A Hybrid Approach for VM Load Balancing in Cloud Using CloudSim

Rajkumar Somani, Jyotsana Ojha

***Abstract*--- In this work a hybrid approach has been proposed for virtual machine level load balancing using concepts of two classical algorithms for load balancing. These two classical algorithms for load balancing are Round Robin Algorithm and Throttled algorithm. It has also been implemented for an IaaS framework in simulated cloud computing environment and the results obtained were analyzed. The hybrid approach gave better results in terms of: Response time, Data center request serving time and Data center processing time, when compared with results of Round robin algorithm, Throttled algorithm and equally spread current execution algorithm separately. The proposed hybrid algorithm was found to be efficient in case of same data size per request as well as for different data size per request.**

***Keywords* --- Round Robin Algorithm, Throttled Algorithm, Response time.**

I. INTRODUCTION

A. Need of Load Balancing in cloud computing

Cloud computing is a current trends of technology, where user is able to rent software, hardware, infrastructure and computational resources as per user basis.

Load Balancing is essential for efficient operations in distributed environments. As Cloud Computing is a greatest platform which provides storage of data in very lower cost and available for all time over the internet, that's why load balancing for the cloud has become a very interesting and important research area. Load balancing helps to attain a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource.

The load can be memory capacity, CPU load, delay or network load. Load balancing is the method of distributing the load between various nodes of a distributed system to get better both resource utilization and job response time while also staying away from a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. Load balancing makes sure that all the processor in the system or every node in the network does more or less the equal amount of work at any moment of time. [10].

The goals of load balancing are to:

(a)Improve the performance

(b)Maintain system stability

(c)Build fault tolerance system

(d)Accommodate future modification.

(e)Increase the availability

(f)Increase the user satisfaction

(g)Improve the resource utilization ratio

(h)Minimize the waiting time of job in queue as well as to reduce job execution time [11] [12].

B. Qualitative Metrics for Load Balancing

Different qualitative parameters or metrics that are considered vital for load balancing in cloud computing are discussed as follows:

1. Throughput: The total number of tasks that have completed execution is called throughput. A high throughput is required for better performance of the system.

2. Associated Overhead: It is the amount of overhead that is produced by the execution of the load balancing algorithm. Minimum overhead is expected for successful implementation of the algorithm.

3. Fault tolerant: It is the ability of the algorithm to perform correctly and uniformly even in conditions of failure at any arbitrary node in the system.

4. Migration time: It is the time taken in migration or transfer of a task from one machine to any other machine in the system. This time should be minimal for improving the performance of the system.

5. Response time: It is the smallest amount of time, a distributed system executing a specific load balancing algorithm takes to respond.

6. Resource Utilization: It is the degree to which the resources of the system are utilized. A good load balancing algorithm provides maximum resource utilization.

7. Scalability: It determines the ability of the system to accomplish load balancing algorithm with a restricted number of processors or machines.

8. Performance: It represents the effectiveness of the system after performing load balancing. If all the above parameters are satisfied optimally then it will highly improve the performance of the system [2].

II. ALGORITHMS

The establishment of an effective load balancing algorithm and how to use Cloud computing resources efficiently for effective and efficient cloud computing is one of the Cloud computing service provider's ultimate goals. The basic idea of cloud computing is to offer resources such as VMs as services on demand. Allocating efficient VM on demand is being carried out with the help of the load balancing

algorithms in the cloud computing. As the load balancing algorithm plays a vital role while deciding which VM is to be allocated on demand of the user. While providing services it is likely to have a number of requests at a time and due to that some requestors need to stay in queue though they have possibility to send request to other service provider.

A. Round Robin Algorithm

It is a static load balancing algorithm, which does not consider the previous load state of a node at the time of assigning jobs. It makes use of the round robin scheduling algorithm for allocating jobs. It selects the first node arbitrarily and then, allocates jobs to all other nodes in a round robin manner [15].

This algorithm works on random selection of the virtual machines. The datacenter controller allocates the requests to a list of VMs on a rotating basis. The first request is allocated to a VM chosen randomly from the group and then the Data Center controller assigns the requests in a circular order. Once the VM is allotted the request, the VM is shifted to the end of the list [13].

Round-robin is by far the simplest algorithm available to distribute load among nodes. Because of this reason it is often the first preference when implementing a simple scheduler. One of the reasons for it being so simple is that the only information required is a list of nodes.

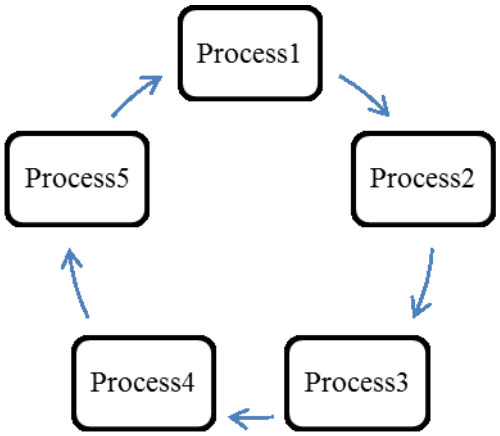

Fig. 1: Round Robin processing [8]

The round robin algorithm is as follows:

Step1. Round Robin VM load Balancer maintains an index of VMs. At start all VM's have zero allocation.

Step2.

a. The data center controller receives the user requests/cloudlets.

b. The requests are allocated to VMs in circular way.

c. The round robin VM load balancer will allocate the time quantum for user request execution.

Step3. After the execution of cloudlets, the VMs are de-allocated by the Round Robin VM Load balancer.

Step4. The data center controller checks for new /pending/waiting requests in queue.

Step5. Continue from step-2.

B. Throttled Algorithm

Throttled load balancer is a dynamic load balancing algorithm. In this throttled algorithm the client first requests the load balancer to find an appropriate Virtual Machine to perform the required operation. The process first starts by maintaining a list of all the VMs. Each row is individually indexed to accelerate the lookup process. If a match is found on the basis of size and availability of the machine, then the load balancer accepts the request of the client and assigns that VM to the client. On the other hand, if there is no VM available that matches the criteria then the load balancer returns -1 and the request is queued.

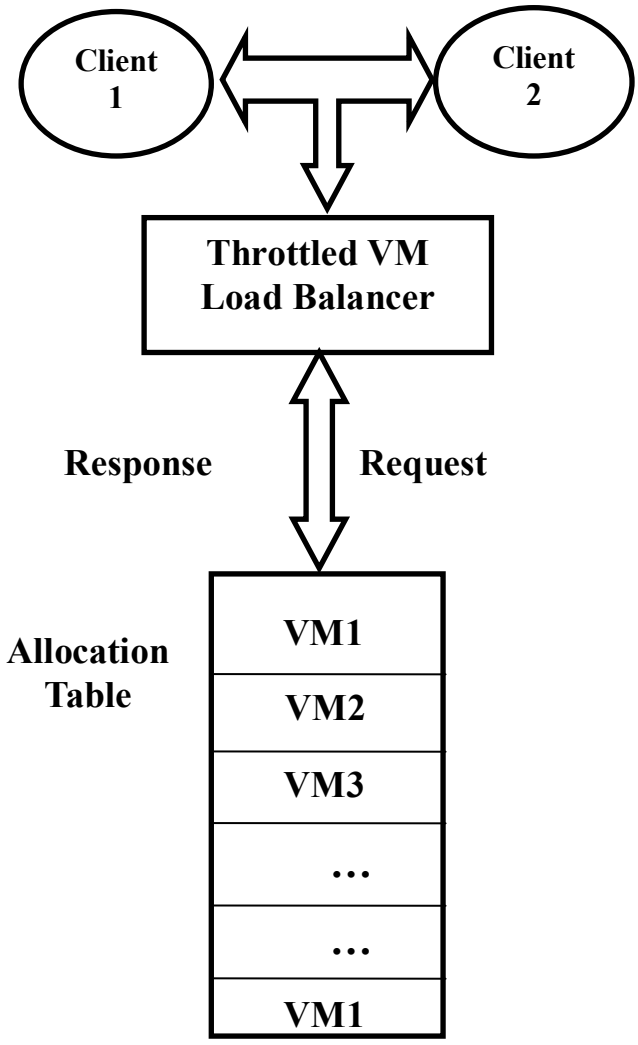

Fig.2. Throttled scheduling process [8]

The throttled algorithm is as follows:

Step1. Throttled VM Load Balancer keeps an index table of VMs and the state of the VM (BUSY/AVAILABLE). At the beginning all VM's are available.

Step2. Data Center Controller receives a new request.

Step3. Data Center Controller queries the Throttled VM Load Balancer for the next allocation.

Step4. Throttled VM Load Balancer parses the allocation table from top until the first available VM is found or the table is parsed completely.

If found:

i) The Throttled VM Load Balancer returns the VM id to the Data Center Controller.

ii) The Data Center Controller sends the request to the VM identified by that id.

iii) Data Center Controller notifies the Throttled VM Load Balancer of the new allocation.

iv) Throttled VM Load Balancer updates the allocation table accordingly.

If not found:

i) The Throttled VM Load Balancer returns -1.

ii) The Data Center Controller queues the request.

*Step5***.** When the VM finishes processing the request, and the Data Center Controller obtains the response cloudlet, it informs the Throttled VM Load Balancer of the VM de-allocation.

Step6. The Data Center Controller checks if there are any waiting requests in the queue. If there are, it carries on from step 3.

Step7. Continue from step 2 [8].

C. Equally Spread Current Execution Algorithm

This algorithm distributes the tasks among VMs in a way to even out the number of active tasks at any given time on each VM. Fig.3. shows the process of this algorithm.

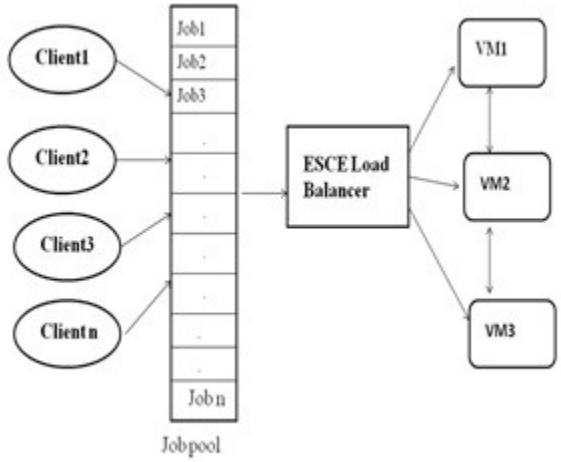

Fig.3. Equally Spread Current Execution Load Balancing [8].

The ESCE Load Balancing algorithm is as follows:

Step1. Find the next available VM .

*Step2***.** Check for all current allocation count is less than max length of VM list allocate the VM.

Step3. If available VM is not allocated create a new one.

Step4. Count the active load on each VM.

Step5. Return the id of those VM which is having least load.

Step6. The VM Load Balancer will allocate the request to one of the VM.

Step7. If a VM is overloaded then the VM Load Balancer will distribute some of its work to the VM having least work so that every VM is equally loaded.

Step8. The data center controller receives the response to the request sent and then allocates the waiting requests from the job pool/queue to the available VM & so on.

Step9. Continue from step-2 [8].

III. PROPOSED HYBRID ALGORITHM

In this work a Hybrid approach having concepts from Round Robin algorithm and Throttled algorithm has been proposed for virtual machine level load balancing. Round robin is a classical algorithm for load balancing that does not consider the current load state of virtual machine while allocating some new job to it. Inspiration for taking current load state into account came from Throttled algorithm. In Hybrid approach current load state of virtual machine was taken into account to take decision about allocation of jobs to virtual machines. The proposed hybrid algorithm was efficient in case of same data size per request as well as for different data size per request.

In Proposed hybrid approach, initially when a cloudlet is received for execution on VMs then an arbitrary VM is chosen and checked for availability. If it is available then cloudlet is allocated to it. Otherwise control keeps moving in circular way through the VM list until it gets an available VM for cloudlet allocation. After allocation of cloudlet, next comparison takes place at the next VM coming in that circular way.

In this proposed hybrid algorithm, the concept of circular way to allocate VMs to cloudlets has been taken from Round Robin algorithm. And inspiration of checking availability on each step has been taken from throttled algorithm.

The proposed algorithm is as follows:

Step1. Hybrid Load Balancer maintains an index of VMs and state of the VMs (busy/available). At start all VMs have zero allocation.

Step2.

a. The Data Center Controller receives the user requests/cloudlets.

b. All VMs are checked in circular way.

c. The requests are allocated to VMs on the basis of their status known from the VM queue:

if status is available: then cloudlet is allocated to that VM.

else availability of next VM is checked for that cloudlet.

d. The Hybrid Load Balancer will allocate the time quantum for user request execution.

Step3. After the execution of cloudlets, the VMs are de-allocated by the Hybrid Load Balancer.

Step4. The Data Center Controller checks for new /pending/waiting requests in queue.

Step5. Continue from step-2.

IV. PROPOSED HYBRID LGORITHM'S DEPLOYMENT

The proposed hybrid algorithm for virtual machine level load balancing is implemented for an IaaS framework in

simulated cloud computing environment and all the results are analyzed. All this work is done using Cloud Analyst tool. This tool is completely based on java.

Following versions of tools and softwares are used during work:

a) Eclipse Kepler

b) Java1.7.0

c) Cloud Analyst (A tool based upon cloudsim)

Four algorithms for virtual machine load balancing were executed one by one in simulated environment and results were analyzed. These four algorithms were:

(a)Proposed Hybrid algorithm

(b)Round Robin algorithm

(c)Throttled algorithm

(d) Equally Spread Current Execution algorithm

V. RESULT ANALYSIS

The hybrid approach gave better results in terms of Response time, Data center request serving time and Data center processing time, when compared with results of Round robin algorithm, Throttled algorithm and equally spread current execution algorithm separately. The proposed hybrid algorithm was efficient in case of same data size per request as well as for different data size per request.

Comparison among these four algorithms is given below in Table1 for same data size of requests and in Table 2 for different data size of requests.

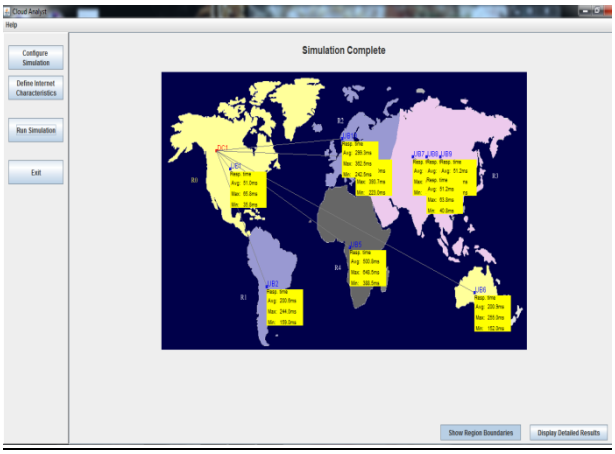

Fig.4. Simulation statistics for hybrid load balancing algorithm for same data size per request

Comparison table for same data size per request (Unit of time is ms)

Table1. Comparative result of load balancing algorithms for same data size of requests

	Hybrid	Round Robin	Throttled	ESCE
Overall Response Time (Avg.)	202.47	202.50	202.48	202.48
Data Center Processing Time (Avg.)	1.29	1.30		
Data Center Processing Time(Max)	4.02	7.26	4.52	5.76
Data Center Request Serving Time for DC2 (Avg.)	1.58	1.63		1.59
Data Center Request Serving Time for DC1 (Max)	2.31	2.88		2.50
Data Center Request Serving Time for DC2 (Max)	4.02	7.26	4.52	5.76

Fig.5. Simulation statistics for hybrid load balancing algorithm for different data size per request

Comparison table for different data size per request (Unit of time is ms)

Table2. Comparative result of load balancing algorithms for different data size of requests

	Hybrid	Round Robin	Throttled	ESCE
Overall Response Time (Avg.)	202.55	202.57		202.57
Data Center Processing Time (Avg.)	1.32	1.33		
Data Center Processing Time(Max)	3.87	8.38	4.26	7.27
Data Center Request Serving Time for DC1 (Avg.)	1.27	1.28		
Data Center Request Serving Time for DC2 (Avg.)	1.59	1.65	1.60	1.62
Data Center Request Serving Time for DC1 (Max)	2.31	2.88		2.50
Data Center Request Serving Time for DC2 (Max)	3.87	8.38	4.26	7.27

VI. CONCLUSION AND FUTURE WORK

Though some factors were considered in this work to check the efficiency of the proposed algorithm, still there is a lot to be done. The proposed algorithm can be compared with more available algorithms implemented in real environment. Cost could not be decreased in the proposed algorithm.

In future the work can be extended with scenario having more datacenters scattered around different locations, and trying to improve load balancing approaches according to different criterions like cost.

VII. ACKNOWLEDGMENT

The authors wish to express their heartfelt gratitude to Mr. Neeraj Sharma, Mrs. Urmila Ojha, Mr. Abhishek Singh, Mr. Yogesh Kumar, Ms. Rishika Jain for providing the facilities for the research and development work and for constant encouragement.

REFERENCES

[1] Ligang He, Deqing Zou, Zhang Zhang, Chao Chen, Hai Jin, Stephen A. Jarvis, Developing resource consolidation frameworks for moldable virtual machines in clouds, *Future Generation Computer Systems 32*, 2014,pp. 69–81.

[2] Jasobanta Laha, Rabinarayan Satpathy, Kaustuva Dev**,** Load Balancing Techniques: Major Challenges in Cloud Computing - A Systematic Review, *International Journal of Computer Science and Network, Volume 3, Issue 1*, February 2014.

[3] Gaochao Xu, Junjie Pang, and Xiaodong Fu, A Load Balancing Model Based on Cloud Partitioning for the Public Cloud*, IEEE Transactions on cloud computing*, 2013.

[4] Monir Abdullah, Mohamed Othman, Cost-Based Multi-QoS Job scheduling using Divisible Load Theory in Cloud Computing, *International Conference on Computational Science*, 2013.

[5] Dhinesh Babu L.D., P. Venkata Krishna, Honey bee behavior inspired load balancing of tasks in cloud computing environments, *Applied Soft Computing 13*, 2013, pp. 2292–2303.

[6] Ruben Van den Bossche, Kurt Vanmechelen, Jan Broeckhove, Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds, *Future Generation Computer Systems 29* ,2013, pp. 973–985.

[7] Jinn-Tsong Tsai, Jia-Cen Fang , Jyh-Horng Chou, Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm, *Computers &OperationsResearch40*, 2013, pp.3045–3055.

[8] Subasish Mohapatra, K.Smruti Rekha, Subhadarshini Mohanty, A Comparison of Four Popular Heuristics for Load Balancing of Virtual Machines in Cloud Computing, *International Journal of Computer Applications Volume 68– No.6,* April 2013, pp. 0975 – 8887.

[9] Uddalak Chatterjee, A Study on Efficient Load Balancing Algorithms in Cloud Computing Environment, *International Journal of Current Engineering and Technology*, December 2013.

[10] Namrata Swarnkar, Asst. Prof. Atesh Kumar Singh, Dr. R. Shankar, A Survey of Load Balancing Techniques in Cloud Computing, *International Journal of Engineering Research & Technology*, August 2013.

[11] Tushar Desai, Jignesh Prajapati, A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing, *International Journal of Scientific & Technology Research Volume 2, Issue 11*, November 2013.

[12] M.Aruna, D.Bhanu, R.Punithagowri, A Survey on Load Balancing Algorithms in Cloud Environment, *International Journal of Computer Applications, Volume 82 – No 16,* November 2013, pp. 0975 – 8887.

[13] MR.Manan D. Shah, MR.Amit A. Kariyani, MR.Dipak L. Agrawal, Allocation Of Virtual Machines In Cloud Computing Using Load Balancing Algorithm, *International Journal of Computer Science and Information Technology & Security, Vol. 3, No.1,* February 2013.

[14] Subasish Mohapatra, Subhadarshini Mohanty, K.Smruti Rekha, Analysis of Different Variants in Round Robin Algorithms for Load Balancing in Cloud Computing, *International Journal of Computer Applications, Volume 69– No.22,* May 2013, pp. 0975 – 8887.

[15] Shanti Swaroop Moharan, Rajadeepan D. Ramesh , Digamber Powar, Analysis of load balancers in cloud

computing, *International Journal of Computer Science and Engineering,Vol. 2, Issue 2*, May 2013, pp.101-108.

[16] Yatendra Sahu, R.K. Pateriya, Cloud Computing Overview with Load Balancing Techniques, *International Journal of Computer Applications, Volume 65– No.24*, March 2013.

[17] Meenakshi, Comparative Study of Load Balancing Algorithms in Cloud Computing Environment, *International Journal of Engineering Research & Technology, Vol. 2 Issue 10,* October 2013.

[18] Akshay Daryapurkar, Mrs. V.M. Deshmukh, Efficient Load Balancing Algorithm in Cloud Environment, *International Journal Of Computer Science And Applications Vol. 6, No.2*, Apr 2013.

[19] Anton Beloglazov , Jemal Abawajy, Rajkumar Buyya, Energy-aware resource allocation heuristics for efficient management of datacenters for Cloud computing, *Future Generation Computer Systems 28* ,2012, pp. 755–768.

[20] Loganayagi.B, S.Sujatha, Enhanced Cloud Security by Combining Virtualization and

Policy Monitoring Techniques, *The International Conference on Communication Technology and System Design* 2011.

[21] Brototi Mondal, Kousik Dasgupta, Paramartha Dutta, Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach, *C3IT*,2012

[22] Ajith Singh. N, M. Hemalatha, An Approach on Semi-Distributed Load Balancing Algorithm for Cloud Computing System, *International Journal of Computer Applications, Volume 56– No.12,* October 2012.

[23] Chenwei Jia, Qiang Liu, Ling Tong, Geographic Information Public Services Platform Based on Cloud Computing Model,*IEEE*,2011.

[24] Zhang Yan-hua, Feng Lei, Yang Zhi, Optimization of Cloud Database Route Scheduling Based on Combination of Genetic Algorithm and Ant Colony Algorithm, *Advanced in Control Engineering and Information Science*, 2011.

[25] Rajkumar Sharma, Priyesh Kanungo, An Intelligent Cloud Computing Architecture Supporting e-Governance, *17th International Conference on Automation & Computing*, 10th September 2011.

[26] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh, Christopher Mcdermid, Availability and Load Balancing in Cloud Computing, *International Conference on Computer and Software Modeling,*2011.

[27] Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, *IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, 2010.

[28] Sohan Singh Yadav, ZengWen Hua, CLOUD: A Computing Infrastructure on Demand, *2nd International Conference on Computer Engineering and Technology,IEEE*, 2010.

[29] Mazin Yousif, Cloud Computing - An IT Paradigm Changer, *ACS/IEEE International Conference on Computer Systems and Applications,* 2010.

[30] Ammar Khalid, Cloud Computing: Applying Issues in Small, *International Conference on Signal Acquisition and Processing*, 2010.

[31] Jianfeng Yang, Zhibin Chen, Cloud Computing Research and Security Issues, *IEEE*, 2010.

[32] Bhathiya Wickremasinghe, Rodrigo N. Calheiros, Rajkumar Buyya, CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications, *24th IEEE International Conference on Advanced Information Networking and Applications,* 2010.

[33] R.Jeyarani, N. Nagaveni, R. Vasanth Ram, Design and Implementation of an efficient Two-level Scheduler for Cloud Computing Environment, *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, 2010.

[34] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Rose, Rajkumar Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Wiley Online Library*, 24 August 2010.

[35] Manish Pokharel, YoungHyun Yoon, Jong Sou Park, Cloud Computing in System Architecture, *IEEE*, 2009.

[36] http://www.makeuseof.com/tag/cloud-computing-work-technology-explained/, accessed on 25/4/2014.

[37] http://www.katescomment.com/definition-of-cloud-computing-nist-g-cloud/, accessed on 25/4/2014

AUTHORS

1. Rajkumar Somani is working as an Associate Professor in Computer Science & Engineering Dept. at ITM, Bhilwara. He has done his B. Tech.,M.Tech, and Ph.D. He has 10 years of industrial experience and 18 years of teaching experience. His research areas include cloud computing, software engineering, ERP System, data mining and warehousing. He has presented various research papers at national and international levels.

2. Jyotsana ojha received her B. E. Degree in Computer Science & Engineering from University of Rajasthan, jaipur, India in 2008 and pursuing her M.Tech degree in Computer Science & Engineering from Rajasthan Technical University, Kota, India. She has experience in the field of teaching. Her area of interest is in website designing.

