A Distribution of Nodes in Big Data using Hadoop Open Source System

Nishant Mathur, Mukul Jain

Abstract*: *Apache Hadoop is an free open source Java framework under Apache Software Foundation. It provides storage of large amount of data efficiently with low costing. Hadoop has two main core components one is HDFS (Hadoop Distributed File System) and second Map Reduce. It is basically a file system and has capability of high fault-tolerant and while deploying supports less cost hardware. It provides the high speed admittance to the relevance data. The Hadoop architecture is based on cluster, which consist of two nodes named as Data -Node and Name-Node which perform the internal activity known as heart beat to process data storage on distributed file system and Map reducing is performed internally to show the clustering of distributed data on localhost of ssh serverwebsite. Large quantity of data is needed to store in distributed file structure, for this Hadoop has played important role. Maintaining the large volume storage, making data duplicity for providing security and recovery of big data for its analysis and prediction.

Keywords: Big Data, Hadoop, HDFS, Map Reduce Data-Node, Name-Node, Task-Tracker and Job-Tracker.

I. INTRODUCTION

The term Big Data is used to depict the collection of composite and large data sets such that it is complicated to store, practice and examine. In today's era when we can't wait even for a five seconds to open a Google page or social network page then how can we wait to analyze this much huge amount of data that are generated everyday by different means such as sensors, Facebook, Twitter, flight data, etc. Let us explain what we really want to explain, suppose a team of marketing people have some huge data with them and they want to perform analysis on that data to improve their marketing scheme, back days it use to happen if some famous vendor in the market approach them to analyze that data at that time, it might have taken such as 10 years to analysis it.

What we can conclude that after long 10 years is there any relevance and meaning of that data for the user, obviously not. So we cannot think of such a long duration should be used and followed for analyzing big size of data in present era. So the improvisation we got for such kind of problems like analyzing and storing large size of data, here comes the technology and concept known as Big Data.

Revised Manuscript Received on January 30, 2020.
***** Correspondence Author

Mr. Nishant Mathur *, Assistant Professor, ICFAI Tech School, ICFAI University, Dehradun , India.
Dr. Mukul Jain, Assistant Professor, ICFAI Tech School, ICFAI University, Dehradun , India.

When we talk about Big Data, the first name comes in mind is "HADOOP" a well known product that introduces as one of the solution for Big Data problems. According to IBM definition of Big Data, it has three components commonly known as three V's they are 1) Volume (data in Exabyte or Petabyte), 2) Variety (Unstructured, Semi Structured and Structured data) and 3) Velocity (data processing speed). By considering every data in these three components we can easily store and retrieve our Big Data.

II. FOUNDATION OF HADOOP

The start of Apache Hadoop Projects was developed from the White paper series which was proposed by Google on Big Table, which uses chunks concepts in GFS from there the idea of Map-Reduce emerged. Google has published whitepaper ice breaking idea for big and distributed file and gave the idea of how to handle Big Data using distributed file system but the code and implementation was never disclosed. Later Yahoo and many other contributors implemented this Google's white paper idea and that's how Hadoop file system framework was proposed. Hadoop is top level Apache Software project led by Yahoo [1][2]. Another term which we have is Hadoop Cluster there might be quite confusion between Hadoop and Hadoop cluster in people's mind. Let us discuss more on their differences.

A. Hadoop

Hadoop is an open source structure, which proposes the functioning of big data sets in a distributed computing surroundings. It has of Map-Reduce, Hadoop Distributed File System (HDFS) and large quantity of interrelated projects like Apache Hive, Hbase, PIG and Zookeeper [11].

B. Hadoop Cluster

Hadoop is unique types of formulated structure for storing, analyzing and managing large amount of structured, semi structured and unstructured files in a distributing atmosphere for computing. These cluster runs on hardware of related and commodity types [11].

III. FOUNDATION WORKINGS OF HADOOP CLUSTER

Hadoop clusture consist three main working components they are Master, Client and Slave. Here Client is neither master nor slave; rather it plays one of the very important roles of loading the data into cluster and submits the jobs to Map Reduce. Also it describes how stored data should be access and retrieves to analyze the reaction after achieving jobs.

Master services consist of service like Name-Node and Job-Tracker while Slave services are Data-Node and Task-Tracker which we will be discussing in detail later in this paper. Master and Slave component of hadoop cluster will be discussed further showing with the specific functioning wherever required.

IV. HADOOP DISTRIBUTED FILE SYSTEM

A. **Introduction-** The Hadoop Distributed File System (HDFS) is a associate venture idea of the Apache Hadoop, the origin of this idea was from Google White paper on Google File System (GFS) [1][2]. It is distributed and extremely error forbearing file system designed to run on less charge commodity hardware. It provides great throughput contact to function of data and also fit for applications with huge data sets [1].

B. **Block Structure-** HDFS is a specifically deliberate file system for managing big data and its sets with the bunch of commodities hardware and analyzing access outline, here streaming access sample works as said in java's slogan i.e. "Write once run anywhere", same way here streaming access pattern in Hadoop means you Write once and read it number of times and never try to change the content of the file. Conceptually, HDFS is a block-controlled file system where connected files are of a fixed size may be of 64Mb or 128Mb broken into blocks.

These blocks with data storage capability are stored crosswise in cluster of one or more equipment [1], as shown in Fig. 1.

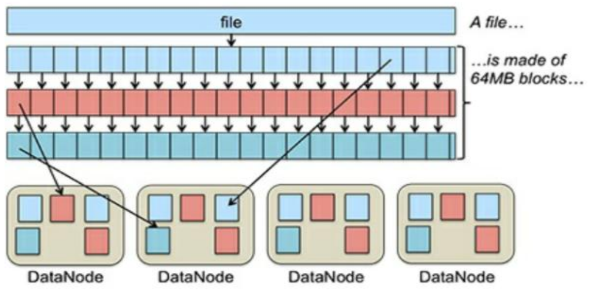


Fig. 1 Example of Block Structure files system

C. **Structural Design-** HDFS is collection of interrelated cluster of point as node where files and directories inhabit. It has two types of services, one is master service and another is slave service. Master services consist of Name Node, Secondary Name-Node and Job-Tracker whereas Data-Node and Task-Tracker consists of Slave services. Each master service can communicate to one other like name node and job tracker can communicate to each other in master service similarly on the other hand each slave service can communicate to each other within here data node and task tracker can communicate to each other, whereas when they have to load and process the data then master can only communicate to its specific slave service, like master's name node can only communicate to slave's data node and master's job tracker can communicate to slave's task tracker ,as shown in Fig. 2.

D. **Name Node-**It is one of the master services. It controlled the data node presents on building blocks in file system. It has meta-data of those data which are stored on data node. If there is any loss of metadata happens in name node and we are unable to access that data from data node this type of failure is known as single point failure. Therefore it requires high reliable hardware for name node.

E. **Data Node-** It is slave service for the name node which is deployed on commodity hardware. It is the location where information is collected and is dependable for managing functions like examine and writes. These services are requested by the clients.

1. Secondary Name Node- As we have already discussed name node in master services, secondary name node is the sub method in master services. Its main responsibility is to perform intermittent checkpoints. At the performance of name node when their failures occur, it can restart the name node utilizing the checkpoint [11].

F. **Job Tracker-** Job-tracker plays one of the important role of master, it is the one to which client request submit performance to carry out the Map Reduce function. Job-tracker submit job to task tracker of slave service and manages overall execution of Map Reduce job.

G. **Task Tracker-** Task tracker plays the role of slave, every task-tracker is accountable to perform and supervise everyone tasks assigned by job-tracker. It manages the data movement.

Task-tracker send the heart strike to job-tracker that tells the job-tracker that yes it is alive and process the job that was submitted to it, If it fails to send heart beat then job tracker assumes that task tracker is crashed then again it submit the job to the corresponding task tracker of other nodes in the cluster who so ever have data that client requested.

V. NAME NODE AND DATA NODE: THE BONDING STORY

When client has some data they wants to store in HDFS for effective storage and fast processing client request the name node that it already have some data but they don't know to which data node they will store data, in response of this request the name node send the information to that data node which have enough free space.

The data stores in the data node need not to be in serial manner it depends on the free available space in it. As we discussed before that name node can communicate to data node for storing the data.

The data-node sends block information and heart hit to let the name-node aware that it is alive and working. In general what happen actually is that data is stored in data node while the name node maintains metadata i.e. data of the data.

This metadata contains information about on which data node client has stored its data and on which data node the replication of data has been made.

If the name node fails or crashes sometime then we say it's a loss of metadata, without the metadata we cannot process to store data from data node.

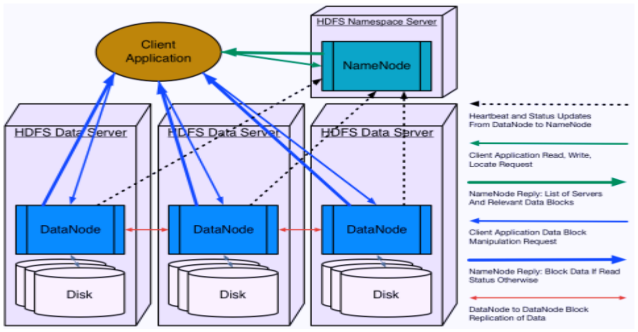

HDFS Namespace Server
NameNode
Client Application
HDFS Data Server
HDFS Data Server
HDFS Data Server
DataNode
DataNode
DataNode
Disk
Disk
Disk
Heartbeat and Status Updates From DataNode to NameNode
Client Application Read, Write, Locate Request
NameNode Reply: List of Servers And Relevant Data Blocks
Client Application Data Block Manipulation Request
NameNode Reply: Block Data If Read Status Otherwise
DataNode to DataNode Block Replication of Data

Fig. 2 Example of HDFS Architecture

As we were discussing about the HDFS which is basically a file system, here question arise is, well if it is a file system, so what kind of file organization it has? We can explain this by saying that HDFS managed usual hierarchical file management in which directories was created by users, files renaming was done by user inside the directory and using HDFS commands reposition or deletion can be done in files. It can also manage Cloudera and Amazon Simple Storage System (Amazon's S3) in Amazon web services as a third-party file system [1][5].

A. **Data Replication in HDFS-** When a client loads its data from local file system to HDFS. Its data node makes two more copies of the same data that means maximum we have three copies of every data on different data nodes and if any kind of failure occurs in any of the data node then the user's data will not be lost. It makes replication of data by default.

B. **Block Size-** HDFS is special file system that uses, to store the data in memory, the block size of 64Mb or 128Mb. In local file system the memory block size used is of 4Kb.

C. **Communication-** There is two types of signals sent between data node and name node

1. **Heart Beat-** The data-node and task-tracker used to launch heart beat to name-node and job-tracker to let their master know that they are alive and in working mode. If in case any of data node is crashed or dead no heart beat is sent to data-node and job-tracker. Data stored in dead data-node is not much longer existing to HDFS client from that port, which is proficiently erased from the system. If the casualty of a node happened then the imitation factor drops below to their lowest significance, the name-node starts duplication of data to bring the imitation factor (i.e. three copies of same data) back to the normalized state [1].

2. **Block Report-** Whenever client stores the block into data node it sends the proper block report to name node. By the help of this block report map reduce is performed on HDFS stored data.

VI. THE IMPLEMENTATION

The implementation of Single Node Hadoop Cluster on Ubuntu operating system is explained in following steps:

Step 1: Download and install JAVA JDK1.8.0_31 from website [14].

Then moved that file to directory created for JAVA

step 2: JDK installation.

Code: sudo mvjdk1.8.0_31/*/usr/lib/nishant/jvm/jdk1.8.0_31

Now run the following command to enable JAVA JDK,

Code: sudo update-alternatives –-install "/usr/nishant/bin/java" "java" "/usr/lib/jvm/jdk1.8.0_31 bin/java" 1

sudo update-alternatives –install "/usr/bin/nishant/javac" "javac" "/usr/lib/jvm/jdk1.8.0_31/bin/javac" 1

sudo update-alternatives –install "/usr/bin/nishant/javaws" "javaws"

"/usr/lib/jvm/jjdk1.8.0_31 /bin/nishant/javaws" 1

After running these commands set java path under .bashrc file by creating a JAVA_HOME variable.

Code: #JAVA_HOME directory setup

export JAVA_HOME="/usr/lib/nishant/jvm/jdk1.8.0_31"

set PATH="$PATH:$JAVA_HOME/bin/nishant"

export PATH

Step 3: Now Install Apache Hadoop free open source.

For this download Apache Hadoop free open source latest Version from its official website [3]

Then edit hadoop-env.sh inside the /home//nishant/user/Downloads/hadoop-1.2.1/conf_by adding following code.

Code:

export JAVA_HOME="/usr/lib/nishant/jvm/jdk1.8.0_31"

After this now pass the path address of Hadoop inside .bashrc file link again by adding the below code.

Code:

export HADOOP_HOME="Downloads/hadoop"

PATH=$PATH:$HADOOP_HOME/bin/nishant

Type the export PATH

Then reconfigured core1-site.xml, hdfs1-site.xml and mapred1-site.xml by adding the following code to respective .xml file.

Code: core1-site.xml

<name>fs.default.name</name>

<value>hdfs://localhost:8000</value>

Code: hdfs1-site.xml

<name>dfs.replication</name>

<value>1</value>

Code: mapred1-site.xml

<name>mapred.job.tracker</name>

<value>localhost: 9001</value>

Step 4: Installation of ssh server

Code: sudo apt-get install ssh

Step 5: To Connecting ssh server

Code: ssh localhost

Step 6: Enter password

Step 7: For the first time formate the name node before starting it

Code: hadoop namenode -format

Step 8: To Start all services

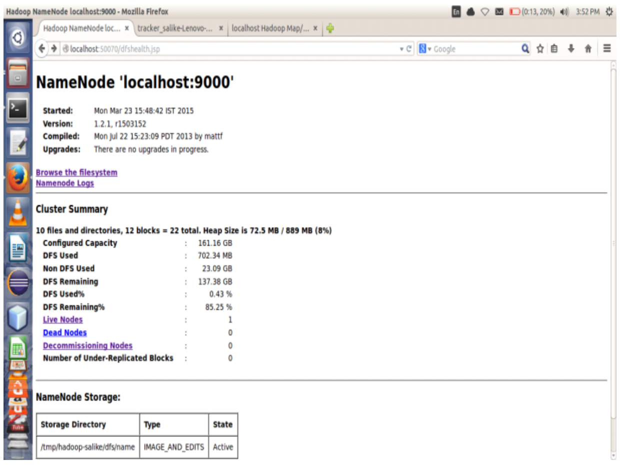
Code: start-all.sh

Step 9: To check connection services has been established or not, enter the following address to browser [3]. You get the following Screen as output, as shown in following screenshot when you enter the following address to browser.

http://localhost:50070/, as shown in Fig. 3.

http://localhost:50030/, as shown in Fig. 4.

http://localhost:50060/, as shown in Fig. 5.



Hadoop NameNode localhost:9000 - Mozilla Firefox

localhost:50070/dfshealth.jsp

NameNode 'localhost:9000'

Started:	Mon Mar 23 15:48:42 IST 2015
Version:	1.2.1, r1503152
Compiled:	Mon Jul 22 15:23:09 PDT 2013 by mattf
Upgrades:	There are no upgrades in progress.

Browse the filesystem
Namenode Logs

Cluster Summary

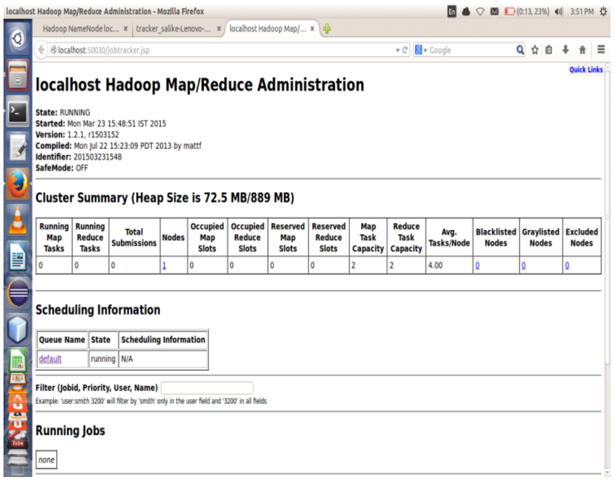
10 files and directories, 12 blocks = 22 total. Heap Size is 72.5 MB / 889 MB (8%)

Configured Capacity	:	161.16 GB
DFS Used	:	702.34 MB
Non DFS Used	:	23.09 GB
DFS Remaining	:	137.38 GB
DFS Used%	:	0.43 %
DFS Remaining%	:	85.25 %
Live Nodes	:	1
Dead Nodes	:	0
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	0

NameNode Storage:

Storage Directory	Type	State
/tmp/hadoop-salike/dfs/name	IMAGE_AND_EDITS	Active

Fig. 3 Example of localhost: 50070



localhost Hadoop Map/Reduce Administration - Mozilla Firefox

localhost:50030/jobtracker.jsp

Quick Links

localhost Hadoop Map/Reduce Administration

State: RUNNING
Started: Mon Mar 23 15:48:51 IST 2015
Version: 1.2.1, r1503152
Compiled: Mon Jul 22 15:23:09 PDT 2013 by mattf
Identifier: 201503231548
SafeMode: OFF

Cluster Summary (Heap Size is 72.5 MB/889 MB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes	Graylisted Nodes	Excluded Nodes
0	0	0	1	0	0	0	0	2	2	4.00	0	0	0

Scheduling Information

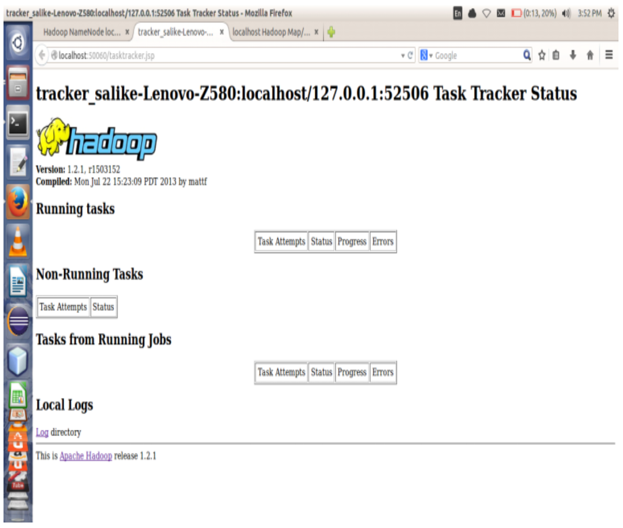
Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

none

Fig. 4 Example of localhost:: 50030



tracker_salike-Lenovo-Z580:localhost/127.0.0.1:52506 Task Tracker Status - Mozilla Firefox

localhost:50060/tasktracker.jsp

tracker_salike-Lenovo-Z580:localhost/127.0.0.1:52506 Task Tracker Status

Version: 1.2.1, r1503152
Compiled: Mon Jul 22 15:23:09 PDT 2013 by mattf

Running tasks

Task Attempts	Status	Progress	Errors

Non-Running Tasks

Task Attempts	Status

Tasks from Running Jobs

Task Attempts	Status	Progress	Errors

Local Logs

Log directory

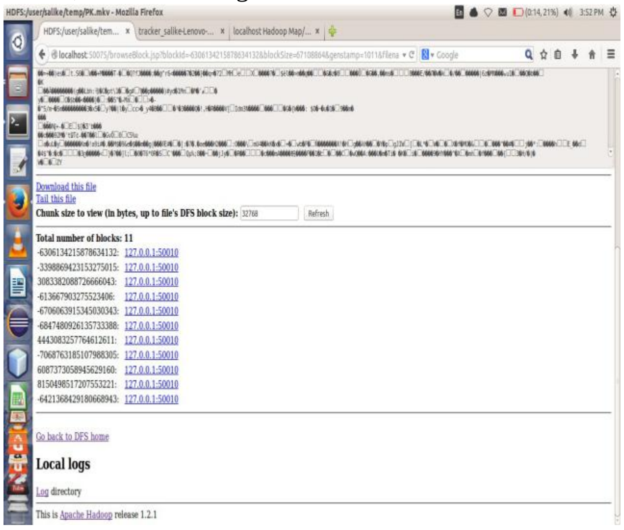
This is Apache Hadoop release 1.2.1

Fig. 5 Example of localhost: 50060

The above screenshot shows the successfully configured Hadoop single node using local host on Ubuntu.

VI. RESULTS

After performing single node implementation on Apache Hadoop, the file which we uploaded as sample data is movie file of size 730.7 Mb with .mkv format that was much heavy file we used as compared to the normal text file, this has been distributed on eleven number of blocks of size 64 Mb each and there time calculation has been optimized and access of data after distribution on different has become fast and easy. If we could have used normal file system of 4 Kb block size then this same file of 730.7 Mb should have used 299307.2 numbers of blocks in memory. This is one of the main reasons why we are using block size of 64 Mb or 128 Mb in HDFS, so we can say by our experiment that lesser the number of memory block used by data, lesser the number of meta data in name node. Output of file distribution in 11 blocks of 64 Mb file sized, is shown in Fig. 6.



HDFS:/user/salike/temp/PK.mkv - Mozilla Firefox

localhost:50075/browseBlock.jsp?blockId=-6306134215878634132&blockSize=67108864&genstamp=1011&filena

[illegible]

Download this file
Tail this file
Chunk size to view (in bytes, up to file's DFS block size): 32768 Refresh

Total number of blocks: 11

```
-6306134215878634132: 127.0.0.1:50010
-3398869423153275015: 127.0.0.1:50010
3083382088726666043:  127.0.0.1:50010
-613667903275523406:  127.0.0.1:50010
-6706063915345030343: 127.0.0.1:50010
-6847480926135733388: 127.0.0.1:50010
4443083257764612611:  127.0.0.1:50010
-7068763185107988305: 127.0.0.1:50010
6087373058945629160:  127.0.0.1:50010
8150498517207553221:  127.0.0.1:50010
-6421368429180668943: 127.0.0.1:50010
```

Go back to DFS home

Local logs

Log directory

This is Apache Hadoop release 1.2.1

FIG. 6 Example Showing Distribution Of 730.7 Mb Files In 11 Hdfs Block

VII. CONCLUSION

Apache Hadoop is an open source disseminated file structure and data managing project that is used for keeping and managing huge data sets. It has two core components first HDFS for storage of data and second Map Reduce for processing of data. Hadoop is not a database. The Hadoop platform was designed to solve Big Data three V's problem. Also it is used 1) For search as in Yahoo, Amazon, Zevents. 2) For log processing as in Facebook, Yahoo etc 3) For Data Warehouse as for Facebook, AOL. 4) For Video-Image examination like New York Times, Eyealike[2]. The Hadoop structure mostly written in JAVA programming language, with some indigenous code in C and control written in shell-script is useful for data management and its analysis for managing big size data and hence can be utilized for data extraction and foretelling.

VIII. FUTURE WORK

Our present work shows block size of 64 Mb has been used, for further we can manage to use the block size to 128 Mb by which we can further reduce the space for Metadata in memory allocation as we can have big data to implement the modified version of algorithm. Also further the working on Hadoop Multi Node cluster's implementation can be achieved which will explore the power of distributed computing on multiple systems by using Apache Hadoop and Apache nutch indexing concept on open source system.

REFERENCES

1. J. Jeffrey Hanson, CTO MAX"An Introduction to the Hadoop Distributed File System" by IBM developer works 01 February 2011.
2. (2015) The Hadoop Apache homepage on wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/Apache_Hadoop
3. (2015) The Hadoop Distributed File System project website. [Online]. Available: https://hadoop.apache.org/
4. (2012) The Big Data University website. [Online]. Available: https://bigdatauniversity.com/wpcourses/
5. (2011) S3 infrastructure homepage on Amazon [Online]. Available: http://aws.amazon.com/s3
6. Agrawal, R., Rajagopalan, S., Srikant, R. and Xu, Y, "Mining Newsgroups using Networks Arising from Social Behavior" in *12th International World Wide Web Conference (WWW 2003)*,(pp. 529-535), Budapest, Hungary, May 2003.
7. Aktas, M. S., Nacar, M. A. and Menczer.F, "Personalizing Page Rank Based on Domain Profiles" in Workshop on Web Mining and Web Usage Analysis, Seattle, WA, Aug 2004.
8. (2004) Search Kit Reference homepage on Apple Developer Connection.[Online]. Available:https://developer.apple.com/library/mac/documentation/UserExperience/Reference/SearchKit/index.html
9. Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A. andRaghavan, S. Searching the Web in ACM TOIT, 2001
10. Joachims, T. Optimizing, "Search Engines Using Click through Data" in ACM Knowledge Discovery and Data Mining, 2002.
11. (2014) Sap hana and Hadoop homepage on saphanatutorial [Online].Available:http://saphanatutorial.com/sap-hana-online-courses
12. (2014) Hadoop cluster on homepage search business analytics's TechTarget[Online].Available:http://searchbusinessanalytics.techtarget.com/definition/Hadoop-cluster
13. (2016) Tsai, Cheng-Fa, and Yao Chiang, "Enhancement of data clustering using TSS-DBSCAN approach for data mining," International Conference on Machine Learning and Cybernetics (ICMLC), IEEE, Vol. 2, pp. 535-540.
14. (2017) Charu C. Aggarwal, "Introduction to special issue on the Best paper from KDD 2016," IBM T.J. Watson Research Center, Vol.11, Issue 4, pp. 39.

AUTHORS PROFILE

Mr. Nishant Mathur received B.Tech, M.E. (Computer Science and Engineering) degree from Delhi College of Engineering and Ph.D.(Pursuing). His specialized fields are Database Systems, Data Mining, Machine Learning and Big Data Analysis. He published several research papers in reputed conferences & Journals. He is Assistant Professor of Computer Science and Engineering at ICFAI Tech School, The ICFAI University, Dehradun.

Dr. Mukul Jain (Assistant Professor) is Doctorate from H.N.B Gharwal University in the field of Operational Research. He earned his Bachelor degree in Mathematics with First Division in 1997 and completed his Masters Degree in Mathematics in 1999 from Gharwal University, Uttarakhand, Dehradun. He has published more than 6 papers in peer reviewed international/national journals and conferences. He is Department In charge at ICFAI Tech School, The ICFAI University, Dehradun.

.