ISSN: 0975-766X
CODEN: IJPTFI
Available Online through Research Article
www.ijptonline.com

WEATHER DATA ANALYSIS USING HADOOP

M.Senthilkumar [1], N.Manikandan [2], U.Senthilkumaran [3], Renga Samy [4]
[1]Assistant Professor (Senior) School of Information technology, Vit University, Vellore.
[2]Assistant Professor (Selection Grade) School of Information technology, Vit University, Vellore.
[3]Associate Professor, School of Information technology, Vit University, Vellore.
[4]Associate Professor, Department of computer science, Eckenforde Tanga University.
Email: mosenkum@gmail.com

Received on: 18.10.2016 Accepted on: 11.11.2016

Abstract

Hadoop an apache product which is an open-source, Java based programming framework is used to support large data sets in a distributed environment. Hadoop has maximum advantage over scalable and fault-tolerant distributed processing technologies. Also,(HDFS)Hadoop Distributed File System is highly fault tolerant and used for applications that have large data sets. Hence HDFS file system using name node, data node and task tracker will perform distribution of job in Hadoop environment .Since Hadoop has overwhelming advantage in optimizing big data , we prefer to use Hadoop to analyze large datasets of weather processing.

Keywords: Distributed file system, Hadoop, distributed systems, MapReduce, weather prediction.

Introduction

Today we have several obstacles in handling more servers which results in failures. Many companies are forced to discard their most essential data because the cost of storing is too high. We all wish that all of our software transactions occurs seamlessly without defects and the data gets stored in structural and well organized way . Hadoop with HDFS and MapReduce overcomes these problems and it is designed to continue to work in the face of system failures and compute large datasets in parallel[9][10].

Hadoop Mapreduce

Hadoop includes MapReduce, a distributed data processing model that runs on large clusters of machines. A Hadoop Map Reduce job mainly has two user-defined functions: map function and reduce function. The input to Hadoop Map Reduce job should be of key-value pairs(k, v) and map function is called for each of these pairs[1] [2].

1.JobTracker and TaskTracker**:**

There are two types of nodes that control the execution process: a jobtracker and tasktrackers. The jobtracker coordinates and schedules all task to run on tasktrackers. Tasktrackers in turn sends progress reports to the jobtracker. If a task fails, the jobtracker can reschedule it on a different tasktracker.The essence of Hadoop map Reduce is that the users just define map and reduce functions. Hadoop's framework take care of everything. Hadoop map Reduce uses the Hadoop Distributed File System to perform I/O performance[3][4].

Architecture

A. Hadoop MapReduce Architecture

MapReduce has two phases: Map phase and Reduce Phase.

1. Prepare Map() input – Map function takes the weather data files as input files and transforms the input into key/value pairs, processes and generates zero.

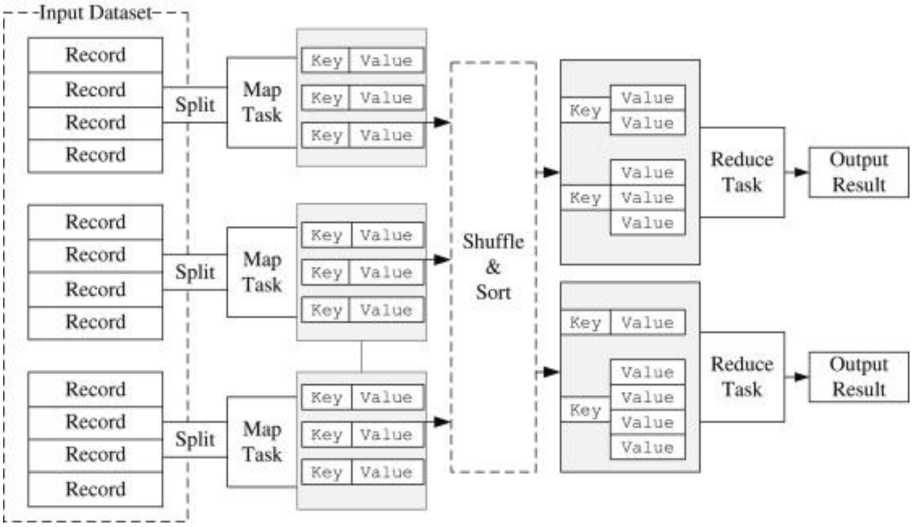


Fig. 1 MapReduce Architecture.

2. Partition function – Each map function is allocated to reducer by application's partition function. The partition function is given key and number of reducers and returns index.

3. Reduce function – This function is used to call for unique key in sorted order. Reduce function takes the intermediate file produced by the map function and adds up the values present in the file to find their sum. After adding up all the values, it finds the mean value of the value and sends it as a output to the user.

4. Output Writer- The Output Writer writes the output of the Reduce to the stable storage, usually a distributed file system.

B. HDFS Architecture

A HDFS cluster has two nodes operating in a master-worker pattern: a name node(the master) and a number of data nodes(workers). A single name node manages file system and all the metadata in directories. The information is stored in two files : namespace image and edit log. NameNode executes file system namespace operations like opening, closing and renaming files and directories. Data node perform block creation, deletion and replication from NameNode[5][6].

NameNode and DataNode are pieces of software. HDFS is built using java language. The architecture does not preclude running multiple DataNodes on the same machine but in a real deployment that is rarely the case.

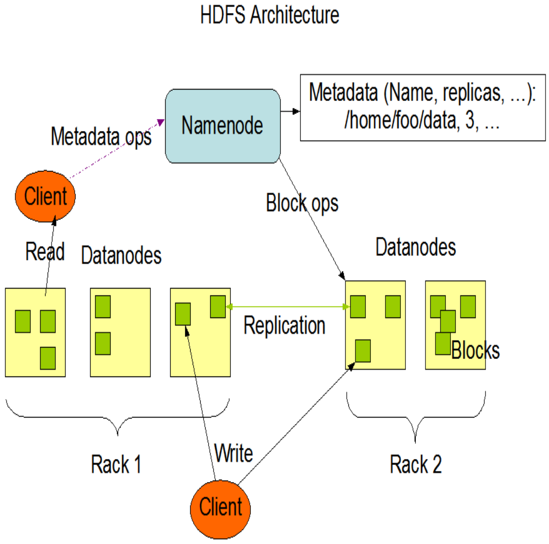


Fig. 2 HDFS Architecture.

The single NameNode simplifies architecture of the system and it is the repository for all HDFS metadata. The system is designed in such a way that user data never flows through the NameNode. One of the important aspect is data nodes directly retrieve data. This design allows HDFS to scale large clients in parallel.

Hadoop

Hadoop has powerful features in which few are discussed here[7].

1.Hadoop has the power to add new nodes without changing the clusters(data). A Hadoop cluster can accommodate more than one node without any difficulty.

2.Since Hadoop is used to store large bytes of data its more affordable when compared to any other servers.

3.Hadoop doesn’t follow norms for structuring data, hence it is flexible to work with. Another importance is multiple data can be combined in Hadoop, whether it is structured or not. Hadoop can perform map and reduce jobs very effectively.

4.Hadoop can redirect the data into another location when there is a fault in the given primary node. When one of the nodes in a cluster fails, the job can be redirected to some other node and hence Hadoop is highly an fault tolerant system.

5.Since Hadoop does parallel computing, system is more effective and efficient in terms of deriving the results.

6.Hadoop offers large cluster of local servers to store large amount of data’s.

Hadoop Environment Setup

Running Hadoop on a node requires few additional softwares installed in prior. They are Internet Information Services(IIS), Visual Studio 2010 service pack 1, Java 6 and above and NetBeans or Eclipse IDE. The operating system we have used is Windows 7 Home premium. The installation procedure for Hadoop and its prerequisite softwares are[8] :

1. Initially the IIS has to be installed in the system. Before the installation of IIS the windows features for IIS ought to be enabled.

2. Following the IIS, the other softwares such as Java, Visual Studio and NetBeans has to be installed.

3. The Microsoft HDInsight Developer Preview and the Hortonworks Data Platform for Windows are installed using the Microsoft Web Platform Installer. The version of Hadoop we have used for this is Hadoop-1.1.0-SNAPSHOT.

4. Once the Hadoop is installed, the desktop icons of Hadoop Name Node Status, Hadoop MapReduce Status, Hadoop Command Line and Hadoop Dashboard are created. The Name Node Status shows the status of the name node, number of live nodes, number of dead nodes, name node logs, HDFS files, Name node storage and cluster summary. The MapReduce Status gives the MapReduce status, cluster summary, Scheduling Information, job summary and log files.The Hadoop Command line is used to execute the Hadoop commands and to run and manage Hadoop MapReduce jobs.The Hadoop DashBoard gives a GUI representation to the Hadoop Interface.

5. If the Hadoop is installed correctly, the Hadoop Name Node status shows the live node as 1.

Data Analysis

The weather data used in this project are based on data exchanged under the World Meteorological Organization (WMO) World Weather Watch Program. The weather data over 9000 stations' data are collected in hourly basis and it is given in the datasets for the years 1999 to 2010.

The daily elements included in the dataset (as available from each station) are:

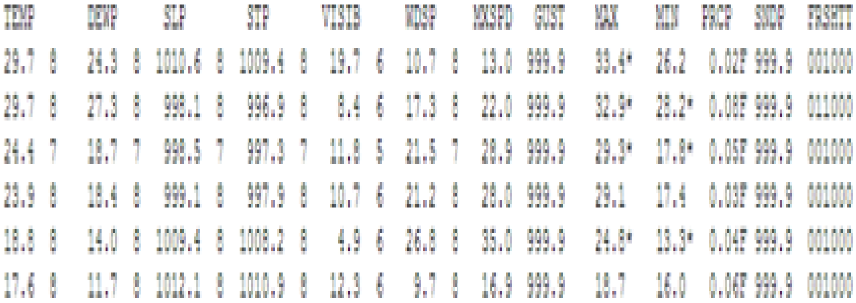
Mean temperature (.1 Fahrenheit) ,Mean dew point (.1 Fahrenheit) ,Mean sea level pressure (.1 mb)

Mean station pressure (.1 mb) ,Mean visibility (.1 miles) ,Mean wind speed (.1 knots) ,Maximum sustained wind speed (.1 knots),Maximum wind gust (.1 knots),Maximum temperature (.1 Fahrenheit)Minimum temperature (.1 Fahrenheit),Precipitation amount (.01 inches),Snow depth (.1 inches),Indicator for occurrence of: Fog, Rain or

Drizzle , Snow or Ice Pellets, Hail ,Thunder, Tornado/Funnel Cloud.Using these values, the weather forecasting for the future years can be done by calculating the mean average temperatures for each day and month for several years. The sea level pressure, temperatures and other climatic conditions repeat over each year based on the various climates. So when the average of these values are processed using Hadoop and the values are calculated, it would be possible to predict the climate conditions for the further years.

Running Mapreduce Programs

The weather dataset has the following structure of data.



TEMP		DEWP		SLP		STP		VISIB		WDSP		MXSPD	GUST	MAX	MIN	PRCP	SNDP	FRSHTT
29.7	8	24.3	8	1010.6	8	1009.4	8	19.7	6	10.7	8	13.0	999.9	33.4*	26.2	0.02F	999.9	001000
29.7	8	27.3	8	998.1	8	996.9	8	8.4	6	17.3	8	22.0	999.9	32.9*	26.2*	0.08F	999.9	011000
24.4	7	18.7	7	998.5	7	997.3	7	11.8	5	21.5	7	28.9	999.9	29.3*	17.8*	0.05F	999.9	001000
23.9	8	18.4	8	999.1	8	997.9	8	10.7	6	21.2	8	28.0	999.9	29.1	17.4	0.03F	999.9	001000
18.8	8	14.0	8	1009.4	8	1008.2	8	4.9	6	26.8	8	35.0	999.9	24.8*	13.3*	0.04F	999.9	001000
17.6	8	11.7	8	1012.1	8	1010.9	8	12.3	6	9.7	8	16.9	999.9	18.7	16.0	0.06F	999.9	001000

Fig. 3 Weather Dataset.

It has all the necessary information to weather forecasting. The data are given in hourly basis for each station which has the same structure of data.Before running the MapReduce programs, the weather data has to be uploaded to the HDFS from the local storage. This can be done by executing the Hadoop command, hadoop fs -copyFromLocal <folder path> <name>.This command is executed in the Hadoop Command Line. To check whether the files has been uploaded in the HDFS, the Name Node Status can be used. 'Browse the files' option in the Name Node Status is used to view the contents stored in the Hadoop Distributed File System. After uploading the file to the HDFS, the jar file with MapReduce programs can be executed with the input and output file paths using the command,hadoop jar jarfilename.jar <inputfolderpath> <outputfilename>

The jar can be created using NetBeans or Eclipse IDE based on our own requirement. We have used NetBeans IDE to create the jar file. When writing our own jar file, the necessary library files has to imported in the program in order for the program to run without any error or exceptions. This Hadoop command runs the MapReduce program and if it executes without a error, the output file is generated with the result of the MapReduce program. The output can be viewed either by using the commands, >hadoop dfs -cat <outputfilename>/part-r-00000 or the NameNode Status can be used to view the output. The part-r-00000 file holds the output content as default. This is how a simple MapReduce program is executed in Hadoop environment.

A. Running Weather MapReduce Program

The dataset shown in Figure 3 is initially given as a input to a java program to remove the unwanted characters and data in it. The output produced by the java program is given as the input to the Hadoop. We have initially coded a java program that would split the MIN column from the weather dataset and save it in the output files. These output files are uploaded to the HDFS. The uploaded files are shown by the figure 4.

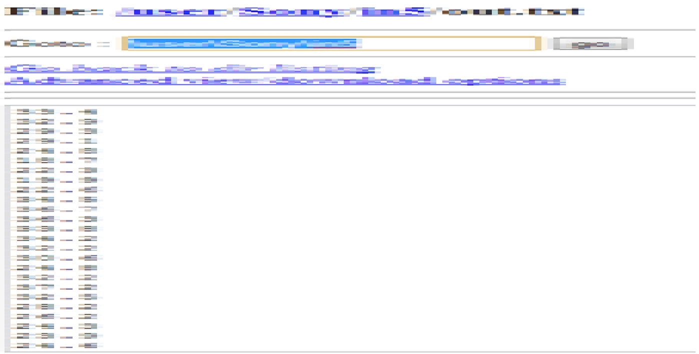
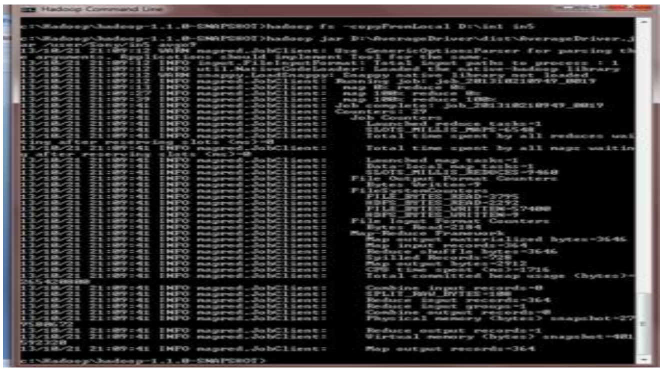


Fig. 4 Input File.

The MapReduce program gets these input files and in map phase the data are mapped into <key, value> pairs. The MapReduce program has a main class, a mapper class and a reducer class. When the jar file is called the main class runs which gets the input arguments and the mapper class is invoked here. The mapped input is passed onto the reducer phase which adds up all the values and calculates its average and writes it to the output file.

Figure 5 shows the execution of MapReduce program to find the average minimum temperature of the weather datasets. The input and output files has been mentioned in the hadoop command and when the program runs, the jobs are allotted to the nodes and the map phase runs generating the <key, value> pairs. When the mapping is completed 100% the reduce phase begins. The Average is been found in the reduce phase and the reduced output result is written into the output record by the output writer function. Since the average is a float variable here, the FloatWritable method is used in the program. The output is stored in the output folder which has a output file(part-r-00000) and the output log details in it.The output folder generated for the calculation of average of minimum temperature of the weather dataset is shown in figure 6

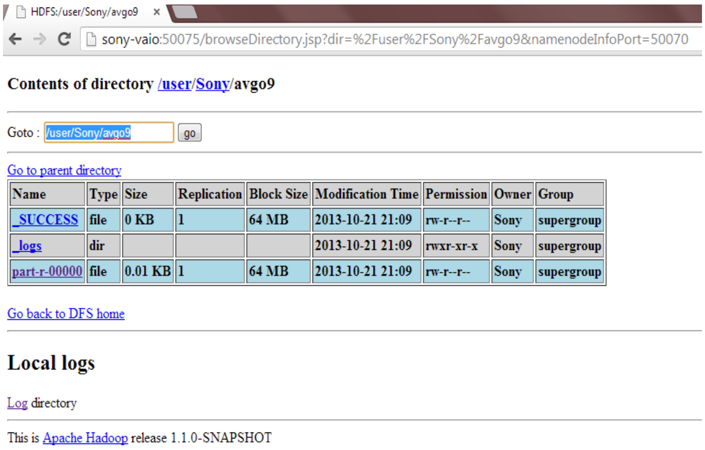


```
c:\Hadoop\hadoop-1.1.0-SNAPSHOT>hadoop fs -copyFromLocal D:\in1 in5

c:\Hadoop\hadoop-1.1.0-SNAPSHOT>hadoop jar D:\AverageDriver\dist\AverageDriver.j
ar /user/Sony/in5 avg09
13/10/21 21:09:12 WARN mapred.JobClient: Use GenericOptionsParser for parsing th
e arguments. Applications should implement Tool for the same.
13/10/21 21:09:12 INFO input.FileInputFormat: Total input paths to process : 1
13/10/21 21:09:12 INFO util.NativeCodeLoader: Loaded the native-hadoop library
13/10/21 21:09:12 WARN snappy.LoadSnappy: Snappy native library not loaded
13/10/21 21:09:12 INFO mapred.JobClient: Running job: job_201310210949_0019
13/10/21 21:09:13 INFO mapred.JobClient:  map 0% reduce 0%
13/10/21 21:09:27 INFO mapred.JobClient:  map 100% reduce 0%
13/10/21 21:09:39 INFO mapred.JobClient:  map 100% reduce 100%
13/10/21 21:09:41 INFO mapred.JobClient: Job complete: job_201310210949_0019
13/10/21 21:09:41 INFO mapred.JobClient: Counters: 29
13/10/21 21:09:41 INFO mapred.JobClient:   Job Counters
13/10/21 21:09:41 INFO mapred.JobClient:     Launched reduce tasks=1
13/10/21 21:09:41 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=6548
13/10/21 21:09:41 INFO mapred.JobClient:     Total time spent by all reduces wai
ting after reserving slots (ms)=0
13/10/21 21:09:41 INFO mapred.JobClient:     Total time spent by all maps waitin
g after reserving slots (ms)=0
13/10/21 21:09:41 INFO mapred.JobClient:     Launched map tasks=1
13/10/21 21:09:41 INFO mapred.JobClient:     Data-local map tasks=1
13/10/21 21:09:41 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=9460
13/10/21 21:09:41 INFO mapred.JobClient:   File Output Format Counters
13/10/21 21:09:41 INFO mapred.JobClient:     Bytes Written=9
13/10/21 21:09:41 INFO mapred.JobClient:   FileSystemCounters
13/10/21 21:09:41 INFO mapred.JobClient:     FILE_BYTES_READ=3795
13/10/21 21:09:41 INFO mapred.JobClient:     HDFS_BYTES_READ=2292
13/10/21 21:09:41 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=57400
13/10/21 21:09:41 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=9
13/10/21 21:09:41 INFO mapred.JobClient:   File Input Format Counters
13/10/21 21:09:41 INFO mapred.JobClient:     Bytes Read=2184
13/10/21 21:09:41 INFO mapred.JobClient:   Map-Reduce Framework
13/10/21 21:09:41 INFO mapred.JobClient:     Map output materialized bytes=3646
13/10/21 21:09:41 INFO mapred.JobClient:     Map input records=364
13/10/21 21:09:41 INFO mapred.JobClient:     Reduce shuffle bytes=3646
13/10/21 21:09:41 INFO mapred.JobClient:     Spilled Records=728
13/10/21 21:09:41 INFO mapred.JobClient:     Map output bytes=2912
13/10/21 21:09:41 INFO mapred.JobClient:     CPU time spent (ms)=1716
13/10/21 21:09:41 INFO mapred.JobClient:     Total committed heap usage (bytes)=
265420800
13/10/21 21:09:41 INFO mapred.JobClient:     Combine input records=0
13/10/21 21:09:41 INFO mapred.JobClient:     SPLIT_RAW_BYTES=108
13/10/21 21:09:41 INFO mapred.JobClient:     Reduce input records=364
13/10/21 21:09:41 INFO mapred.JobClient:     Reduce input groups=1
13/10/21 21:09:41 INFO mapred.JobClient:     Combine output records=0
13/10/21 21:09:41 INFO mapred.JobClient:     Physical memory (bytes) snapshot=27
9588672
13/10/21 21:09:41 INFO mapred.JobClient:     Reduce output records=1
13/10/21 21:09:41 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=401
6592320
13/10/21 21:09:41 INFO mapred.JobClient:     Map output records=364

c:\Hadoop\hadoop-1.1.0-SNAPSHOT>
```

Fig. 5 Execution of MapReduce Program.



HDFS:/user/Sony/avgo9
sony-vaio:50075/browseDirectory.jsp?dir=%2Fuser%2FSony%2Favgo9&namenodeInfoPort=50070

Contents of directory /user/Sony/avgo9

Goto : /user/Sony/avgo9 go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
_SUCCESS	file	0 KB	1	64 MB	2013-10-21 21:09	rw-r--r--	Sony	supergroup
_logs	dir				2013-10-21 21:09	rwxr-xr-x	Sony	supergroup
part-r-00000	file	0.01 KB	1	64 MB	2013-10-21 21:09	rw-r--r--	Sony	supergroup

Go back to DFS home

Local logs

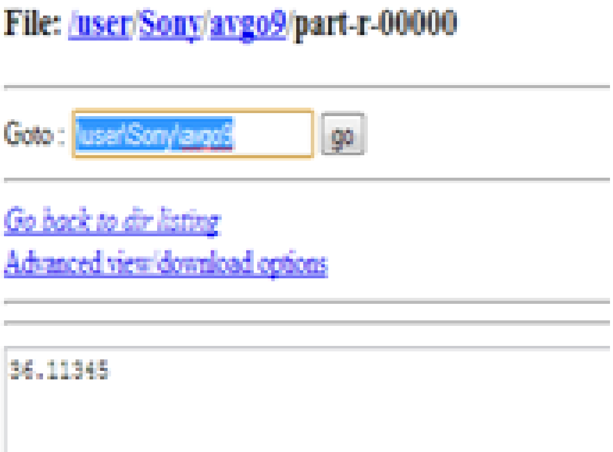
Log directory

This is Apache Hadoop release 1.1.0-SNAPSHOT

Fig. 6 Output Folder in HDFS.

The output file part-r-00000 holds the output which is shown in figure 7.

Similarly, the average maximum temperature, sea level pressure, dew point, station pressure, Mean visibility, Mean wind speed, Maximum sustained wind speed, average maximum wind gust, Precipitation amount and Snow depth can be calculated the using the Hadoop program.



File: /user/Sony/avgo9/part-r-00000

Goto : /user/Sony/avgo9 go

Go back to dir listing
Advanced view/download options

```
36.11345
```

Fig. 7 MapReduce Output.

Conclusion

Using Hadoop processing of large weather datasets has become simple and when this analysis is done with dynamic weather datasets, Hadoop can sure be a great tool in prediction of weather.

References

1. Kyong-Ha Lee, Yoon-Joon Lee, Hyunsik Choi, Yon Dohn Chung, Bongki Moon : Parallel Data Processing with MapReduce

2. Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler, Yahoo! : The Hadoop Distributed File System.

3. Chaiken,r.,Jenkins,b.,larson,P.,ramsey,b.,shakib,d.,Weaver,s.,andZhou,J.sCoPe:easyandefficientparallelprocessingofmassivedatasets.InProceedingsoftheConferenceonVeryLargeDatabases,2008.
4. K.Wang,J.Han,B.Tu,J.Dai,W.Zhou,andX.Song,"AcceleratingSpatialDataProcessingwithMapReduce",inProc.ICPADS,2010,pp.229-236
5. J. Dean, S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” In Proc. of the 6th Symposium on Operating Systems Design and Implementation, San Francisco CA, Dec. 2004.
6. Y. Bu et al . HaLoop: Efficient iterative data processing on large clusters. Proceedings of the VLDB Endowment, 3(1-2):285–296, 2010.
7. MapReduce: Simplified Data Processing on Large Clusters. Available at http://labs.google.com/papers/mapreduceosdi04.pdf
8. Fun with hadoop and mapreduce. Available at: http://snaggled.github.io/2010/10/29/Fun-with-MapReduce-Hadoop.html
9. M.Senthilkumar,Dr.p.ilango,"A Survey on Job Scheduling in Big Data",CYBERNETICS AND INFORMATION TECHNOLOGIES,vol.16,Issue.3,pp.35-51,2016.
10. M.Senthilkumar, Dr.p.ilango,"Analysis of DNA Data Using Hadoop Distributed File System",Research Journal of Pharmaceutical, Biological and Chemical Sciences,vol.7,Issue.3,pp.793-803,2016.