCs, laptops, Mobile phones, etc.) are connected to the internet. These devices submit their requests and expect a response without delay. Cloud computing is continuously developing to organizing applications for increased scalability, availability and fault tolerance. [1].

The above deployment models are classified on the basis of access permissions. The Public cloud can be accessed by anybody.[2] The private cloud is used by the organization and it is to be accessed only by specific people who have permission to access it. Hybrid cloud is a combination of private and public cloud computing models[3]. Similarly, a community cloud is used by several organizations but it cannotbe accessed by people not having appropriate permissions[4].



Fig 1: Architecture of cloud computing

Services provided by cloud computing are:

1. Software as a Service (SaaS): In this service, customers are able to hire software hosted by vendor[5].
2. Platform as a Service (PaaS): In this service, customers can hire infrastructure and programming tools for creating applications[6].
3. Infrastructure as a Service (IaaS): In this service, customers can hire fundamental computing resources like processing, networking, storage, etc[7].

The paper is organized as follows: Section 2 covers concept of Load balancing and section 3 explains some load balancing algorithms[8]. After discussing Load balancing algorithms westudy comparison between these algorithms in Section 4 and then we conclude this paper with findings.[9]

2. Load Balancing

Various types of loads in cloud computing are CPU load, network load, memory capacity issue etc. [10].

Copyright © 2016 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

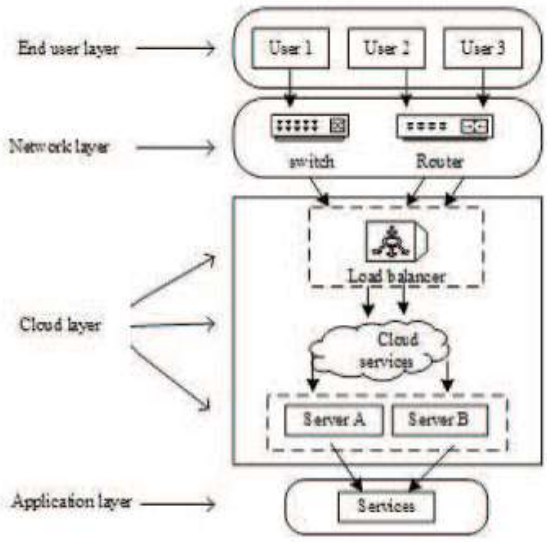


Fig 2 : Load Balancing

Need of Load balancing:

- To maximize user satisfaction.
- To increase resource utilization.
- To minimize execution time and waiting time of task.
- To maximize performance.
- To increase fault tolerance
- To maintain stability of system[11].

Load balancing uses metrics for improving its performance. These metrics are explained below:

- ✓ ***Throughput*:** It is measure of execution of process and volume of time required for it.
- ✓ ***Makespan*:** It is span of time between arriving of job and destroying of job.
- ✓ ***Response Time*:** It is measure oftime taken by task to arrive in the system.
- ✓ ***ResourceUtilization*:** It is appropriate and efficient way of assigning there source for performing.
- ✓ ***Overhead*:** It is Interlinkbetween neighbouring nodes on itstask movement.
- ✓ ***Fault Tolerant*:** System should avoid taskreplication in device. System should be able to rectify faults.
- ✓ ***Migration Time*:** It istime for moving from one node to another node[12].
- ✓ ***Performance*:** It is measure of execution result of system.
- ✓ ***Scalability*:** It is ability of system to expand or decrease number of processors and machines influenced by it.
- ✓ ***WaitingTime*:** It is time spent by task waiting in thequeue for arriving[13].

3. Load Balancing

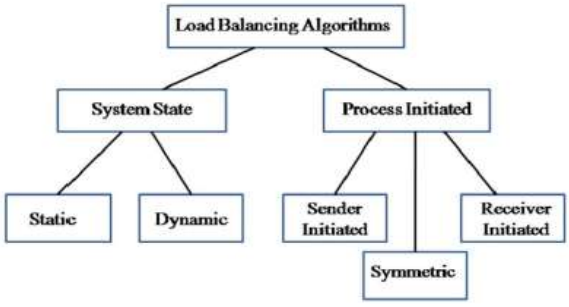


Fig 3: Load Balancing Algorithms

Load balancing is classified in three types based on who is initiator of process. They are as follows:

1. Sender Initiated: In this type, sender initiates execution of load balancing algorithm, after detecting over loaded nodes[14].
2. Receiver Initiated: In this type, imbalance in load is detected by receiver. The receiver than initiates' execution of load balancing algorithm.
3. Symmetric: It is a mixture of both sender and receiver initiated category of load balancing algorithms[15].

Load balancing algorithms are basically divided into two types depending on state of system. They are explained below:

3.1 Static Load Balancing

In static load balancing tasks are assigned evenly to all nodes. These tasks are assigned before execution process starts. The algorithm has to have prior knowledge of system like, processing, memory, performance and power. The static algorithms do not need information of current state of node. It takes decisions during compile time. Properties of nodes should be known in advance. Static algorithms are less complex. Static algorithms work better, when variation of load is low[16].

3.1.1 Round Robin Algorithm

Round Robin Algorithm uses slice mechanism to process data. This algorithm selects first node randomly and then assigns tasks to other nodes by using Round Robin technique. In this algorithm, time is divided into slices. All nodes have assigned time interval[17]. Given task should be performed by node within assigned time interval. In case task is not completed within allocated time, task has to wait for next slot. The tasks are assigned to processor without any priority. Since distribution of load is not uniform, it is not possible to predict running time of node in advance. The algorithm is usually, avoided because execution time is not known in advance.[18]

3.1.2 Min-Min Algorithm

In this algorithm, parameters of job are known in advance. Cloud manager deals with jobs having minimum execution time on priority. This means that jobs having maximum executing time have to wait for undefined time period. Before assigning tasks to processor, the assigned tasks are updated in processor and then are removed from waiting queue. Most important drawback of this algorithm is that it leads to starvation[19].

3.1.3 MIN-MAX

Working of Max-Min algorithm is quite similar to Min-Min algorithm before finding out minimum execution time. After execution time is known, cloud manager takes care of tasks having maximum execution time. The assign task is removed from list and execution time of unassigned tasks is informed to processor. The algorithm performs well since requirements are known in advance. Improvement in efficiency and concurrent execution of tasks is possible in advance ersion of this algorithm [20].

3.1.4 Opportunistic Load Balancing Algorithm

This algorithm does not consider current workload of VM. Its tries to keep all nodes busy. This algorithm deals with tasks in random order. Tasks are assigned to node randomly. The tasks are executed slowly because current execution time of nodes is not calculated.This algorithm provides load balance schedule, but results are not good.[21]

3.2 Dynamic load Balancing

In dynamic load balancing, decision for load balancing is made in current state of system. The previous state and behaviour of nodes is checked before assigning tasks to nodes. The Dynamic load balancing algorithms are implemented in two forms which are distributed and non-distributed. In distributed dynamic load balancing, all nodes interact with one another and distribute tasks among nodes. In non-distributed dynamic load balancing, all nodes work independently and achieve their tasks. Dynamic load balancing algorithms implement various policies like transfer policy, location policy, selection policy and information policy [22].

- Transfer Policy: The role of transfer policy is to select a job to transfer from local node to remote node.
- Selection Policy: The role of selection policy is to specify processors involved in load exchange.
- Location Policy: The role of location policy is to select destination node for transferred task.
- Information Policy: The role of information policy is to collect information about nodes.

These policies are used to balance load considering changes in state of nodes. Load of heavily loaded node is transferred to lightly loaded node[23].

3.2.1 Ant Colony

This is designed for seeking optimal path like a path between food and colony of ants. The important responsibility of this algorithm is to distribute work load efficiently among nodes in system. In this algorithm, head node is regional load balancing node. Once request is sent, ant starts seeking path to food from head node. The ants collect data from cloud, and assign task to node using the data. After task is assigned, ant moves with overloaded node to next node to check if the node is overloaded. This continues till job is completed. The result is then reported considering individual result of every ant. After collecting report from individual ant, reports of all ants are combined together to develop a complete report. [24].

3.2.2 Honey Bee

Honey bee is published according to the manner of honey bees. Honey bees are of two types: finders and reapers. The finder honeybee finds source of honey and do waggle dance to indicate quality and quantity of honey [25]. Each server has a process queue. If the request is received it calculates the profit, similar to waggle dance of bees. If profit is high, then server stays else it moves to forage. It indicates whether the state is overloaded, under loaded, loaded or balanced. Current VMs are combined, based on this. Separate queue is maintained for each and every node. Task is executed, depending on priority. Advantages of this algorithm are less response time and less waiting time. Disadvantage of this algorithm is its minimum throughput.[26]

3.2.3 Biased Random Sampling

Biased random sampling treats all servers like nodes. This algorithm can be represented as virtual, graph. This graph has load on each node. When load balancer receives request from client, it assigns job to node that has minimum degree. These processes are assigned and deleted by random sampling technique. Important parameter is threshold value[27]. If it is greater than current walk length the node executes its job and if it is less than current walk then node moves to another randomly selected node. Increase in number of servers, decreases performance of this algorithm[28].

3.2.4 Active clustering

Active Clustering is known as improved random sampling. This algorithm uses concept of clustering. The algorithm groups together similar nodes. This grouping increases throughput.Match-Maker method is introduced in this algorithm. This method connects neighbouring nodes which are similar to each other. After the similar nodes are connected, match maker node is disconnected. Load is balanced equally by iterating this process. This algorithm has high performance and high throughput. The resources are utilized efficiently along with increase in throughput[29].

Further important algorithms of load balancing are (v) Genetic algorithm and (vi) Firefly algorithm. We will compare these two algorithms on basis of their two reasons [30].

The two reasons are

- Completion Time
- Processor Utilization

3.2.5 Genetic Algorithm

Genetic algorithm is based on evaluation and natural genetics principles. The algorithm combines past results with recent domains of research. Survival of fittest technique is used for information exchange in his algorithm. Generation can be defined as a collection of artificial creatures. New generation is created from set of strings of previous generation[31]. Sometimes new strings are tries for research and development. Another feature of genetic algorithm is randomization. Though the algorithm is randomized, I efficiently exploit historical data for research in current domain. Most of the optimization methods move from one point to another in decision space using transition rule. The transition rule decides next point. This method is error prone because it can locate false peaks in multimodal, which is many peaked search space. On the contrary, genetic algorithm works on database of points simultaneously[32]. Thus, it has less probability of finding false peaks. Genetic algorithm is comparatively very simple and it does not involve any complexities other than 'copying strings' and 'swapping partial strings'.

There are two main attractions of genetic algorithm are simplicity of operation and power of effect[33]. For this algorithm to be more effective, exploitation and exploration are equally important. The effectiveness is achieved by selection, crossover, and mutation. Selection is source of exploitation[34]. The mutation and crossover are sources of exploration. There exist direct trade-off of exploration and exploitation with mutation. Increase in rate of mutation makes mutation more disruptive. It continues becoming disruptive until the exploitative effects of selection are overwhelmed[35].

3.2.6 Firefly algorithm

Firefly algorithm can be described using three idealized rules:

(1) Fireflies are unisexual creatures. Thus, he firefly can attract any other firefly, regardless of their sex[36].
(2) For fireflies attractiveness is directly proportional to brightness. The less bright firefly moves towards brighter firefly. The brightest firefly moves randomly[37].
(3) The brightness of a firefly is determinedby objective function. The brightness is directly proportional to objective function[38].

The Firefly algorithm is uses simulated cloud network. The network deals with set of requests and servers. This algorithm is implemented in three steps. In first step population is generated. In second step, scheduling index calculation is subjected. In third step, the scheduled list is optimised[39].

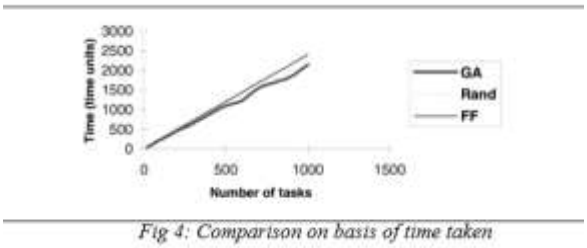


Fig 4: Comparison on basis of time taken

The Figure 4 shows the completion time taken by all three algorithms increases with increase in number of tasks. As the number of tasks increases, time taken to complete these tasks increases [40]. The result showed that Genetic algorithm performed way much better than other algorithms. We can observe that the gap between Genetic algorithm and firefly algorithm widens with increase in number of tasks[41]. This shows that genetic algorithm reduces total completion time as compared to other algorithms.

3.3 Average Processor Utilization

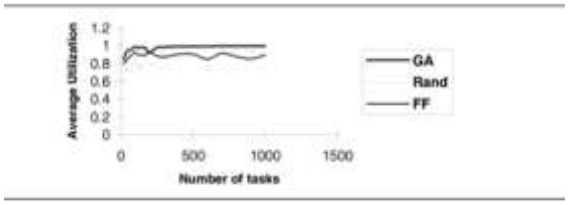


Figure 5: Comparison on basis of processor utilization

Genetic algorithms outperform other algorithms like firefly algorithm in terms of processor utilization. The processor utilization for genetic algorithms is 85 to 99 per cent and for firefly algorithm processor utilization was 80 to 91 per cent. When he number of asks increased processor utilization of Genetic algorithm reached 99 per cent. His indicates that Genetic algorithm works better in case of more tasks. The result of research is displayed in Figure 5.

Algorithms	Static / Dynamic	Benefits	Limitations
Round Robin	Static	• Time quantum is fixed. • Easy to understand.	• Larger task requires more time to complete. • Context switches may occur because of short quantum time.
Max-Min	Static	• Works better.	• Lot of time is required for completing a task.
Min-Min	Static	• Performs better, if tasks allocated are small.	• Leads to starvation of tasks. • Task variation and machine variation is unpredictable.
OLB	Static	• Resource utilization is improved.	• Requires more time for completion of task.
Ant Colony Optimization	Dynamic	• Information is collected faster. • Minimum Make span.	• Complex. • Since, network is over headed; long time is spent in searching. • Number of ants is not clear.
Honey Bee	Dynamic	• Maximum Throughput. • Minimum Response time.	• VM is required for executing high priority tasks.
Biased Random Sampling	Dynamic	• Better performance and resource utilization.	• More response time.
Active clustering	Dynamic	• Nodes with similar characteristics are grouped together.	• Performance decreases with increase in number of nodes.

I. Table2: Comparison of Algorithms by Evaluation Parameters[8]

Parameter	Round Robin	OLB	Min-Min	Active Clustering	Min-Max	Ant Colony	Honey Bee	Biased Random Sampling
Throughput	High	Low	High	Low	High	High	High	High
Overhead	High	Low	High	High	High	Low	Low	Low
Fault Tolerance	No	No	No	No	No	No	No	No
Response Time	Fast	Slow	Fast	Slow	Fast	Slow	Slow	Slow
Resource Utilization	High	High	High	Low	High	High	Low	Low
Scalability	No	No	No	No	No	Yes	Yes	No
Performance	Fast	Fast	Fast	Slow	Fast	Fast	Fast	Fast
Speed	N/A	Slow	Fast	N/A	Slow	Fast	Fast	N/A
Complexity	Low	High	Low	N/A	Low	Low	Low	N/A
Fairness	Yes	No	No	N/A	No	No	No	N/A

4. Conclusion

CC gives service to user over network. Major problem in cloud computing is load balancing. Overloading of a single node or multiple nodes negatively affects performance of the system. To avoid this and other effects of overloading, various load balancing algorithms are implemented. In this paper, we have studied some load balancing algorithms, proposed by various researchers. These load balancing algorithms are compared on basis of various parameters in this paper for further research.

References

[1] D. Saranya, L. SankaraMaheswari "Load Balancing Algorithms in Cloud Computing: A Review" International Journal of Advanced Research in Computer Science and Software Engineering. Volume 5, Issue 7, July 2015 ISSN: 2277 128X.

[2] G. Kanmani, E Jayabalan "A survey on recent improved load balancing algorithms in cloud environment" SSRG International Journal of Computer Science and Engineering – NCSACT – 2017 ISSN : 2348-8387.

[3] Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms" 2012 IEEE Second Symposium on Network Cloud Computing and Applications.

[4] Sidra Aslam, Munam Ali Shah "Load Balancing Algorithms in Cloud Computing: A Survey of Modern Techniques" 2015 National Software Engineering Conference (NSEC 2015).

[5] T. Deepa Dr. DhanarajCheeluA Comparative Study of Static and Dynamic Load Balancing Algorithms in Cloud International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017).

[6] P. Byuvol, L. Gabsalikhova, I. Makarova, E. Mukhametdinov, G. Sadygova, "Improving the Branded Service Network Efficiency based on its Functioning Evaluation", Astra Salvensis, Supplement No. 2, p. 373, 2017.

[7] RatanMishra,AnantJaiswal "Ant colony Optimization: A solution of Load balancing in Cloud. International Journal of Web & Semantic Technology (IJWesT) Vol.3, No.2, April 2012

[8] Shanti Swaroop Moharana, Raja Deepan D. Ramesh Digamber-Powar" Analysis of Load Balancers in Cloud Computing",Vol.2,Issue.2, (May2013)

[9] Albert Y. Zomaya, Senior Member, IEEE, Yee-HweiTeh "Observations on Using Genetic Algorithms for Dynamic Load-Balancing". IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 12, NO. 9, SEPTEMBER 2001.

[10] Performance Tradeoffs in Static and Dynamic Load Balancing Strategies, NASA March 1986

[11] Bohn, R. B., Messina, J., Liu, F., Tong J. and Mao J. (2011) 'NIST Cloud Computing Reference Architecture', IEEE World Congress on Services, 400 Washington, DC, pp. 594–596.

[12] Tsai, C. W., and Rodrigues, J. J. (2014) 'Metaheuristic scheduling for cloud: A survey', IEEE Systems Journal, 8(1), pp. 279-291.

[13] Mishra, S. K., Puthal, D., Sahoo, B., Jena, S. K., and Obaidat, M. S. (2017) 'An adaptive task allocation technique for green cloud computing', 405 The Journal of Supercomputing, pp. 1-16.

[14] Ibrahim, A. H., Faheem, H. E. D. M., Mahdy, Y. B., and Hedar, A. R. (2016) 'Resource allocation algorithm for GPUs in a private cloud', International Journal of Cloud Computing, 5(1-2), pp. 45-56.

[15] Korolev, A. and Sussman, B., 2000. A technique for habit classification of cloud particles. Journal of Atmospheric and Oceanic Technology, 17(8), pp.1048-1057.
[16] Helmer, E.H., Kennaway, T.A., Pedreros, D.H., Clark, M.L., Marcano-Vega, H., Tieszen, L.L., Ruzycki, T.R., Schill, S.R. and Sean Carrington, C.M., 2008.
[17] Li, P., Li, J., Huang, Z., Gao, C.Z., Chen, W.B. and Chen, K., 2017. Privacy-preserving outsourced classification in cloud computing. Cluster Computing, pp.1-10.
[18] Land cover and forest formation distributions for St. Kitts, Nevis, St. Eustatius, Grenada and Barbados from decision tree classification of cloud-cleared satellite imagery. Caribbean Journal of Science, 44(2), pp.175-198.
[19] Alonso–Calvo, R., J. Crespo, M. Garc'ia–Remesal, A. Anguita, and V. Maojo (2010) "On Distributing Load in Cloud Computing: A Real Application for Very-large Image Datasets", Procedia Computer Science (1)1, pp. 2669– 2677, doi: 10.1016/j.procs.2010.04.300.
[20] American_Society_for_Quality (2006) "Idea Creation Tools―Affinity Diagrams," http://asq.org/learn-aboutquality/idea-creation-tools/overview/affinity.html (current June 20, 2011).
[21] Anthes, G. (2010) "Security in the Cloud," Communications of the ACM (53)11, p. 16. Armando, F. (2011) "Cloud Computing―What's in It for Me as a Scientist?" Science (331)6016, p. 406
[22] Banerjee, P., R. Friedrich, C. Bash, P. Goldsack, B.A. Huberman, J. Manley, et al. (2011) "Everything as a Service: Powering the New Information Economy," Computer (44)3, pp. 36–43.
[23] Barki, H., S. Rivard, and J. Talbot (1993) "A Keyword Classification Scheme for IS Research Literature: An Update," MIS Quarterly, June, pp. 209–225.
[24] Barnhill, D.S. (2010) "Cloud Computing and Stored Communications: Another Look at Quon v. Arch Wireless," (Privacy Law) (Annual Review of Law and Technology), Berkeley Technology Law Journal (25), pp. 621–648. Bellovin, S.M. (2011) "Clouds from Both Sides, IEEE Security & Privacy (9)3, pp. 88–88
[25] Currie, C. (2008) "Painting the Clouds," EDUCAUSE Review (43)6, p. 28.
[26] Cusumano, M. (2010) "Technology Strategy and Management: Cloud Computing and SaaS as New Computing Platforms," Communications of the ACM (53)4, pp. 27–29.
[27] Deelman, E. (2010) "Grids and Clouds: Making Workflow Applications Work in Heterogeneous Distributed Environments," The International Journal of High-Performance Computing Applications (24)3, p. 284.
[28] Kumar, R. and Charu, S., 2015. An importance of using virtualization technology in cloud computing. Global Journal of Computers & Technology, 1(2).
[29] Luo, S., Lin, Z., Chen, X., Yang, Z. and Chen, J., 2011, December. Virtualization security for cloud computing service. In Cloud and Service Computing (CSC), 2011 International Conference on (pp. 174-179). IEEE.
[30] Mijumbi, R., Serrat, J., Gorricho, J.L., Bouten, N., De Turck, F. and Boutaba, R., 2016. Network function virtualization: State-of-the-art and research challenges. IEEE Communications Surveys & Tutorials, 18(1), pp.236-262.
[31] Azad N., Ghandvar P., Rahimi Z., "Online Search Behaviour of Customers in Shoe Market", Astra Salvensis, Supplement No. 2, p. 793, 2017.
[32] Pankaj Sareen, (March 2013) "Cloud Computing: Types, Architecture, Applications, Concerns, Virtualization and Role of IT Governance in Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, ISSN: 2277 128X.
[33] [32] Farzad Sabahi, (February 2012) "Secure Virtualization for Cloud Environment Using Hypervisor-based Technology", International Journal of Machine Learning and Computing, Vol. 2, No. 1.
[34] Siddharth Jain, Rakesh Kumar, Anamika, Sunil Kumar Jangir, (Dec 2014) "A Comparative Study for Cloud Computing Platform Open Source Software", ABHIYANTRIKI: An International Journal of Engineering & Technology (AIJET), Vol. 1, No. 2, pg: 28-35.
[35] Rakesh Kumar, Bhanu Bhushan Parashar, (November 2014) "Dynamic Resource Allocation and Management Using OpenStack", National Conference on Emerging Technologies in Computer Engineering (NCETCE) – 2014, Supported by Computer Society Chapter, IEEE Delhi Section.
[36] Lin, W., Liang, C., Wang, J.Z. and Buyya, R., 2014. Bandwidth-aware divisible task scheduling for cloud computing. Software: Practice and Experience, 44(2), pp.163-174.
[37] Robertazzi, T.G., 2003. Ten reasons to use divisible load theory. Computer, 36(5), pp.63-68.
[38] Kim, S. and Weissman, J.B., 2004, August. A genetic algorithm based approach for scheduling decomposable data grid applications. In Parallel Processing, 2004. ICPP 2004. International Conference on (pp. 406-413). IEEE.
[39] Takabi, H., Joshi, J.B. and Ahn, G.J., 2010. Security and privacy challenges in cloud computing environments. IEEE Security & Privacy, (6), pp.24-31.
[40] Zhang, Q., Cheng, L. and Boutaba, R., 2010. Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications, 1(1), pp.7-18.
[41] Rong, C., Nguyen, S.T. and Jaatun, M.G., 2013. Beyond lightning: A survey on security challenges in cloud computing. Computers & Electrical Engineering, 39(1), pp.47-54.
[42] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing V2.1,"