International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:4, 2016

A Two Level Load Balancing Approach for Cloud Environment

Anurag Jain, Rajneesh Kumar

***Abstract*—**Cloud computing is the outcome of rapid growth of internet. Due to elastic nature of cloud computing and unpredictable behavior of user, load balancing is the major issue in cloud computing paradigm. An efficient load balancing technique can improve the performance in terms of efficient resource utilization and higher customer satisfaction. Load balancing can be implemented through task scheduling, resource allocation and task migration. Various parameters to analyze the performance of load balancing approach are response time, cost, data processing time and throughput. This paper demonstrates a two level load balancer approach by combining join idle queue and join shortest queue approach. Authors have used cloud analyst simulator to test proposed two level load balancer approach. The results are analyzed and compared with the existing algorithms and as observed, proposed work is one step ahead of existing techniques.

***Keywords*—**Cloud Analyst, Cloud Computing, Join Idle Queue, Join Shortest Queue, Load balancing, Task Scheduling.

I. Introduction

CLOUD computing is a new service delivery model over the internet. Its main characteristics which have increased its popularity are virtualization, scalability, ubiquitous and pay as per usage. Cloud services can be in the form of any application software, system software and hardware. These services can be delivered through public, private and hybrid cloud [1].

Dynamic nature of job arrival process may cause over utilization and underutilization of resources. This may result in performance degradation in terms of service level agreement. To get rid of this problem, an adaptive load balancing algorithm is required for cloud environment [2], [3].

The rest of this paper is organized as follows: Section II covers the fundamentals of load balancing. Introduction of simulator is given in Section III. Section IV includes literature review. Proposed load balancing approach is given in section V. Results and analysis are given in Section VI. Conclusion and enhancement scope is given in Section VII.

II. Load Balancing Basics

Load balancing means distribution of tasks among different available resources so that no one is over or underutilized. Resources can be data center, physical machine or virtual machine. Mapping of task with resources can be implemented at various levels. Mapping can be done between task and data center, task and host in a data center, task and virtual machine in a host. Load balancing can also be done through scheduling of tasks in a virtual machine or host or data center. Another way to do load balancing is migration of tasks from overloaded resource to under loaded resource. To handle this issue of load balancing in an efficient there is a need of efficient and effective load balancing strategy which is specifically suitable for cloud environment.

A. Load Balancing Types

Broadly load balancing strategies for cloud environment can be divided into following categories [4], [5]:

- Static Approach: This approach is suitable for homogeneous and non-dynamic environment. In this approach, algorithm is defined during design time and it remains same throughout. There is no scope of reconfiguration with the changing scenario. Static Load balancing algorithms assign the tasks to the nodes based only on the prior defined ability of the node to process new requests.
- Dynamic Approach: This approach considers the current parameters while assigning task to a node. It is more suitable for cloud environment. Such algorithms are hard to implement as they have to constantly monitor the nodes and task progress and take the decision based upon that.
- Adaptive approach: This approach also considers the current parameter while making a task assignment decision. It is different from dynamic approach in the sense that it not only changes the parameter of load balancing strategy but also do the changes in the algorithm with the change of system load. It is hardest to implement but it is most suited to cloud environment.

B. Metrics

Efficiency of a load balancing approach can be analyzed on the following parameters [6]:

- Response time: It shows the time taken by a scheduling approach to respond a task in the system. Its least value is desirable.
- Cost: It involves the migration cost, processing cost and storage cost. An efficient algorithm always tries to minimize this factor.
- Throughput: This parameter tells us about the number of tasks completed per unit of time. Its higher value is desirable.
- Scalability: This parameter shows how algorithm handles the variation in number of tasks. If algorithm handles the

Anurag Jain, Assistant Professor is with the Geeta Institute of Management and Technology, Kanipla, Kurukshetra, Haryana, INDIA (phone: +91-9466096567; fax: +91- 01744-279801; e-mail: anuragjain@gimtkkr.com).

Dr. Rajneesh Kumar, Professor is with the M. M. Engineering College, Maharishi Markandeshwar University, Mullana, Ambala, Haryana, INDIA (drrajneeshgujral@mmumullana.org)

jobs efficiently in such environment, then it is called scalable.

- Fault tolerance: It shows the algorithm's potential to handle the situation of fault and its recovery power from the failure.

III. Cloud Analyst Simulator

B. Wickremasinghe et al. [7] have given cloud analyst simulator for cloud environment. Basis of this simulator was their earlier given simulator cloudsim. But cloudsim does not have any graphical user interface. This limitation was overcome in Cloud analyst. Cloud analyst has a graphical user interface which makes it easy to configure the simulation environment. It can also generate the results in the form of graphs.

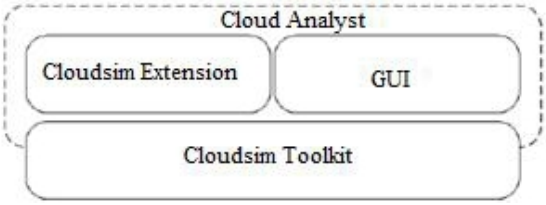

Fig. 1 Cloud Analyst Structure [7]

A. Components of Cloud Analyst

- VMLoadBalancer: This module plays a very crucial role as it simulates the various load balancing approaches which are used for task scheduling or task migration from one VM to another.
- Simulation: This component handles the management of various parameters related to simulation, generates & executes the simulation.
- UserBase: By generating the varying traffic as per the configuration of simulation parameter, this component simulates a user base.
- CloudAppServiceBroker: This module simulates the service brokers which controls the routing of traffic among user bases and data centers.
- Internet: This module simulates the Internet & simulates the traffic routing actions.
- Internet Characteristics: This component is responsible for management of internet characteristics during simulation. It includes the latencies & accessible bandwidths among regions and present performance level data for the data centers.
- Data Center Controller: This component manages the activities related to data centre.
- GUI Package: This package provides graphical user interface, which work as the front end manager for managing screen transitions, application, and additional user interface activities [7].

IV. Related Work

This section includes the literature review of load balancing approaches that have been merged to propose the new approach.

Yi Lu et al. [8] have discussed the Join Idle Queue (JIQ) scheduling approach for load balancing. Authors have implemented a two level scheduling. To realize the concept of two levels scheduling, authors have used the distributed scheduler. Number of schedulers is very less in comparison to number of virtual machines. Every scheduler will maintain a queue of idle virtual machines. On receiving a task, scheduler first consults its idle queue. If it finds any virtual machine which is idle, then it immediately assigns the task to that virtual machine and removes that virtual machine from its idle queue. If it does not find any idle virtual machine, then it randomly allots that task to any virtual machine. Virtual machine after job completion, update about its status to any of the randomly chosen idle queue associated with a scheduler. This approach has separated the task of discovery of idle servers from the task of job assignment to virtual machine.

Algorithm JIQ()
{
- Every Scheduler maintains the list of idle virtual machines in its idle queue.

While (there is a task received by cloud broker)
{
- Cloud broker forward the task randomly towards any of the scheduler.
- On receiving a task, scheduler checks its idle queue.

If (idle queue is not empty) then
{
- Scheduler removes the idle server from the queue and assigns the received task.

}
Else
{
- Scheduler assigns the task to any randomly selected virtual machine

}
If (any virtual server get idle) then
{
- Virtual server randomly selects scheduler & adds itself to the idle queue of that scheduler.

}
}
}

Varun Gupta et al. [9] have discussed the Join Shortest Queue (JSQ) scheduling approach for load balancing in distributed environment. This approach uses only single scheduler, which dispatch the task towards that virtual machine whose queue length is small. Scheduler maintains a VM allocation table which stores the queue length corresponding at each virtual machine. This helps scheduler to redirect the received task towards a suitable virtual machine. No queues are maintained at scheduler level. Queues are maintained only at virtual machine level.

Algorithm JSQ()
{
- Scheduler initializes the vm allocation table.

While (there is a task received by JSQ scheduler)
{
- Scheduler forwards the task towards that VM whose queue length is smallest & update VM allocation table.

If (any virtual machine completes the task) then
{
- Update the VM allocation table accordingly.

}
}
}

V. Proposed Work

A two level scheduler (JIJS) for load balancing in cloud environment has been proposed by integrating Join Shortest Queue approach and Join Idle Queue approach. JIJS has taken the characteristics of join idle queue and join shortest queue. It is distributed in nature as it uses multiple schedulers in place of one centralized scheduler. Idea of hybrid load balancing approach has been inspired from [10].

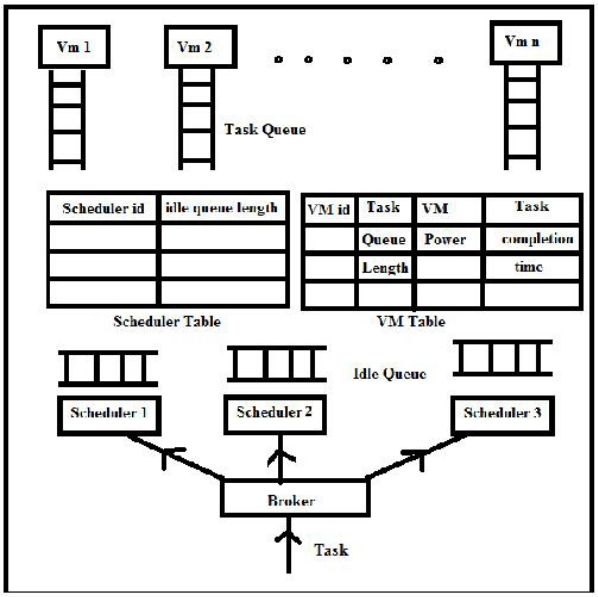

Fig. 2 JIJS Approach

A. Steps of Proposed Algorithm

- Step 1: Through scheduler table choose the scheduler whose idle queue length is largest, assign the task to idle virtual machine through the selected scheduler and update the scheduler table accordingly.
- Step 2: If idle queue of all scheduler is empty then Scheduler access the VM table and select that virtual machine whose queue length is shortest and assign the task to that virtual machine.
- Step 3: On completion of task, VM update the VM table. If any VM gets idle, then it adds itself to the idle queue of that scheduler whose queue length is shortest. Scheduler updates the scheduler table accordingly.

TABLE I
Different Data Structures Used in JIJS

Table Name	Who Can Update	Who Can Access	Information Stored
Idle Queue	Scheduler	Scheduler	Ids of idle virtual machines
Scheduler table	Scheduler	Broker, Scheduler and VM	Scheduler id, its idle queue length
VM table	Virtual Machine	Scheduler, Virtual Machine	VM id, its queue length

Algorithm JIJS()
{
- Populate the VM Table and scheduler table;

While (there is a task received by broker)
{
- Broker access scheduler table & it assign task to that scheduler whose idle queue length is largest;
- If idle queue length of all scheduler is null then it assigns task to any scheduler;
- On receiving a task, scheduler checks its idle queue;

If (idle queue is not empty) then
{
- Scheduler removes the idle VM from the idle queue;
- Assign task to removed VM;
- Updates the scheduler table and VM Table;
- Continue;

}
Else
{
- Scheduler accesses the VM table;
- Identify VM whose task length is shortest;
- Assign the task to the identified VM;
- Updates the VM Table;
- Continue;

}
}
}

VI. Results & Analysis

A. Simulation Configuration

- **Simulation Duration:** 10 hrs
- **Service Broker Policy:** Optimize Response Time
- **Cost Model:**
- Virtual machine usage cost per hour = $ 0.05
- Memory usage cost per second = $ 0.025
- Data storage cost on cloud per second = $ 0.05
- Data transfer cost per 1GB (from cloud to internet or vice versa) = $ 0.05
- **Grouping Model:**
- No of concurrent users from a single user base= 500
- No of concurrent requests a distinct application server instance can sustain= 50
- Length of executable instruction per request= 125 bytes
- **Resource scheduler policy to virtual machine:** Time-shared
- **Data Center hardware configuration:**

- No of processors on physical machine= 2
- Processing power= 75 MIPS
- Storage devices= 75GB
- Memory= 1GB
- Internal bandwidth= 5000 MBPS

- **Data Center virtual machine specification:**

- RAM = 512 MB
- Storage quota= 5 GB
- Architecture = x86
- Operating system = Linux
- Virtualization technique = Xen
- Bandwidth = 500 MBPS

- **User Base Specification:**

User Base	Region	No of requests per user per hour	Data size per requests in bytes	Peak hour start time	Peak hour end time
UB1	0	50	50	19	21
UB2	1	45	75	14	16
UB3	2	60	100	3	5
UB4	3	75	90	18	20
UB5	4	35	65	11	13
UB6	5	30	70	17	19

Fig. 3 User Base Specification [11]

B. Results

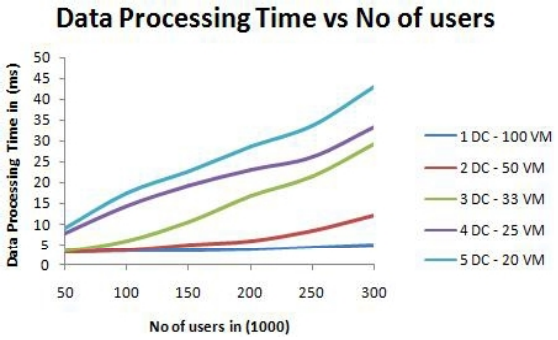

Fig. 4 Comparative Analysis of Data Processing Time with No. of Users for JIJS Approach

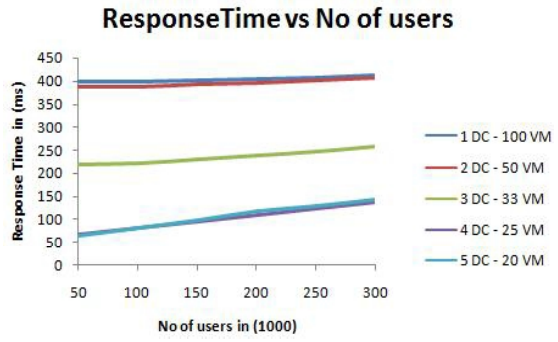

Fig. 5 Comparative Analysis of Response Time with No. of Users for JIJS approach

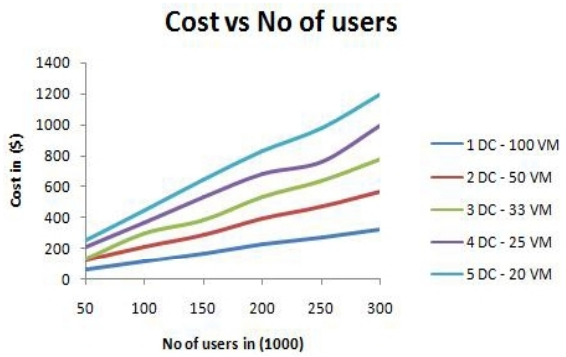

Fig. 6 Comparative Analysis of Cost with No. of users for JIJS approach

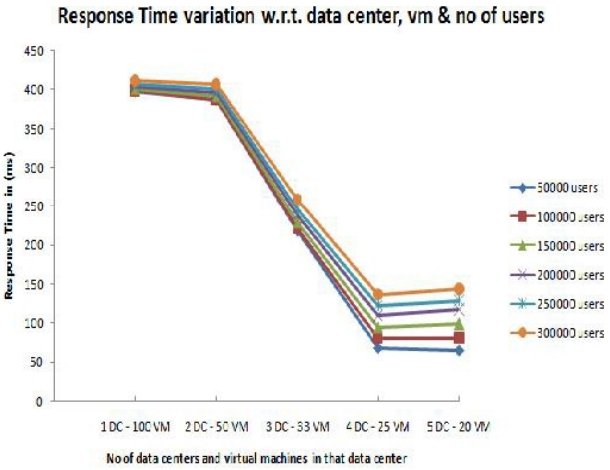

Fig. 7 Comparative Analysis of Response Time with Data Center Configuration for JIJS approach

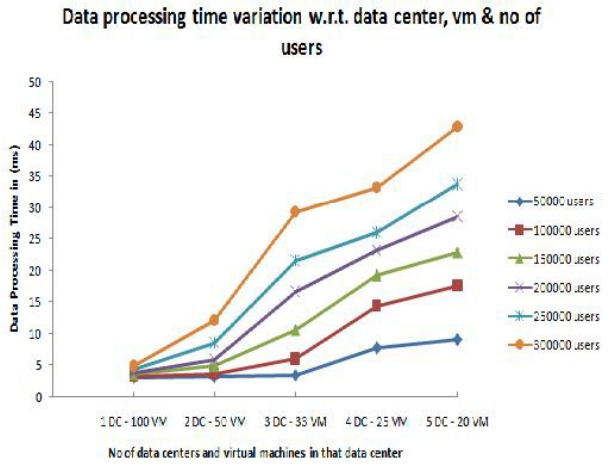

Fig. 8 Comparative Analysis of Data Processing Time with Data Center Configuration for JIJS approach

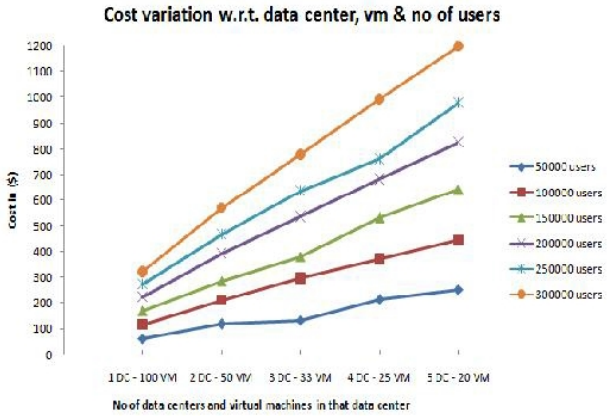

Fig. 9 Comparative analysis of Cost with Data Center Configuration for JIJS approach

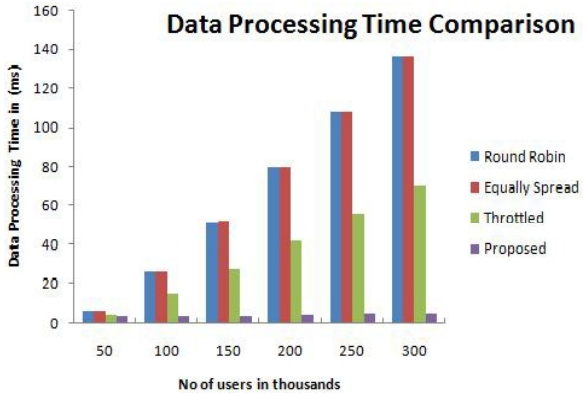

Fig. 10 Data Processing Time Analysis of Different Strategies

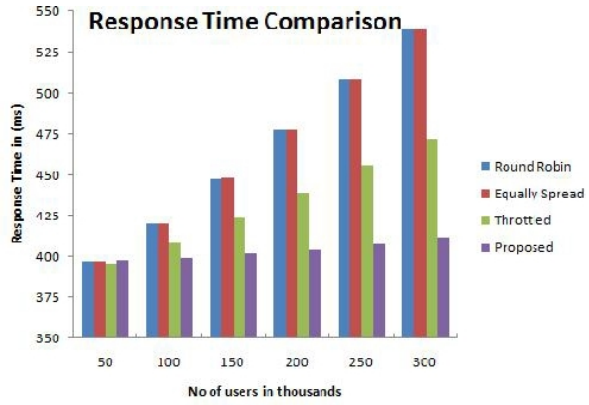

Fig. 11 Response Time Analysis of Different Strategies

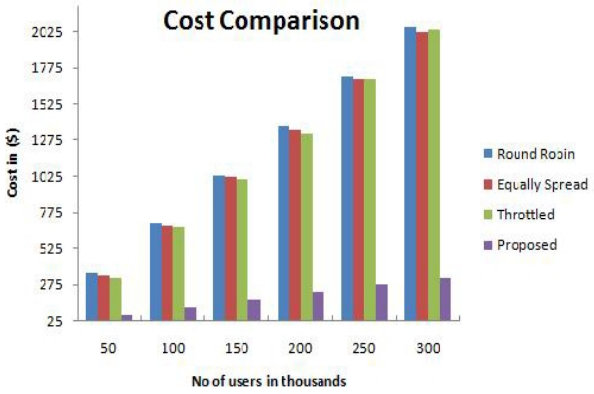

Fig. 12 Cost Analysis of Different Strategies

C. Analysis

TABLE II
ANALYSIS OF JIJS APPROACH ON VARIOUS PARAMETERS

Parameter	Analysis
Data Processing Time	➢ With increase of number of user data processing time increases. ➢ With the increase of data centre but same total number of VM, data processing time increases due to geographical distance factor.
Response Time	➢ With increase of number of user response time increases. ➢ With the increase of data centre but same total number of VM, Response time decreases.
Cost	➢ With increase of number of user cost increases. ➢ With the increase of data centre but same total number of VM, cost increases due to geographical distance factor.
Scalability	➢ From Figs. 4-9, it can be concluded that proposed approach behaves in the same way in terms of response time, data processing time and cost with the change in number of users, number of data centre and number of virtual machine. So proposed approach is scalable in nature.

VII. CONCLUSION & FUTURE SCOPE

A two level load balancing approach for cloud environment has been given by combining join shortest queue and join idle queue approach. Join idle queue approach is used at first level, while join shortest queue approach is used at second level. Proposed approach has been tested using cloud analyst simulator in many different environments. Experimental results have validated that proposed approach is best suited to cloud environment.

As a future scope, algorithm can be improved by embedding a prior testing module for load balancing, which test for the need of migration of task from one virtual machine to another. This testing mechanism will inform us about the overloading of virtual machine and data center in advance.

REFERENCES

[1] A. Jain and R. Kumar, "A Taxonomy of Cloud Computing", *International Journal of Scientific and Research Publications*, vol. 4, no. 7, pp. 1-5, July 2014.
[2] B. P. Rimal, C. Eumni and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems", *Fifth International Joint Conference on INC, IMS and IDC*, pp. 44-51, Aug. 2009.

[3] P. Sasikala, "Cloud computing: Present Status and Future Implications' *International Journal of Cloud Computing*, vol. 1, no. 1, pp. 23-36, 2011.
[4] A. Khiyaita, M. Zbakh, H. El Bakkali, and D. El Kettani, "Load balancing Cloud Computing: State of Art", *National Days of Network Security and Systems (JNS2)*, pp. 106-109, Apr. 2012.
[5] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi and J. Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms", *Second Symposium on Network Cloud Computing and Applications (NCCA)*, pp. 137-142, Dec. 2012.
[6] Amandeep, V. Yadav and F. Mohammad, "Different Strategies for Load Balancing in Cloud Computing Environment: A Critical Study", *International Journal of Scientific Research Engineering & Technology (IJSRET)*, vol. 3, no. 1, pp. 85-90, Apr. 2014.
[7] B. Wickremasinghe, R. N. Calheiros and R. Buyya, "Cloudanalyst: A Cloudsim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications", *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010)*, pp. 446-452, Apr. 2010.
[8] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. Larus and A. Greenberg, "Join-Idle-Queue: A Novel Load Balancing Algorithm for Dynamically Scalable Web Services", *Elsevier Science Publishers,* vol. 68, no. 11, pp. 1056-1071, Nov. 2011.
[9] V. Gupta, M. Harchol-Balter, K. Sigman and W. Whitt, "Analysis of Join-the-Shortest-Queue Routing for Web Server Farms", *International Symposium on Computer Modelling, Measurement and Evaluation*, *Elsevier Science Publishers*, vol. 64, no. 9, pp. 1062-1081, Oct. 2007.
[10] S. C. Wang, K. Q. Yan, W. P. Liao and S. S. Wang, "Towards a Load Balancing in a Three-Level Cloud Computing Network", *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, pp. 108-113, July 2010.
[11] http://www.internetworldstats.com as on Aug. 2015.

Anurag Jain is working as Assistant Professor & Head in Computer Science Dept. in Geeta Institute of Management and Technology, Kurukshetra. He is in teaching field since 2003. He received his B.Tech. and M.Tech. degrees from Kurukshetra University, Kurukshetra, India, in 2003 and 2009, respectively. He has guided 7 M.Tech. students. 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, privacy, and reliability in Cloud Computing. His email id is er.anuragjain14@gmail.com.

Dr. Rajneesh Kumar Gujral 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