Scalable and Trustworthy Load Balancing Technique for Cloud Environment

Anurag Jain [#1], Rajneesh Kumar [*2]
[#] Associate Professor, Computer Science Department
Chandigarh Engineering College, Landran, Mohali, Punjab, India
[1] er.anuragjain14@gmail.com
[*] Professor, Computer Science Department
M.M.E.C., M. M. University, Mullana, Ambala, Haryana, India
[2] drrajneeshgujral@mmumullana.org

***Abstract*— Cloud computing is the outcome of rapid growth of internet. Its future depends upon the optimum resource utilization, higher customer satisfaction and efficient installation of infrastructure. Due to ubiquitous, pay per usage, elasticity and lower installation cost many service providers are providing services through cloud. This has inundated the task at cloud service provider end. So, need of efficient and cloud compatible load balancing strategy has been felt. Load balancing is the major research area among the key challenges in the field of cloud computing. It involves mapping between task & resource in such a way that all resources are utilized in optimal manner. At the same time it is also desirable that load balancing approach must be scalable, fault tolerant and fault recoverable so that trust of cloud user will increase in the services provided by cloud service provider. Keeping all the desirable characteristics of cloud environment in consideration, authors have proposed weighted random walk based load balancing approach which not only increases the user trust in cloud services but also matches with the cloud essential characteristics. Weighted random walk based approach has implemented the biased random walk concept. Biasing has been implemented using weight of virtual machine which is based on virtual machine capacity and no of tasks in its queue.**

Keyword- Cloud Computing, Load Balancing, Biased Random Sampling, Scalable, Trustworthy, Reliability.

I. INTRODUCTION

After water, gas, telephone and electricity, next utility in our life is cloud computing. It is so because today every service provider is providing services through cloud. Before its emergence, all resources (hardware and software) for any services have to be purchased. But after cloud, instead of purchasing all the resources, we can use the resources through services provided by cloud service provider. Moreover, we can utilize these resources from anywhere, any time without any human intervention and we have to pay as much as we have utilized [1], [2]. When services are in the form of hardware then it is called Infrastructure as a Service. When cloud service provider is providing services in the form of environment where user can run their application then it is called Platform as a Service. When users are utilizing the application software available on a cloud then it is called software as a service [3]. In this way cloud computing has changed the paradigm of computing. Using the concept of virtualization, cloud service provider is providing the same services to multiple users at the same time on sharing basis. But this has raised the issue of load balancing and security on cloud service provider end [4], [5].

Paper is organized as follows: In section II authors have covered the basics of load balancing and security issues. Section III gives the information about the related work and basis of weighted random walk approach. In section IV, authors have discussed the weighted random walk approach with algorithm. Section V gives the analytical analysis of weighted random walk based load balancing approach. Probabilistic analysis of weighted random walk based load balancing approach has been given in section VI. In Section VII, authors have described the characteristics of the weighted random walk based load balancing approach. Conclusion and future scope has been given in section VIII.

II. LOAD BALANCING & SECURITY

Load balancing involves mapping between task and resources in such a way that all resources are utilized in optimal manner and services are provided according to service level agreement. Mapping can be done at task data centre level, task host level and task virtual machine level. If this mapping procedure is already fixed then this load balancing is called static load balancing. While if it is not fixed, it is changing itself according to present scenario, then it is called dynamic in nature. Also if mapping procedure is implemented at central level then it is called centralized approach. If it is implemented on multiple locations then it is called distributed approach. Also by implementing the mapping logic at different level, mixed approach can be implemented. Response time, data processing time, throughput, scalability and cost are some of the parameter on which load balancing technique can be analysed [6], [7].

In cloud environment, resources are shared between multiple users. Data is stored on cloud and user is not aware of actual storage location. Whenever user want to do change in data stored over cloud, either user has to download the data or user has to give access of data to the service provider. Due to all these factors there is always an issue of data security [8], [9].

The word Trust implies a work of reliance. It does not have any universally accepted definition. User does trust on service provider if service provider delivers the services as per commitment. It is a faith in the expertise and capability of other. If a system gives us unsatisfactory results in terms of performance, then we do less trust on that system [10], [11]. Improper mapping between task and resources may result in degradation of quality of service and non satisfaction of service level agreement. This results in decrement of users trust in cloud. So as a soft security service user trust can be enhanced through proper mapping of task and resource. If we deploy proper fault tolerance and fault recovery mechanism with mapping procedure, this will improve the quality of service and raise the users trust in capabilities of cloud.

III. RELATED WORK

O. A. Rahmeh et al. in [12] have proposed a load balancing approach for distributed network using biased random walk. Biasing was achieved on the basis of remaining resources. Through simulation they have shown their proposed approach as scalable and reliable. They have assumed their virtual machine were homogeneous in nature.

S. S. Manakattu et al. in [13] have proposed a load balancing algorithm which has implemented the biased random sampling. Parameters chosen by them for implementation of biasing were queue length and processing time. Through simulation, they have shown the proposed approach as efficient in terms of response time in comparison to other biased random sampling based approach.

M. Randles et al. in [14] have proposed biased random walk based load balancing approach for distributed environment. Biasing is achieved through job token which stores the information like no of steps in random walk and node id having maximum in-degree. Value stored in the token helps in the formulation of path during biased random walk. Experimentally they have shown their approach as efficient in terms of throughput.

V. Ariharan et al. in [15] have used the concept of token to implement load balancing through biased random walk. Token stores the information of load. Node having least load is used for task assignment.

D. Sumathi et al. in [16] have proposed a scheduling strategy for cloud environment which is based on trust based mechanism. Authors have modified the Heterogeneous Earliest Finish Time scheduling algorithm. Through proposed scheduling algorithm authors have tried to lower down the failure ratio which improves the users trust. Through simulation authors have shown the improved efficiency of proposed algorithm.

M. Firdhous et al. in [17] have proposed trust management system for cloud environment which is based on behaviour of honey bee. Authors have analysed many optimization problems which have been solved using honey bee algorithm. Authors have identified the similarity between those optimization problems and cloud environment. Then on the basis of the observation, author have concluded that honey bee algorithm can be used for solving the issue of trust management in cloud computing.

P. Gupta et al. in [18] have proposed a trust and reliability based load balancing algorithm for cloud environment. Authors have prepared a trust model and that model is used to propose the load balancing algorithm. Their trust model has used the performance of virtual machine as a parameter to calculate the trust value of the data centre. This trust value has been used to prioritize the data centre during task scheduling process. Authors have embedded the trust model with honey based load balancing algorithm.

IV. WEIGHTED RANDOM WALK BASED LOAD BALANCING APPROACH

Directed Graph G = (V, E) has been used to represent the load balancing scenario. V represents the set of available virtual machines (nodes) while E represents the set of edges used to connect nodes. Virtual machine in the system may have heterogeneous capacity. Based on the capacity of virtual machine, number of jobs in their queue and trust value of virtual machine, a weight is assigned to every machine. There will be an outgoing edge from node A to node B iff weight of node B is more in comparison to node A. When a task has been received at data center, a node has been selected randomly and biased random walk is organized from that node. After the end of biased random walk task is assigned to that node where it has ended. Now accordingly weight of that node is recalculated and edges connectivity of graph is updated. After task completion, task will depart from virtual machine and again weight of the node is recalculated and graph connectivity is updated accordingly.

Also if any virtual machine fails, then its weight is set to zero and edges connectivity of graph is updated accordingly. Jobs in the queue of failed node will be migrated to other nodes by following the weighted biased random walk approach. This results in the isolation of failed node which increases reliability of the system. Also when faulty node rectify its fault then during graph connectivity it gets become part of the virtual graph.

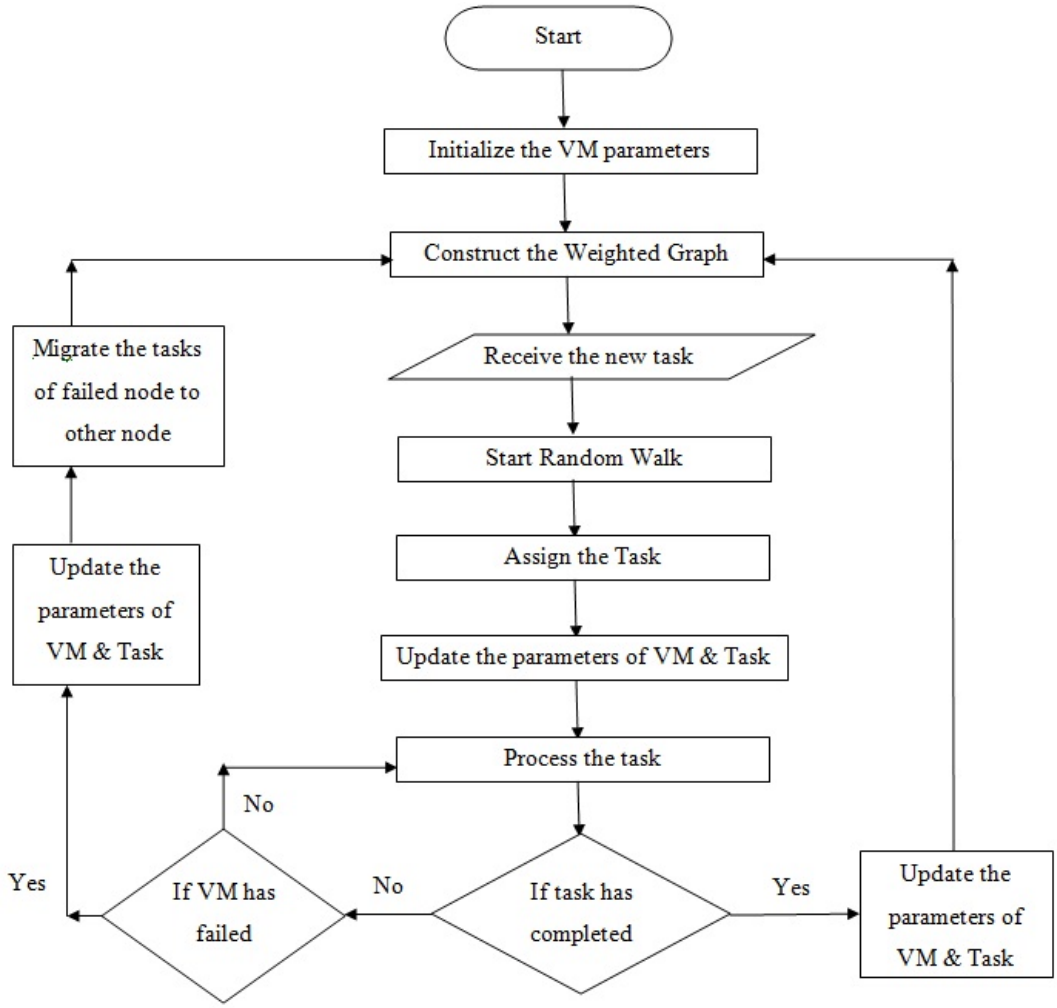

Fig. 1. Flowchart of weighted biased random walk approach

Algorithm Load_Balancing()

{

Step 1: Identify the available virtual machine and initialize their parameters (RAM, Storage, No of processors, Processing cores, no of tasks in queue, no of failure, trust value and weight).

Step 2: Construct the virtual graph according to weighted biased random walk approach.

Step 3: On receiving a task initialize the task parameters. (task id, size, task in time).

Step 4: Start biased random walk starting from any randomly chosen node where it ends assign task to that node.

Step 5: After task assignment, recalculate the weight of that node and update the graph connectivity accordingly. Update the parameters of virtual machine (no of tasks in queue, task id's and weight). Also update the parameters of task (VM id's, VM weight, walk length, position in queue and processing time).

Step 6: On completion of any task, recalculate the weight of that node where task has completed and update the connectivity of graph. Update VM parameters (no of tasks in queue, task id's and VM weight). Also update the parameters of task (task out time and task completion time).

Step 7: If a node fails then set its weight as zero, increment the no of failure, update the trust value and update the graph connectivity. Migrate the task to other active nodes according to the weighted biased random walk approach. Update the parameters of VM (no of failure, trust value, weight, no of tasks in queue, task ids).

}

Following information related to each task is maintained in data centre:

- Task ID- Auto incremented
- Task Size- Auto generated randomly
- Walk Length- Length of biased random walk before VM assignment to task
- VM id- id of VM to which task has been assigned
- Weight of VM- weight of VM to which task has been assigned
- Task In Time- Time at which task has been entered in data centre.
- Task Out Time- Time at which task has left data centre after processing
- Processing Time- Time duration during which task has been processed by processor
- Waiting Time- Time for which task has waited in queue and migration time (in case of failure)
- Task Completion Time- Task Out Time-Task In Time
- No of Migration- No of times task has moved from one VM to another
- Position in Queue- Position of task in the waiting queue

Following information related to each virtual machine is maintained in data centre:

- VM id- Id of VM, assigned in the beginning
- Task ID- Ids of task which are in the queue of VM
- Trust Value- Initial trust value for all the machines will be same. With every failure it will be decremented by some amount. Trust Value = Trust Value – (No of failure*Scaling factor)
- No of failure- No of failure will be zero in the beginning. Every time a VM fails, its value will be auto incremented
- Processing Capacity- VM processing power will depend up on the no of processors, processing cores, RAM size and Storage size. It is expressed in terms of MIPS.
- Weight- Weight= (VM Processing Power/No of tasks in Queue) + Trust Value, In the beginning Weight= VM Processing Power + Trust Value
- No of tasks in queue- With the arrival of every new task, it will be incremented and with the departure of every task, its value will be decremented

V. ANALYTICAL ANALYSIS

Let there are N numbers of virtual machines with different processing capacity in the data center. Then the virtual graph representing the present scenario of data center will have N nodes and at most N*(N-1)/2 no of edges. In degree and out degree of every node will vary between 0 to (N-1). Sum of in degree and out degree of any vertex of virtual graph will vary between 0 to (N-1) [19].

This virtual graph will convert in to a complex network. Through the analysis of structural and dynamic properties of graph, we can analyze the system. Using graph theory [20], the complex network representing virtual graph can be analyzed.

L. Lov'asz et al. in [21] has mentioned that probability of stopping a random walk at a node is directly proportional to in degree distribution of the node.

In the one extreme case when all the nodes have the same weight then number of edges in the virtual graph will be zero and maximum length of biased random walk will also be zero. In another extreme case when all the nodes have different weight then number of edges in the virtual graph will be N*(N-1)/2 and maximum length of biased random walk will be (N-1). So on average after N/2 number of moves during biased random walk, weighted biased random walk approach will be able to find the best virtual machine for task assignment. So time complexity of the weighted biased random walk approach to find the best virtual machine for task assignment will be O (n) which is linear in nature.

By assigning the job to the node having highest weight instead of last node in the random walk, it can be concluded that weighted biased random walk load balancing algorithm will perform better. This will also improves its performance in terms of parameter like response time, data processing time and scalability.

In degree of any node X refers to the number of nodes whose weight are less than the weight of X while out degree of any node X refers to the number of nodes whose weight are more than the weight of X. Whenever a node receives a new task, its weight get decreases and its out degree may increase and in degree may decrease. Also on completion of a task its weight will increases and its out degree may decreases and in degree may increases.

So all the time, in degree of every node will be close to average degree of node. So it can be concluded that graph will always approach to regular graph. This indicates the balance distribution of node.

VI. PROBABILISTIC ANALYSIS

Statistical model of the weighted biased random walk system will help in prediction of overall performance in terms of mathematical formula. Behavior of each node can be correlated with machine repairman model which is a special case of birth death process. Arrival of task on a node is considered as birth while task departure is considered as death. No of states will be finite. State of this birth death process will be labeled by the weight of the node. Label of the first state will be initial capacity of the node. Label of last state will be zero which indicates it can't accommodate any more tasks. With the arrival of every new task there will be decrement in the weight of the node and so there will be a state change accordingly. Also after the completion of a task weight of the node will increase and again state will change. With the arrival of more and more task, chances of arrival of new tasks will decrease. So forward transition probability will change accordingly. But backward transition probability will be dependent on the processing capacity of the node.

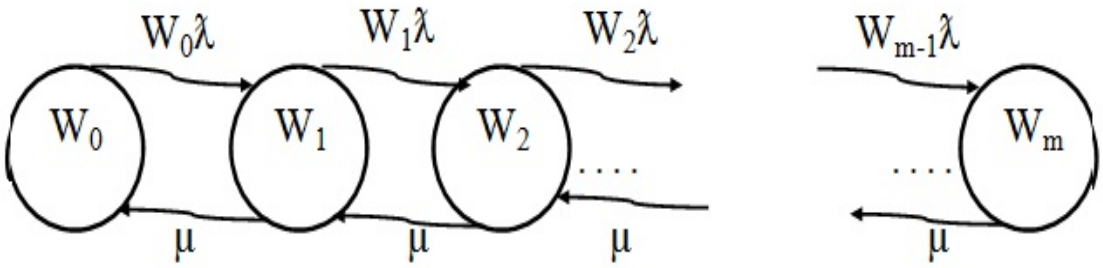

Fig. 2 State transition diagram of a node

W_i = weight of the node after arrival of ith task in the system / total weight of complete graph

W_m = indicate that weight of node is zero, now it can't accommodate any more task.

It has been assumed that tasks are arriving in the system according to Poisson distribution with mean task arrival rate λ. Any node in the virtual graph will be selected through biased random walk on the basis of weight of the node. So probability of moving in forward direction or birth will be $W_i\lambda$, where i denote the number of tasks already in the queue of that node. Also it has been assumed that resources among the tasks stored in the queue of a node will be scheduled according to space shared policy [23]. So the task departure rate or death rate μ will remain same throughout.

Steady state probability distribution of node will help us to calculate probability of k no of tasks assigned to a node in steady state. Let p_k denotes this probability. Then its value will be equal to

$$p_k = \left(W_{k-1}\,\lambda/\mu\right) * p_{k-1} \qquad 1 \le k \le m \tag{1}$$

Through backtracking equation (1) can be written as:

$$p_k = \left(W_{k-1}\,\lambda/\mu\right) * \left(W_{k-2}\,\lambda/\mu\right) \dots\dots\dots * \left(W_0\,\lambda/\mu\right) * p_0 \qquad 1 \le k \le m \tag{2}$$

Now equation (2) can be written as:

$$p_k = p_0 * \left(\lambda/\mu\right)^k (W_0 * W_1 * W_2 * \dots\dots * W_{m-2} * W_{m-1}) \qquad 1 \le k \le m \tag{3}$$

If we apply summation on both sides of equation (3) then left hand side of equation (3) will become 1(sum of probability of all tasks in steady state is one). So from this p0 can be calculated as follows:

$$p_0 = \frac{1}{1+\sum_{k=0}^{m}\left(\lambda/\mu\right)^k \prod_{k=1}^{m} W_{k-1}} \tag{4}$$

VII. CHARACTERISTICS OF WEIGHTED BIASED RANDOM WALK APPROACH

On the basis of analytical and probabilistic study of weighted biased random walk approach, it can be concluded that it possesses the following characteristics:

1. Distributed: As there is no centralized unit for task virtual machine mapping, so it can be concluded that weighted biased random walk approach is distributed in nature.

2. Dynamic: For task virtual machine mapping, weighted biased random walk approach is considering the remaining capacity as well as no of tasks in the queue. So it is dynamic in nature.

3. Trustworthy & Reliable: Weighted biased random walk approach after identifying faulty nodes, migrate its task to other working node & isolate the faulty node. Moreover trust value is also associated with each node.

With the failure of every node, trust value of every node will decrease and this will decrease its weight. After conversion of non responsive node in to responsive node, it can again consider it for task virtual machine mapping. So it can be concluded that weighted biased random walk approach is fault tolerance and also able to recover from failure. This makes proposed approach trustworthy & reliable.

4. Scalable: After analytical analysis of weighted biased random walk approach, it can be concluded that graph formed during implementation of weighted biased random walk approach will always be near to regular graph. It will not be affected by number of nodes and number of tasks. This makes weighted biased random walk approach scalable in nature.

5. Efficient: Due to following the concept of biased random walk, weighted biased random walk approach will always identify the best virtual machine available in data centre. Also, it has been identified through analytical analysis that on average N/2 no of moves will be used by weighted biased random walk approach to identify the best virtual machine. So, it can be concluded that it is efficient in nature in terms of parameter like response time and throughput.

VIII. CONCLUSION

In this paper, authors have proposed a load balancing approach which has used the concept of biased random walk. Random walk for searching virtual machine will stop at that virtual machine whose weight is highest among all the virtual machine. Analytical & probabilistic analysis of weighted biased random walk approach has been done. It has been found that weighted biased random walk approach is distributed, dynamic, scalable, efficient and trustworthy. Due to these characteristics proposed approach best suited for cloud environment.

As a future scope authors have planned to test the weighted biased random walk approach on cloud simulator & extend the weighted biased random walk approach during selection of data center also.

REFERENCES

[1] A.Jain and R. Kumar, “A Taxonomy of Cloud Computing,” International Journal of Scientific and Research Publications, vol. 4(7), pp. 1-5, July 2014.

[2] P. Sasikala, “Cloud Computing: Present Status and Future Implications,” International Journal of Cloud Computing, vol. 1(1), pp. 23-36, 2011.

[3] Q. Zhang, L. Cheng and R. Boutaba, “Cloud Computing: State-of-the-Art and Research Challenges,” Journal of Internet Services and Applications, vol. 1(1), pp.7-18, May 2010.

[4] M. Ahmed, A. S. Md. Raju Chowdhury, M. Ahmed and Md. M. H. Rafee, “An Advanced Survey on Cloud Computing and State-of-the art Research Issues,” International Journal of Computer Science Issues, vol. 9(1), pp. 201-207, January 2012.

[5] T. J. Lehman and S. Vajpayee, “We Have Looked At Clouds From Both Sides Now,” in Proc. SRII Global Conference (SRII),California, USA, pp. 342-348, March 30-April 2, 2011.

[6] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi and J. Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms," in Proc. Second Symposium on Network Cloud Computing and Applications (NCCA), London, England, pp. 137-142, December 3-4, 2012.

[7] A. Khiyaita, H. E. Bakkali, M. Zbakh and D. E. Kettani, "Load Balancing Cloud Computing: State of art," in Proc. 2nd National Days of Network Security and Systems (JNS2), Marrakech, Morocco, pp. 106-109, April 20-21, 2012.

[8] D. Chen and H. Zhao, "Data Security and Privacy Protection Issues in Cloud Computing," in Proc. International Conference on Computer Science and Electronics Engineering (ICCSEE), Zhejiang, China, pp. 647-651, March 23-25, 2012.

[9] K. Ren, C. Wang and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16(1), pp. 69-73, January February 2012.

[10] K. M. Khan and Q. Malluhi, “Establishing Trust in Cloud Computing,” IT Professional, IEEE, vol. 12(5), pp. 20-27, Sept.-Oct. 2010.

[11] K. Hwang and D. Li, “Trusted Cloud Computing with Secure Resources and Data Coloring,” IEEE Internet Computing, IEEE, vol. 14(5), pp. 14-22, Sept.-Oct. 2010.

[12] O. A. Rahmeh, P. Johnson and A. Talebbendiab, “A Dynamic Biased Random Sampling Scheme for Scalable and Reliable Grid Networks,” Infocomp Journal of Computer Science, vol. 7(4), pp. 1-10, 2008.

[13] S. S. Manakattu and S. D. Madhu Kumar, “An improved biased random sampling algorithm for load balancing in cloud based systems,” in Proc. International Conference on Advances in Computing, Communications and Informatics (ICACCI), Chennai, India, pp. 459-462, August 03-05, 2012.

[14] M. Randles, O. A. RAHMEH, P. Johnson and A. Taleb-bendiab, “Biased Random Walks on Resource Network Graphs for Load Balancing,” The Journal of Supercomputing, Springer, vol. 53(1), pp. 138-162, July 2010.

[15] V. Ariharan and S. S. Manakattu, "Neighbor Aware Random Sampling (NARS) algorithm for load balancing in Cloud computing," in Proc. IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, pp. 1-5, March 5-7, 2015.

[16] D. Sumathi and P. Poongodi, “An Improved Scheduling Strategy in Cloud Using Trust Based Mechanism,” International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 9(2), pp. 637-641, 2015.

[17] M. Firdhous, O. Ghazali, S. Hassan, N. Z.Harun and Azizi Abas, “Honey Bee Based Trust Management System for Cloud Computing,” in Proc. 3rd International Conference on Computing and Informatics (ICOCI), Bandung, Indonesia, pp. 327-332, June 8-9, 2011.

[18] P. Gupta, M. K. Goyal and P. Kumar, "Trust and reliability based load balancing algorithm for cloud IaaS," in Proc. IEEE 3rd International Advance Computing Conference (IACC), Ghaziabad, India, pp. 65-69, February 22-23, 2013.

[19] B. Bollobás, “Random Graphs,” Academic Press, London, England, 1985.

[20] P. Erdos and A. Renyi, “On random graphs,” Publication Mathematics, 6, 1959.

[21] L. Lov’asz and P. Winkler, "Mixing of Random Walks and Other Diffusions on a Graph," Surveys in Combinatorics, London Mathematical Society, Lecture Note Series, 1995.

[22] K. S. Trivedi, Probability & Statistics with Reliability, Queuing & Computer Science Applications, Prentice Hall, New Delhi, INDIA, pp. 383-388, 1982.

[23] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. Derose and R. Buyya, "Cloudsim: a Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," Software—Practice & Experience, Vol. 41(1), pp. 23-50, January 2011.

AUTHOR PROFILE

Anurag Jain is working as Associate Professor, Computer Science Department in Chandigarh Engineering College, Landran, Mohali. He is in teaching field since 2003. He received his B. Tech. & M.Tech. degree from Kurukshetra University, Kurukshetra, India, in 2003 and 2009, respectively. He has guided 7 M. Tech. students. He has about 15 publications in International Journals and Conferences. Currently he is pursuing Ph.D. from M, M, University Mullana Ambala in the area of cloud computing. His research interests are in the areas of load balancing & security in Cloud Computing. His email id is er.anuragjain14@gmail.com.

Dr. Rajneesh Kumar is working as Professor in the Department of Computer Science and Engineering, M.M Engineering College, M. M. University Mullana, Ambala. He obtained his PhD (Computer Science and Engineering) in 2012 under faculty of engineering, M.M University, Mullana, MTECH (IT) in 2007 from University School of Information Technology, GGSIP University Delhi and BE (Computer Science and Engineering) in 1999 from Punjab Technical University (PTU), Jalandhar (Punjab). He supervised 28 M. Tech, 1 M. Phil and currently supervising 8 PhD research scholars. He has about 40 publications in International Journals and Conferences. His research area includes Cloud Computing, Wireless Communications, Mobile Ad hoc & Sensor based Networks and Network Security etc. His email id is drrajneeshgujral@mmumullana.org.