ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

Linear Programming Approach for Load Re-Balancing in HDFS

Pravin S Patil[1], Prof R.R.Badre[2]

P.G. Student, Department of Computer Engineering, MIT Academy of Engineering, Alandi, Maharashtra, India [1]

Associate Professor, Department of Computer Engineering, MIT Academy of Engineering, Alandi, Maharashtra, India[2]

ABSTRACT: A cloud computing is a gripping technology. The distributed file system forms a central building block for cloud computing applications which supports Map-Reduce Programming paradigm. In such file system the nodes simultaneously serve computers in network and perform storage operations.

However, during processing the file gets partitioned into number of chunks so that Map Reduce task can be performed in parallel over the nodes. Nodes may fail when they are upgraded, replaced or added. Stakeholders perform file operation such as create, delete and append results in load imbalance in DFS. Uniform distribution of file chunks gets bit difficult in distributed file system. The centralized load balancing technique includes central load balancer which may develop bottleneck problem. Proposed system sheds light on load rebalancing problem in distributed file system and attempt to design a load rebalancing algorithm with linear technique approach to allocate file chunks appropriately to system considering low cost, less execution time. The system includes Hadoop Distributed File System for maintaining logs and data sets.

KEYWORDS: Cloud Computing, DFS(Distributed file system), Load Balancing ,Linear Programming, Hadoop Distributed File System.

I. INTRODUCTION

In distributed systems/cloud computing the necessary operations such as create, append, delete along with certain changes according to cloud notations are continuously performed. Cloud users share resources and information using cloud operations. The cloud is rather popular for features such as Scalability, On Demand Service, Pricing, Quality of Service etc. For storage purpose the default technologies such as Map-Reduce distributed file systems, virtualization are included in Cloud. The Distributed file system technology uses technique of creating chunks on cloud computing application based on the Map-Reduce programming paradigm which incorporates master-slave architecture, i.e Master which act like Name node and Slave which act like Data node. Master/Server takes big tasks , divides it into sub tasks and assigns it to worker node i.e. to multiple slaves to solve assigned tasks individually. Map-Reduce programming technique in Distributed File System partitions a large file into number of small chunks and allocates each chunk to different node to perform Map-Reduce function parallel over each worker node. e.g Consider word count application task which finds occurrences of each distinct word in large file. The submitted large file is partitioned in fixed-size chunks or parts and assigned to different cloud storage node. Big file is divided in fixed-size chunks or parts by system and assigned them to different cloud storage nodes. The storage node does its task by calculating occurrences of each distinct word by scanning and parsing its own chunk. Server/Master assembles result from each storage node to calculate the final result. In Distributed file system the load of each storage node is directly proportional to number of file chunks that node consists. In cloud systems rapid increase in storage and spread of network is the main issue for balancing load in large scale distributed systems. There is need to balance the load over multiple worker node to achieve system performance, resource utilization, response time and stability. The storage nodes, in cloud, number of files and assesses to that file increases then the central node (master in Map-Reduce) becomes bottleneck. The load imbalance problem can be solved by designing load balancing algorithm, in which storage nodes are structured over network based on the distributed hash table (DHT); each file chunk having rapid key lookup in DHTs, in that unique identifier is assign to each file chunk. The proposed system intents to reduce the movement cost which is caused by

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

load imbalance problem due to nodes used to maximize the network bandwidth. In proposed system the Submitter client application submits the task to server. After receiving of the task, server accepts the job/task along with volunteer parameters, task parameters distribute the task to volunteer clients/storage nodes. Algorithm like Linear Programming is used for load distribution based on current load that further submits to volunteer clients/storage nodes. The volunteer clients/storage node completes the assigned job and send the response to the server. Server intern assembles the result and send the reply back to Submitter client application.

II. RELATED WORK

"Load Rebalancing for Distributed File Systems in Clouds", Hung-Chang Hsiao, HaiyingShen[1]here authorhad proposed algorithm which is compared against centralized approach and a competing distributed solution presented in the literature. The future work will be implemented in Hadoop distributed file system incorporating cluster environment.

"A New Approach to Improve Load Balancing for Increasing Fault Tolerance and Decreasing Energy Consumption in Cloud Computing",Ali Moghtadaeipour, Reza Tavoli[2]here author presents a novel strategy to improve load balancing for increasing fault tolerance and reduce energy consumption by ranking the tasks and virtual machine in cloud computing by fuzzy method.

"Analytical Literature Survey on Existing Load Balancing Schemes in Cloud Computing", GarimaRastogi, Dr Rama Sushil[3]here author focuses on balancing the load in cloud environment, need of load balancing, existing literature on load balancing algorithms, and widely used performance metrics for load balancing and detailed analysis on the algorithms taken in the literature.

"A secured load balancing architecture for cloud computing based on multiple clusters", Mohamed Belkhouraf, Ali Kartit and Hassan Ouahmane, Mohamed Belkhouraf, Ali Kartit and Hassan Ouahmane[4]here author proposed on how the system address the subject of load balancing in cloud computing present a semi centralized and multi cluster architecture.

"Efficient Utilization of Virtual Machines in Cloud Computing using Synchronized Throttled Load Balancing", ShikhaGarg_, Rakesh Kumar Dwivediyand HimanshuChauhanz[5]here author presents an algorithm on load balancing to reduce the situation of overload or under load on virtual machines that leads to improve the performance of cloud substantially.

"Cluster Based Load Balancing in Cloud Computing",SurbhiKapoor, Dr.ChetnaDabas[6]hereauthor presentsCluster based load balancing which considers resource specific demands of the tasks and reduces scanning overhead by dividing the machines into clusters. The algorithm gives better results in terms of waiting time, execution time, turnaround time and throughput as compared to existing throttled and modified throttled algorithms.

"Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm",Reena Pan war, Prof. Dr.BhawnaMallick[7]the author presents a dynamic load management algorithm for distribution of the entire incoming request among the virtual machines effectively.

"Load Balancing in Cloud Computing Environment Based on An Improved Particle Swarm Optimization",Kai Pan ,Jiaqi Chen,[8]author presentsimproved particle algorithm to achieve resource load balancing optimization in the cloud environment.

"Load Balancing Job Assignment for Cluster-Based Cloud Computing",Yean-Fu Wen and Chih-Lung Chang[9]the author proposed work which combines the characteristics of cloud with Grid Computing to achieve load-balancing mechanism. The system contributes to adopt queuing model to calculate the round-trip time and nodal processing to evaluate the end-to-end delay.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

"Cloud Light Weight: a New Solution for Load Balancing in Cloud Computing", MohammadrezaMesbahi, Amir MasoudRahmani, Anthony Theodore Chronopoulos[10]here author presents a new load balancing method, as Cloud Light Weight (CLW), which balances the Virtual Machines'(VM) workload in cloud computing DC, but it also assures QoS for users.

III. EXISTING SYSTEM

The central node in distributed file system, become a performance bottleneck because of linear increase in storage nodes, files and number of accesses to file. The dependence of central node emerge with bottleneck problem results in load imbalance issue which possess heavy load.

IV. PROPOSED SYSTEM

The proposed system implements load re-balancing system in cloud using linear programming. The server incorporates mechanism to calculate the load of Ad-Hoc volunteers. It will decide how much load should be given to each temporary data nodes. The server's smart linear programming while run-time will automatically divide the time taking task i.e image file in chunks and transmit the load to the data nodes / clients who are available, this achieves uniform distribution of chunks and reduces overall system execution time. In turn the server will have fine knowledge of how much load should be passed to each available data node improves the utilization of hardware resources. Parallelyhadoop distributed file system maintains logs and data sets efficiently.

V. SYSTEM ARCHITECTURE

The system compose of three different entities like Submitter Client Application (Submits task to server), Cloud Server (For load balancing) and Ad-Hoc Volunteer client (Process the task and submit response to server).

Fig 1 shows Architecture of Proposed System.

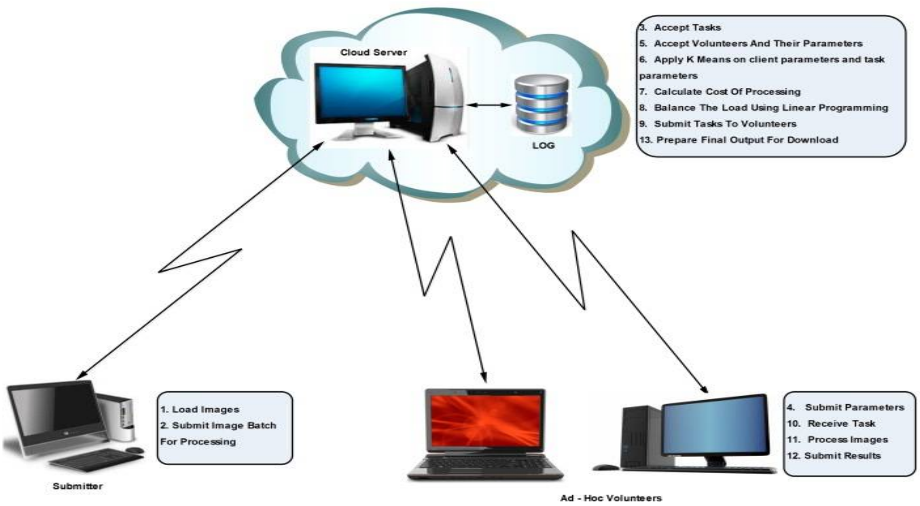


Fig 1: Architecture of Proposed System

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

Client/Submitter Application (Which will be responsible to Submits task to server)

- The task chosen istime consuming which actually takes time to execute , the submitter application is responsible to submitsImage Batch toserver for processing.

Cloud Server (For load balancing)

- Once accepting time taking task , the server consequentially accepts volunteers and their parameters. The Server applies K-means on client parameters and task parameters. The processing cost is calculated. The Linear Programming technique balances load efficiently after then the task is submitted to the volunteers appropriately. Lastly server prepares final output for Download.

Volunteer client (Process the task and submit response to server)

- The volunteer is responsible to submit parameters to server. It receives task from server. Once image gets processed the generated result is submitted back to the Cloud Server.

Task Submitter

- The task submitter is responsible to submit the task to server. There can be any 'n' number of submitter. The cloud is capable of executing all of them. Hence there is no limitation on number of submitters.

The Fig.2 shows detailed Architecture of the system.

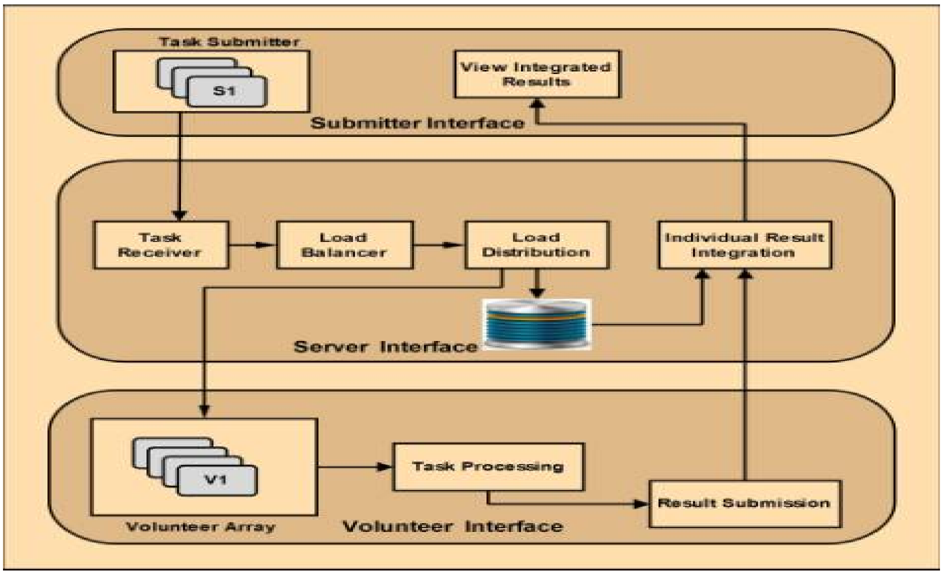


Fig.2: Detailed Architecture of Proposed System

Task Receiver

- Task receiver is responsible for receiving and forwarding task to load balancer. In proposed system the task taken is image on which basic reprocessing is performed. Consider I←{I1,I2,I3…In}, Where I is set of image on which pre-processing shall be performed.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

Load Balancer

- After receiving task the load balancer divides it in chunks and distribute them among the number of volunteer. If I1 is an input image of size 250 X 250 where we have total number of bytes as 62,500 and we have for example 3 volunteers so image shall be breaked into chunks as I1C1, I1C2, I1C3. After this a load balancing algorithmic equation is applied based on which image is passed to load distributor.

Load Distribution Module

- Load distribution is responsible to forward the image to the volunteer clients based on parameters calculated from load balancer.

Volunteer Array

- There may be any number of 'n' volunteer in the network. The task received from load distribution shall be processed here. For example assume the task in pre-processing of image in Gray Scale. The output image is send back to server via Result Submission.

Individual Result Integration

The braked image in chunks is to be integrated together.

VI. ALGORITHM FOR PROPOSED SYSTEM

Inputs

Set Of chunks c1,c2,......cn

Fixed Constants

chunk cost m> Cost Per Chunk for Low Configuration PC Devices

chunk cost c> Cost Per Chunk for High Configuration PC Devices

chunk time m> Time Per Chunk (In CPU Cycles i.ein Milliseconds) for Low Configuration PC

chunk time c> Time Per Chunk (In CPU Cycles i.ein Milliseconds) for High Configuration PC

TotalChunk > Total number of Chunks of all files

According to Linear Programming Technique :

Aim :To Find Chunk m and Chunk c

Chunk m→ Total Chunk to be processed by mobile devices

Chunk c→ Total Chunk to be processed by computers

Step 1: To find TotalChunk = Chunk m + Chunk c

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

Step 2 : To find Minimize Total Cost (TC)

i.eMin(TC)

so,

TC = (Chunk m * ChunkCost m) + (Chunk c * ChunkCost c)

Step 3 : To find Minimize Total Time (TT)

i.eMin(TT)

so, TT = (Chunk m * ChunkTime m) + (Chunk c * ChunkTime c)

VII. EXPERIMENTAL ANALYSIS AND RESULT

The system is implemented in small scale cluster environment which consists of Client/Submitter Application (Which is responsible to Submits task to server), Cloud Server(For load balancing) and Volunteer client(Process the task and submit response to server).

Table 1 shows the Ad-Hoc Volunteer name, the RAM size,their initial cost before processing any image task and the clustering details as how similar volunteers are clubbed on same cluster.

ID	VOLUNTEER NAME	VOLUNTEER RAM	VOUNTEER COST
1	vol1	4096	1
2	v2	2048	0
3	v3	8192	2
4	vol4	2048	0
5	v5	8192	2
6	h1	4096	1
7	h2	4096	1
8	h3	4096	1

Table 1: Submitter/ Volunteer Information and Cluster Details

Table 2 shows the Ad-Hoc client name, Cluster cost, the received job time, Job processed time, total time required to process job, the Job Count received by each client, cost required to process each job. along with all the processor information i.e total job processed , total cost required to process job along with memory and CPU utilized by each volunteer.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

PROC ID	CLIENT NAME	CLIENT COST	GETJOBTIME	PRCOCESS...	TOTAL REQ ...	JOBSCOUNT	COST PER J...
1	h1	1	1480858180...	1480858189...	9	1	9.0
2	h2	1	1480858180...	1480858290...	110	1	110.0
3	h1	1	1480858181...	1480858189...	8	1	8.0
4	h2	1	1480858181...	1480858290...	109	1	109.0
5	h1	1	1480858182...	1480858189...	7	1	7.0
6	h2	1	1480858182...	1480858290...	108	1	108.0
7	h1	1	1480858183...	1480858189...	6	1	6.0
8	h2	1	1480858183...	1480858290...	107	1	107.0
9	h1	1	1480858184	1480858189	5	1	5.0

Table 2:Job Processing Information

Table 3. shows all the processor information i.e total job processed , total cost required to process job along with memory and CPU utilized by each volunteer.

PROCESSOR	TOTAL PRO...	TOTAL COST	MEMORY UT...	CPU UTILIZE
h1	10	45	1882	64
h2	37	2147	1924	64
h3	10	44	2044	66

Table 3: Total Job Processing Information

Table 4 shows details of Ad-Hoc volunteer's name, total jobs processed by each volunteer, total cost required to process the jobs, the utilization of RAM and CPU to process respective jobs.

PROCESSOR NAME	TOTAL JOBS	TOTALCOST	MEMORY UTILIZE	CPU UTILIZE
h1	10	45	1882	64
h2	37	2147	1924	64
h3	10	44	2044	66

Table4: All Processor Information

Table 5 displays Job Submitter information i.e Job ID, image name, image size, send time of image task, the received time of image task, total time required to process whole image by Linear Programing Algorithm, time required by FIFO technique to process same image task along with the total cost required to process all jobs.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

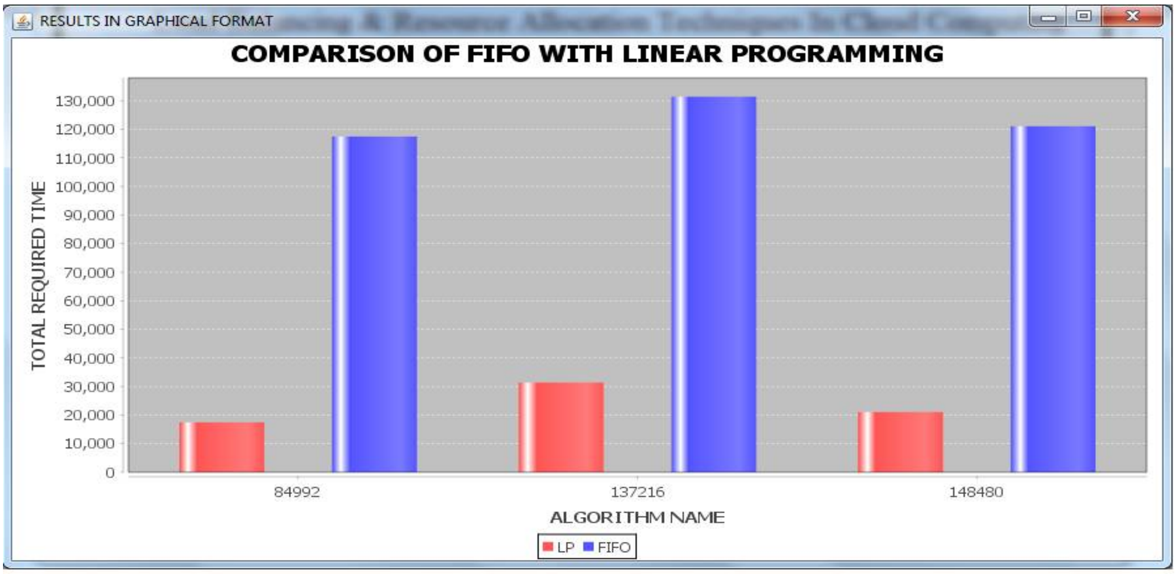
Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

JOBID	JOBNAME	JOB SIZE	JOBSENDTIME	JOBRECIVETI...	TOTAL TIME ...	FIFO	COST
1	DSC_8522.JPG	84992	14808581729...	14808581903...	17464	27464	0
2	DSC_8395.JPG	137216	14808582088...	14808582402...	31404	41404	0
3	DSC_8404.JPG	148480	14808582713...	14808582924...	21068	31068	2236

Table 5: Submitter Information

GRAPH 1 The graph projects time required to process same image task by Linear Programming Algorithm compared with FIFO technique.



Graph 1: Comparison of FIFO with Linear Programming Technique

VIII. CONCLUSION

The proposed system attempts to resolve load re-balancing problem in distributed file systems specialized for large-scale, dynamic and data-intensive clouds by allocating the chunks of files as uniformly as possible among the nodes such that no node manages an excessive number of chunks. Particularly, system implements linear algorithm/technique which will exhibit a fast convergence rate along with fine optimized results and increases system accuracy by 62% in comparison with FIFO technique.Parallely system hasincorporated Hadoop Distributed File System for maintaining logs and data sets.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

REFERENCES

[1] Hung-Chang Hsiao, HaiyingShen,"Load Rebalancing for Distributed File Systems in Clouds" IEEE, TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 5,2013.
[2] Ali Moghtadaeipour, Reza Tavoli,"A New Approach to Improve Load Balancing for Increasing Fault Tolerance and Decreasing Energy Consumption in Cloud Computing" , IEEE, 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), 2015.
[3] GarimaRastogi, Dr Rama Sushil," Analytical Literature Survey on Existing Load Balancing Schemes in Cloud Computing" ,IEEE, International Conference on Green Computing and Internet of Things (ICGCIoT), 2015.
[4] Mohamed Belkhouraf, Ali Kartit and Hassan Ouahmane, Mohamed Belkhouraf, Ali Kartit and Hassan Ouahmane,
" A secured load balancing architecture for cloud computing based on multiple clusters" IEEE, International Conference on Cloud Technologies and Applications (CloudTech), 2015.
[5] ShikhaGarg_, Rakesh Kumar Dwivediyand HimanshuChauhanz," Efficient Utilization of Virtual Machines in Cloud Computing using Synchronized Throttled Load Balancing" IEEE, 1st International Conference on Next Generation Computing Technologies (NGCT), 2015.
[6] SurbhiKapoor, Dr.ChetnaDabas," Cluster Based Load Balancing in Cloud Computing" IEEE, Eighth International Conference on Contemporary Computing, 2015
7] Reena Pan war, Prof. Dr.BhawnaMallick," Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm" IEEE, International Conference on Green Computing and Internet of Things (ICGCIoT), 2015
8] Kai Pan ,Jiaqi Chen," Load Balancing in Cloud Computing Environment Based on An Improved Particle Swarm Optimization",IEEE, 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2015.
9] Yean-Fu Wen and Chih-Lung Chang," Load Balancing Job Assignment for Cluster-Based Cloud Computing" ,IEEE, Sixth International Conference on Ubiquitous and Future Networks (ICUFN), 2014.
10] MohammadrezaMesbahi, Amir MasoudRahmani, Anthony Theodore Chronopoulos," Cloud Light Weight: a New Solution for Load Balancing in Cloud Computing" IEEE, International Conference on Data Science & Engineering (ICDSE), 2014.

BIOGRAPHY

Mr. Pravin S Patil, M.E Student, Dept of Comp Engg, MIT,AOE,Alandi(D),Pune. Received B.E. degree in Computer Engineering from North Maharashtra University, Jalgaon, India, in 2007. Interesting Area is Cloud Computing.

Prof. R.R.Badre received her M.E. degree in Computer Science and Engineering from Shivaji University, Kolhapur. She is currently working as an Associate Professor in the Department of Computer Engineering in MIT Academy of Engineering, Pune. She has published more than 8 papers in both International journals and conferences. She is also a member in Indian Society of Technical Education.