Load Balancing In Distributed File Systems Using Clouds

[1]Pravin S. Patil, [2]Aniruddha P. Kshirsagar, [3]Rohini B.Kokare, [4] Ashvini Nikhade
[1]Assistant Professor, [2]Assistant Professor, [3]Assistant Professor

Abstract: Distributed File System constructs essential blocks in distributed computing dependent on Map Reduce programming overview. The nodes perform computation and storing operations. The input file is divided into specific amount of pieces. Generally, the nodes tends to fail when up gradation, replacement and addition operations take place in distributed file system. Due to this operations the files parallel are automatically modified which lets in load imbalance problem, proper distribution of file chunks are not possible in cloud DFS. In the server the load balancer arises with bottleneck issue. The system proposed framework sheds light on load re-balancing problem caused in DFS frameworks particular for designating the file chunks as consistently as conceivable among the storage nodes with the end goal that none of the storage node deals with an unreasonable number of file chunks. Specifically, the proposed system will include linear algorithm which will have a quick convergence rate. The proposed system will be including HDFS to maintain logs and set of data and will be analyzed by using big cluster.

Keywords: DFS(Distributed file system),HDFS(Hadoop Distributed File System), Linear Programming, Load Balancing

1. Introduction

Many numbers of computers connect with each other in cloud computing network. The cloud operates on internet and is responsible for various operation like delete, create, append and replace of nodes. The IT users exchange the information, share resources with each other under cloud umbrella. The cloud characteristics are, User Centric, Platform Independent, Scalability, On Demand Service, Powerful and Versatility. Cloud incorporates Map-Reduce programming, Virtualization and distributed file systems for the information and data storage reason. An established model, for example, Distributed file system utilizes procedure of creating file chunks on distributed cloud computing operation dependent on the Map-Reduce paradigm technique. Map-Reduce includes slave-master architecture i.e Slave will work as Data node and Master will work as Name Node. The Name node i.e Master node is responsible to divide the large task into multiple number of chunks and assign the task/work to slave to resolve the problem separately. The Map-Reduce programming paradigm in DFS breaks the big file or given task into number of pieces so called as chunks and give each chunk separately to the performing node on basis of MapReduce function running parallel over each storage node. By considering example of word count application which try to find the occurrences of every distinct node or machine in big file task. The system is responsible to break a large file into equal pieces or chunks by assigning those chunks to the network of nodes under cloud umbrella. The master node identify the presence of each distinct word by applying technique of scanning parsing of its own chunks assigned. The Master node accumulates the result from each node attached in cloud to calculate and find the final result. The load of each node is decided with assumption of the file chunks allotted to it in runtime. We can say that number of file chunks are directly proportional to load of each storage node. The cloud system or DFS in cloud is progressively inclined to issue that a frameworks increment away and arrange the load balancing issue immerges. There ought to be some method to adjust the load balance over various hub or frameworks to improve framework execution, asset usage, reaction time and solidness in the system. As number of storage increases in nodes i.e number of files, assesses to that file increase then middle or central node (Master Node which is implementing Map-Reduce) is facing problem of bottleneck. The solve the load imbalance problem there should be technique designed to eliminate load balancing problem by finding appropriate load re-balancing algorithm. The storage nodes are based on the structured over the network based on DHT i.e Distributed Hash Table, each chunk is looked up by rapid key in DHT table, which is tagged by the unique identifier. The proposed system targets to minimize the movement cost which is caused by load imbalance problem due to nodes which uses maximum network bandwidth. In the system the client submitter application will try to submit the task to server. After receiving the task the server or master node will divide the task to volunteer clients or storage nodes. The linear programming algorithm may be used for load distribution which is based on current load which will be given to volunteer client/storage node. The clients/storage nodes also spelled as volunteer will try to complete the task and send the response or result to the server. The master of server will assemble the result and will send the reply back to Submitter client application gradually.

2. Need of Project

The proposed system will focus on upgrade for Resource Utilization, perform Optimized Scheduling with the objective that End customer can get Cloud Cost Justification on big cluster.

3. Related Work

Survey of Different Load Balancing Approach-Based Algorithms in Cloud Computing: A Comprehensive Review[1] The paper focuses on many number of load balancing algorithms with pros and cons. According to survey there is enough scope for researchers to build a good load balancing algorithm for cloud computing environment.

Load Balancing Tools and Techniques in Cloud Computing: A Systematic Review[2] In this paper, author have completed an review of various existing load balancing techniques, tools and methods pursued by their near investigation dependent on various parameters. After this near investigation, it is presumed that different procedures spread distinctive parameters to break down the techniques. However, none of existing method has been viewed as every one of the parameters as uncovered by work. Some other vital parameters likewise can be considered like carbon emanation, vitality utilization, and knowledge for green and smart burden balancer for distributed computing condition.

An Advanced Load Balancing Strategy For Cloud Environment[3] A load balancing technique proposed in this paper, which combined with the limit window methodology and propelled AR expectation demonstrate. Trials demonstrated that our methodology tackle the incessant relocation issue, yet additionally improved the dependability of the server farm. In future, researcher might want to investigate different issues in this system, for example, appropriated reaction, quick reaction time, and dealing with error.

Load Balancing in Cloud Computing:Challenges &Issues[4] As per author Load adjusting is significant test (issue) in cloud registering. The key point of load adjusting is to fulfill client's needs by appropriating outstanding burden among different hubs in framework and boost asset use and improves framework execution. So effective burden adjusting is indispensable for framework execution, asset use, dependability, amplifies the throughput and limits the reaction time that are the primary goals of this paper. To adjust the heap among various hubs in framework, there are a few burden adjusting calculations could be presented. This paper exhibits the outline of distributed computing, distributed computing engineering, virtualization, load adjusting and a few difficulties identified with adjusting load in distributed computing.

A Survey of Load Balancing Challenges in Cloud Environment[5] In this paper, the author sheds light on different kinds of LB calculations which help to adjust the heap. The essayist attempts to audit different research papers that are identified with load adjusting calculations. In this exploration paper, creator talk a few calculations like Ant Colony Optimization, Max-Min,Round-Robin, Biased Random Sampling and Min-Min calculation so arriving at resolution which strategy is best to adjust the load.

Load Balancing in Cloud Data Center Using Modified Active Monitoring Load Balancer[6] In this paper author stay away from two circumstances by disseminating the load among the virtual machines in a suitable way dependent on need, state and memory usage. With the assistance of the proposed calculation the heap of the client solicitations can be conveyed among the virtual machines effectively. Not withstanding, the proposed calculation is productive for client demands load appropriation among the virtual machines however it doesn't think about the virtual machines unwavering quality and vitality mindfulness hence the future work centers around load distribution of clients ask for alongside VM Reliability and VM vitality mindfulness.

An in-depth analysis and study of Load balancing techniques in the cloud computing environment[7] The paper focuses around broad investigation through the execution of the two load adjusting calculations to be specific Max-Min and Min-Min dependent on our chose cloud condition. The aftereffect of our assessment demonstrates that the Max-Min performs superior to Min-Min regarding make length. In any case, there are different works of load adjusting in cloud condition which demonstrates that Min- Min beat the Max-Min calculation. The two calculations have their very own advantages and disadvantages, where relying on the cloud condition one beats the other. On the off chance that the quantity of lighter errands dwarfs the heavier assignments, at that point Max-Min performs well superior to the Min-Min regarding asset usage and make range despite what might be expected if there are numerous heavier undertakings it results in Min-Min to perform superior to Max-Min. So author concludes that the execution of burden adjusting in cloud doesn't rely on any calculations however it is simply founded on the cloud condition picked. Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm[8] In present paper, a dynamic calculation is proposed which will deal with the heap approaching by concentrating on their present status at cloud let for all free VMs to be utilized at demand task and will take more demands that are dynamic in nature. In future on the off chance that both dynamic and static burden is to be blended, at that point the this calculation can be improved adequately and furthermore by consolidating the ideal models of parallel and superior processing reaction time and use of VMs might be additionally enhanced.

4. Problems in existing system

Gradual and linear increase in the storage nodes, simultaneous increase and accesses to file the central node faces the problem of bottleneck, which is not efficient to accommodate large number of file access due to Map reduce program paradigm. The direct dependence on central node come up with load imbalance problem which possess heavy load

5. Proposed System

The proposed framework will comprises of three elements like the Submitter Client Application (Submits job to server), Server (For load adjusting) and Volunteer client(Process the assignment and responsible to submit the response to server).Fig 1 projects block diagram for proposed system

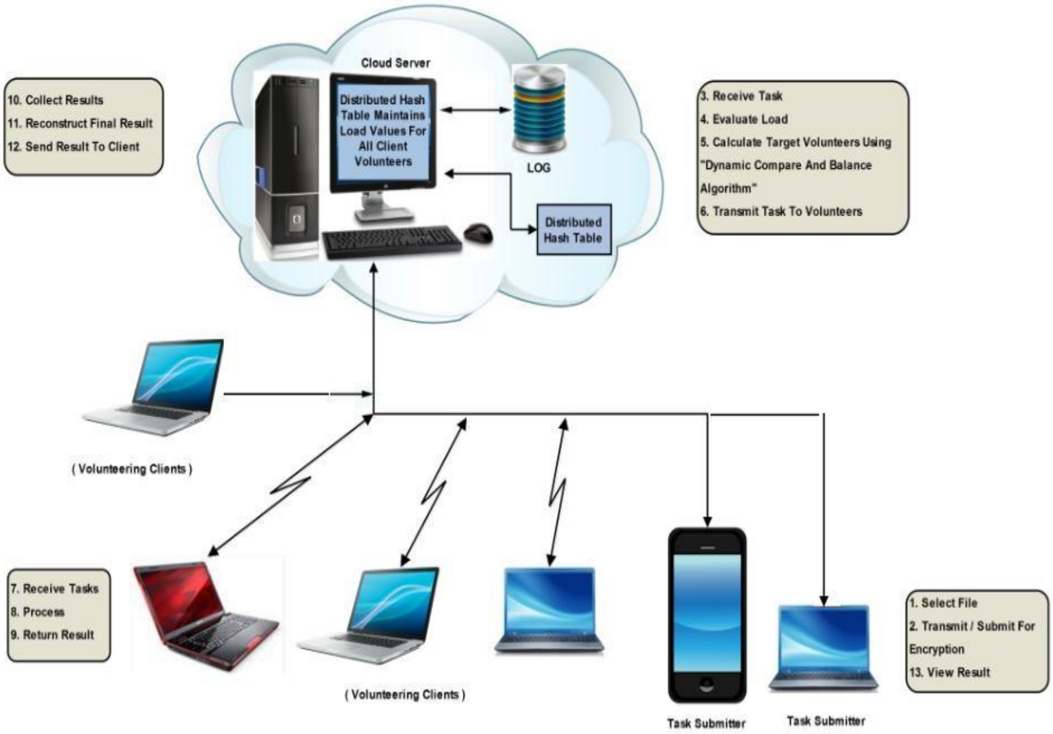

10. Collect Results
11. Reconstruct Final Result
12. Send Result To Client
Cloud Server
Distributed Hash Table Maintains Load Values For All Client Volunteers
LOG
Distributed Hash Table
3. Receive Task
4. Evaluate Load
5. Calculate Target Volunteers Using "Dynamic Compare And Balance Algorithm"
6. Transmit Task To Volunteers
(Volunteering Clients)
7. Receive Tasks
8. Process
9. Return Result
(Volunteering Clients)
Task Submitter
Task Submitter
1. Select File
2. Transmit / Submit For Encryption
13. View Result

FIGURE 1. Block diagram of Proposed System

The Submitter Client Application (Submits job to server)

- The customer application will be capable to present the job to server.
- The submitted job will be time taking which will set aside more effort to execute.
- The undertaking considered in the proposed framework will be Image handling assignment or application

Server (To manage load)

- After accepting job from the server will circulate the task to volunteer client nodes
- The algorithm running responsible for distribution in server as Linear programming can be utilized at the server side.
- According to current load the job will be partitioned/ submitted to volunteer customer.

Volunteer customer (Process the job and submit result to server)

- Volunteer customer will total the procedure and after that send the result to server.
- Server will send the expected result back to Submitter application.

The Fig.2 Proposed system Architecture

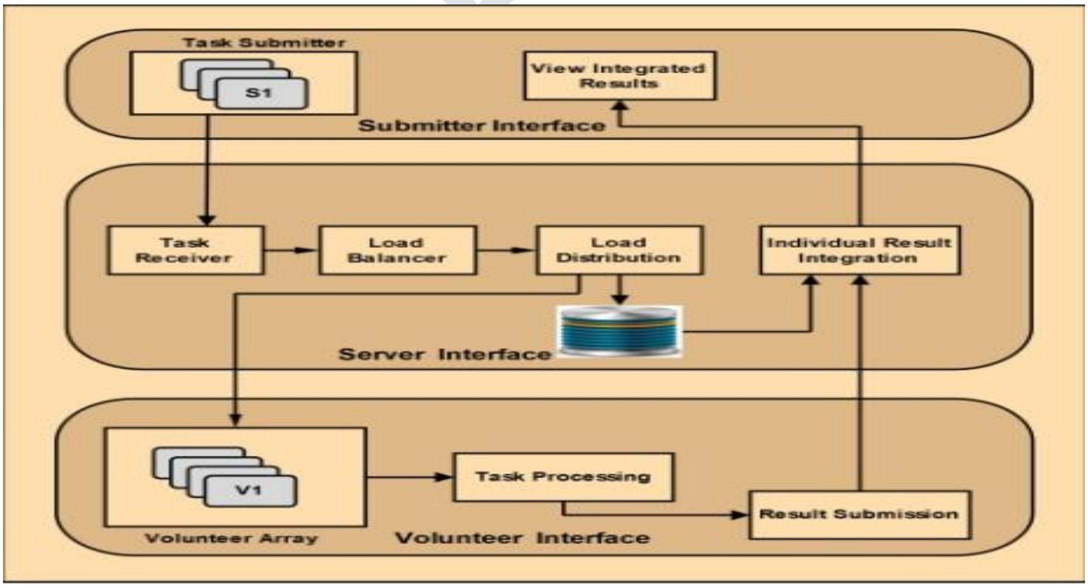


FIGURE 2. Block diagram of Proposed System

Job Submitter

- It is dependable to present the errand to server. There might be any 'n' number of submitter. The cloud is equipped for executing every one of them. So we can say that there is no constraint on number of submitters.

Job Receiver

- It will get the undertaking and forward to load balancer. In our framework the errand considered will be a picture on which essential reprocessing will be performed i.e shading picture changed over to dark scale picture. Gives take access thought that Im{I1m,I2m,I3m… Imn}, Where Im is set of picture on which pre-preparing will be performed.

Load Balancer

- After getting assignment by burden balancer the activity is isolated in parts or pieces and appropriated among the quantity of volunteer, let us state that if I1m is an info picture of size 250 X 250 where we have absolute number of bytes as 62,500 and in our framework we have for instance 3 volunteers so picture will be breaked into lumps as I1mC1, I1mC2, I1mC3. After this a heap adjusting algorithmic condition will be connected dependent on which picture will be passed to load distribution

The Load Distribution

- The load circulation will additionally advance the picture to the volunteer customers dependent on parameters determined from the load balancer.

Volunteer Array

- Assume that there will be any number of 'n' volunteer in the system. The load distribution process will be undertaking in the wake of getting, eg accept the assignment in pre-handling of picture in Gray Scale. Here every pixel from picture will be extricated. The stream is clarified in following pseudo code.

```
for(int x=0;x<ht;x++)
  {
   for (y=0;y<wdth;y++)
    {
        cal=ip image[y][x];        int b=(cal AND 0xff)        int g=(cal>>8) AND 0xff)
        int b=(cal>>16) AND 0xff)

        Gray Scale=(r+g+b)/3;
        Out Image[y][x]=Image(Gray Scale);
    }
}
```

The yield image will be conveyed back to server through Result Submission.

Singular Result Integration

- End Image submitted will be breaked in chunks currently, will be coordinated together.

6. Algorithm for Proposed System

As indicated by ongoing examination the activity on capacity nodes, i.e the document and gets to the record directly builds load on the server which results in execution bottleneck, which can not suit an extensive number of records because of essence of different customers and Map Reduce application. So reliance on focal hub to take care of or handle the issue put overwhelming burden. To determine the issue ,completely disseminated burden rebalancing calculation can adapt load unevenness issue. The framework proposes utilization of Linear programming calculation/strategy which will be utilized to pick up enhanced outcomes and obviously framework precision. Notwithstanding that the proposed framework won't allow any readymade dataset. At server side client log's and submitted errand information will be kept up dependent on that heap adjusting will be finished. The calculation extends a few factors which are examined as pursues

Consider set of Chunks c1,c2,......cn as Input

Proposed system will consist some fixed constants

Chunk Cost M □ Cost of each Chunk For Mobile Devices in network

Chunk Cost C □The Cost of each Chunk For Computer Machine in network

Chunk Time M □Time required to process each Chunk for attached mobile devices in network

Chunk Time C □ Time required to process each Chunk for attached Computers in network

The total Chunk □ Total no's of Chunks of all files given as task

According to Linear Programming Technique:

Aim : To calculate/find Chunk M and Chunk C

Chunk M □ Total file Chunk can be processed by mobile devices in network

Chunk C□ Total file Chunk to be processed by computers machine in network

Step no 1: To find Total Chunk = Chunk M + Chunk C

Step no 2 : To find Minimal Total Cost (TC) i.e Min(TC)
so, TC = (ChunkM * ChunkCostM) + (ChunkC * ChunkCostC)

Step no 3 : To find Minimal Total Time (TT) i.e Min(TT)

so, (Total Time)TT = (Chunk M * Chunk Time M) + (Chunk C * Chunk Time C)

7. Platform

Neatbeans will be used as an IDE.
Eclipse in case of android as client interface.
Hadoop Distributed File System will be used for data base and maintaining logs. Glassfish will be used for Application Server.

Client Interface:

Java or Android based on the data from Internet

S/W Requirement

Operating System	:	Windows/XP/Win 7
Application Server	:	Glass Fish
Front End	:	Java
Database /Backend	:	My sql

H/W Requirement

Processor	:	Core i3 & Above
Speed	:	2.40 GHz
RAM	:	4 GB
Hard Disk	:	500 GB
Key Board	:	Standard Windows
Mouse	:	Optical Mouse

8. Graphical Result

The graph indicates accuracy of the project.

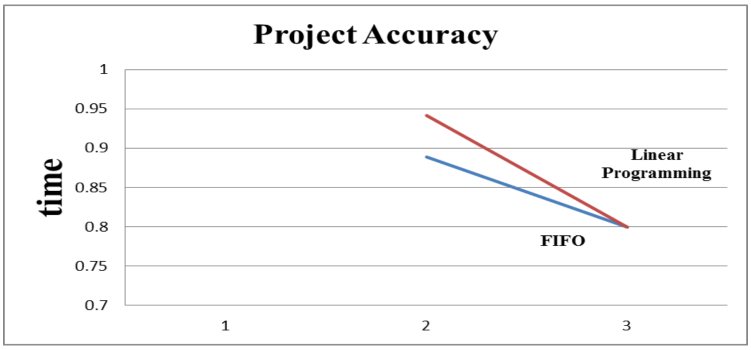
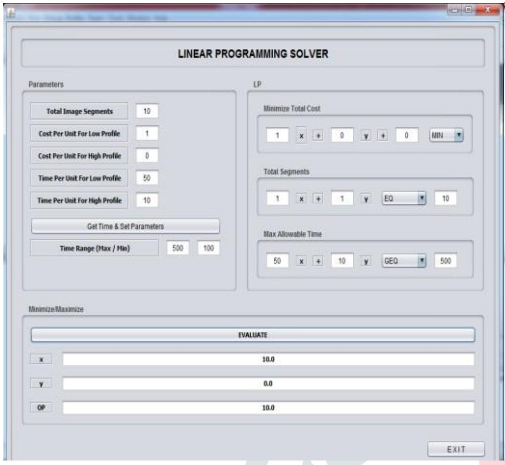


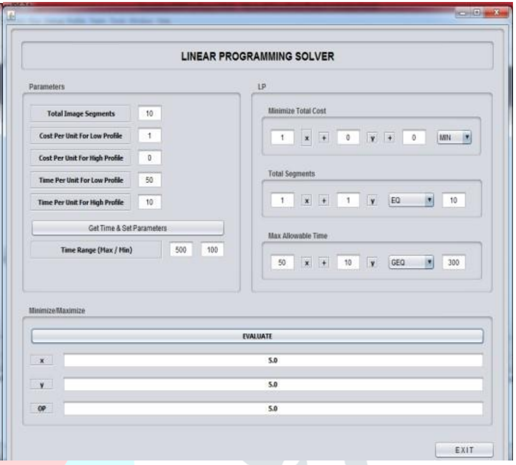
FIGURE 3 . No of tasks with respect to time

The final results of the proposed system can be expected as given below.



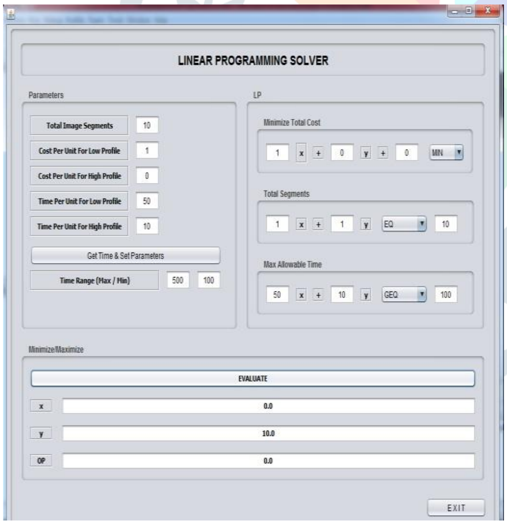
Total Image Segments: 10
Cost Per Unit For Low Profile: 1
Cost Per Unit For High Profile: 0
Time Per Unit For Low Profile: 50
Time Per Unit For High Profile: 10
Time Range (Max / Min): 500 100
Minimize Total Cost: 1 x + 0 y + 0 MIN
Total Segments: 1 x + 1 y EQ 10
Max Allowable Time: 50 x + 10 y GEQ 500
x: 10.0
y: 0.0
OP: 10.0

FIGURE 4. Chunk Distribution Max



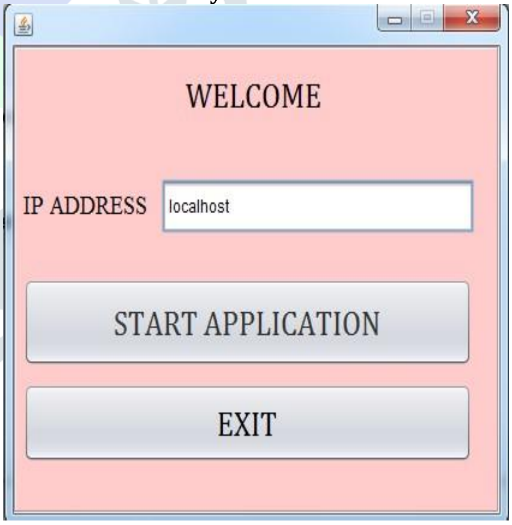
Total Image Segments: 10
Cost Per Unit For Low Profile: 1
Cost Per Unit For High Profile: 0
Time Per Unit For Low Profile: 50
Time Per Unit For High Profile: 10
Time Range (Max / Min): 500 100
Minimize Total Cost: 1 x + 0 y + 0 MIN
Total Segments: 1 x + 1 y EQ 10
Max Allowable Time: 50 x + 10 y GEQ 300
x: 5.0
y: 5.0
OP: 5.0

FIGURE 5. Chunk Distribution with Boundary conditions



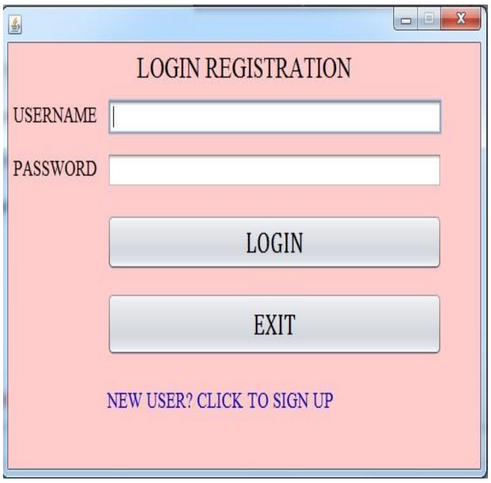
Total Image Segments: 10
Cost Per Unit For Low Profile: 1
Cost Per Unit For High Profile: 0
Time Per Unit For Low Profile: 50
Time Per Unit For High Profile: 10
Time Range (Max / Min): 500 100
Minimize Total Cost: 1 x + 0 y + 0 MIN
Total Segments: 1 x + 1 y EQ 10
Max Allowable Time: 50 x + 10 y GEQ 100
x: 0.0
y: 10.0
OP: 0.0

FIGURE 6. Chunk Distribution with Min Boundary Conditions



IP ADDRESS: localhost

FIGURE 7 Login Registration Page

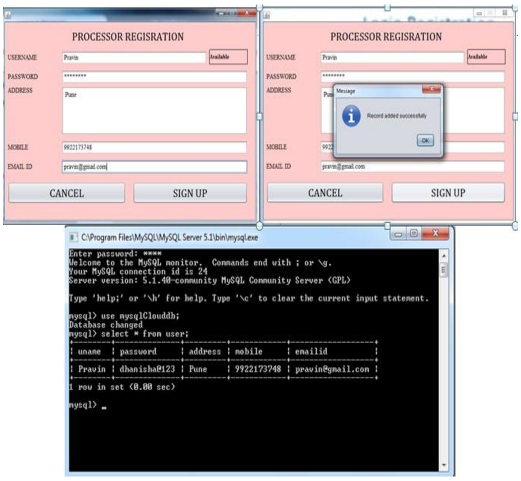


LOGIN REGISTRATION

USERNAME

PASSWORD

FIGURE 8. Login Registration New User Sign Up Page



PROCESSOR REGISRATION

USERNAME Pravin Available

PASSWORD ********

ADDRESS Pune

MOBILE 9922173748

EMAIL ID pravin@gmail.com

Message

Record added successfully

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 24
Server version: 5.1.40-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use mysqlClouddb;
Database changed
mysql> select * from user;
+--------+--------------+---------+------------+------------------+
| uname  | password     | address | mobile     | emailid          |
+--------+--------------+---------+------------+------------------+
| Pravin | dhanisha@123 | Pune    | 9922173748 | pravin@gmail.com |
+--------+--------------+---------+------------+------------------+
1 row in set (0.00 sec)

mysql> _
```

FIGURE 9. Database Record

Conclusion

The proposed framework expects to take care of burden re-adjusting issue for huge scale information serious mists by allotting the lumps of records in disseminated document frameworks consistently among capacity hubs with the end goal that none of the hub deal with any number of excess chunks. The framework especially will incorporate straight calculation/strategy which will have quick union rate with fine enhanced outcomes and framework exactness. In future, framework will be in-corporating HDFS to keep up logs and accumulation of data in big cluster environment.

REFERENCES

[1] Survey of Different Load Balancing Approach-Based Algorithms in Cloud Computing: A Comprehensive Review[1]Springer Nature Singapore, Pte Ltd. **(2019)**, doi.org/10.1007/978-981-10-8055-5_10

[2] Load Balancing Tools and Techniques in Cloud Computing: A Systematic Review[2] Springer Nature Singapore Pte Ltd. **(2018)**, doi.org/10.1007/978-981-10-3773-3_18

[3] An Advanced Load Balancing Strategy For Cloud Environment[3] 2016 IEEE, **(2016)** 17th International Conference on Parallel and Distributed Computing, Applications and Technologies

[4] Load Balancing in Cloud Computing:Challenges &Issues[4] 2016 IEEE, **(2016)** 2nd International Conference on Contemporary Computing and Informatics (ic3i)

[5] A Survey of Load Balancing Challenges in Cloud Environment[5] Proceedings of the SMART -2016, IEEE Conference ID: 39669 5th International Conference on System Modeling & Advancement in Research Trends, 25th_27'h November, **(2016)**

[6] Load Balancing in Cloud Data Center Using Modified Active Monitoring Load Balancer[6] , 2016 IEEE, 978-1-5090-0673-1/16/$31.00 ©**(2016)** IEEE

[7] An in-depth analysis and study of Load balancing techniques in the cloud computing environment[7],ScienceDirect 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

[8] Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm[8] , 978-1-4673-7910-6/15/$31.00 © **(2015)** IEEE