Frequent Pattern Identification using Map Reduce Paradigm

S. Vinodhini[1], M. Vinoth[2]
[1,2]Post Graduate Student
Department of Computer Science and Engineering
Sri Venkateswara College of Engineering, Chennai, India

M. K. Diwakar[3], M. Rajesh[4]
[3,4]Under Graduate Student
Department of Information Technology
P.B College of Engineering, Chennai, India

Abstract- **Data mining is a rapidly growing field and it has become a happening research area. Some of the sub research areas that comes under data mining are data classification, data clustering, text classification, frequent pattern mining, semantic web mining, ontology based mining etc. The proposed system comes under the frequent pattern mining and our objective is to find frequent patterns in order to increase the profit by making the frequent items available consistently. Initially dataset is collected from various online resources and volume of the dataset is big data. Multinode cluster is setup using various tools such as HortonWorks, VirtualBox, Putty and WinSCP in order to create Linux environment in Windows. Hadoop MapReduce paradigm that distributes the loads among the nodes inorder to reduce the time of computation is used. HortonWorks file browser is used to give the input and download the intermediate output. Token based approach is proposed in order to find the frequent pattern from the intermediate output. There are many existing methods for analyzing big data and frequent pattern but the proposed system is fault tolerant, time consuming and highly reliable.**

Keywords- Frequent Pattern Mining, Big Data Analysis, Mutinode Setup, Token Based Approach.

I. INTRODUCTION

Data Mining is an important aspect of every organization's growth. Every company has loads of data to be accessed and processed. Those data must be handled in a way such that there are no chances of any data loss. Data mining handles tasks such as anomaly detection, clustering, classification, regression, summarization, prediction, combinations, sequential patterns etc. Our project involves the task of determining frequent pattern from a given data set (i.e.) big data. Frequent patterns are required to be identified because of the hidden facts in the dataset. Frequent patterns can easily adapt to the data mining tasks and identifying the frequent pattern consumes less time. From a frequent pattern, the frequent items in the data sets can be identified and they also represent the relationship between the datasets.

We use the concept of Hadoop Mapreduce in our project, which is a free to use java based programming framework developed by Apache software foundation, for executing various applications involving multiple nodes of vast terabytes. In our project the dataset used are in form of text file, as Hadoop accepts only text files as input. The softwares that are being used are oracle virtual box for presenting the Linux environment in Windows. Hortonworks sandbox is used as the framework environment for running the Hadoop. Since we need a medium to create a connection with network with local host, the software putty is used to host the network, this allows us to host the Hadoop local host. We also need to secure our data from malwares and other spammers, for such purpose we are using Windows Secure Copy (WINSCP) which helps in providing secure transaction in the browser window.

MapReduce paradigm is being used in our project for parallel processing of large datasets. In MapReduce job the map task involves mapping the input among all the nodes, the output of which is fed as input into the reduce framework, which counts the total number of outcomes.

The content of paper ordered in the following manner. Section 1 describes about the introduction, Section 2 describes about the related works of our Existing system. Section 3 describes about the proposed system of this paper. Section 4 describes about the implementation. Section 5 describes about the conclusion.

II. RELATED WORKS

The frequent pattern mining is an active method used now a day to reduce and compare the candidate patterns. The project is based on identifying frequent candidate patterns, the association analysis is used to find comparisons between the patterns. In this paper a new candidate head set in initialized which can be splitted into smaller sets. This provides advancement in using the frequent mining pattern prior to the previous method. This project can be widely used by investors of cross selling markets [4].

The project involves about the various scheduling techniques that are being used by the Hadoop mapreduce concepts in cloud environments. The default Hadoop scheduler FIFO is used to initialize other scheduling techniques. The default scheduler includes a normal queue which has individual task that are progressed one after another. The other scheduling techniques that are initialized through the default Hadoop scheduler are fair scheduler and capacity scheduler. Advancements on scheduler improvements are provided by (LATE) Longest Approximate Time to End, Dynamic priority scheduling, Deadline constraint scheduler and Delay scheduling [5].

The project involves the usage of big data at enterprise data of YAHOO using the Hadoop Distributed File System. The work involves partitioning of data and computing hosts on YAHOO, the partitioning and computation is done using MapReduce Paradigm. It also involves using the various data nodes of HDFS client for initializing the process and also prevents from data loss [6].

The frequent pattern mining algorithm is used to identify required data from set of reviews provided by customers on various online shopping website for multiple products. The main advantage is that, this project provides advanced techniques in providing the outcome. The various techniques are Information Extraction, Association Rule Mining and the outcome of the project is evaluated by performance measuring using pruning effect and precision & recall trade off [7].

The paper involves the usage of various frequent pattern mining algorithms like Apriori algorithm, Partition based algorithm, DFS and hybrid based algorithms. These mining patterns techniques are compared based on various important aspects like comparison of Apriori and association mining technique and Apriori with hybrid [8].

III. SYSTEM DESIGN

The process involves identification of frequent patterns from the dataset. Hadoop has a unique file system called HDFS which virtually stores the big data in the form distributed files among various nodes. Multi-node setup can be formed using various tools and creating the environment as Linux. Frequent set is generated using MapReduce paradigm that distributes the loads among the nodes in order to reduce the time of computation. Finally prediction of frequent pattern and infrequent pattern elimination can be done using new token based approach.

The architecture diagram depicts the overall implementation of this work. The input dataset is uploaded into the browser, the architecture diagram consists of two modules.

A) Hadoop Common Package - It is used to start the hadoop process (Source code, documentation)
B) HDFS – It is used to store and access large datasets, which consists of a Name node to host file system and to take snapshots which prevents file system corruption.

Hortonworks sandbox is the testing environment, putty is used to host network within a system & WINSCP is the windows browser used for secure use of data. The Final output can be downloaded & viewed from the browser.

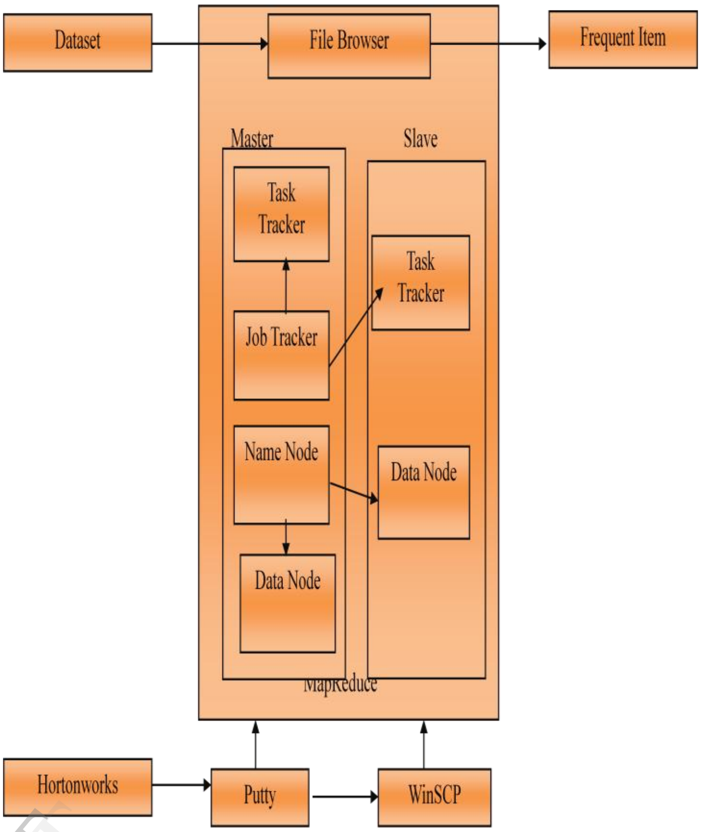


Fig. 1. Architecture of proposed system

A. Dataset Creation

The day, month & yearly sales of various shops are collected from various websites. These datasets must be in the text format. Hadoop can analyze data only when it is in text format. Thus the data's must be converted into text format.

B. Multi-node setup

The Process Hadoop can be accessed only in Linux, In our project to use Hadoop in Windows, we are bringing the concept of Multi-node setup. To make this possible we are using the following software.

a) Oracle Virtual Box- This software is used to bring the Linux environment in Windows.
b) Hortonworks Sandbox – We are importing this Linux operated software into the virtual box, this software is normally used as a testing environment.
c) Putty- Putty is an open source terminal emulator and network file transfer application. It can be used to make SSH connections with the server.
d) Windows Secure Copy- WinSCP is an open source SFTP, SCP and FTP client for Microsoft Windows. Inorder to make secure file transfers between local computer and remote computer, WinSCP offers basic file manager and file synchronization functionality. For secure transfers, it uses Secure Shell (SSH) and supports the SCP protocol.

C. MapReduce Job

The input data used is the sales sheet of multiple shops, usually a sales sheet consists of N number of data's of multiple counts. To sort and count the exact number of data's and its count, we are using this MapReduce job. The concept we are using is MapReduce Paradigm. This concept is a software pattern where Map is a separate function and Reduce is a separate function.

a) Map() - This function is used to sort the total no of data's available.
b) Reduce() – This function is used to display the total count value's of each data.

The Working procedure of MapReduce Job consists of,
a) Splitting
b) Mapping
c) Shuffling
d) Reducing

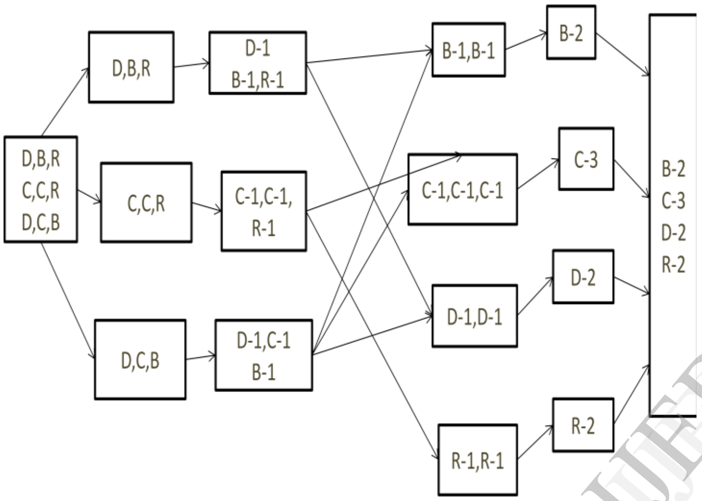


Fig. 2. MapReduce concept

D. Frequent Pattern Identification

The Frequent Pattern Identification module is used to display the frequent patterns from the output of MapReduce Job module. MapReduce module can only sort and count the available data, to display the frequent pattern; frequent pattern identification module is used. A Threshold value can be set by the user to eliminate unwanted data's. In this module, we are using an approach called Token Based approach, which is used to produce accurate frequent pattern's of data. The input of this module is the outputs from the previous module. The output displayed after the elimination of the threshold value is the frequent pattern.

E. Performance Evaluation

This module is used to evaluate the performance of our project to evaluate the performance, we use Confusion matrix, and this matrix provides the accuracy of the patterns.

TABLE 1 Confusion matrix

Confusion Matrix	Frequent	Infrequent
Frequent	True Positive	False Negative
Infrequent	False Positive	True Negative

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}$$

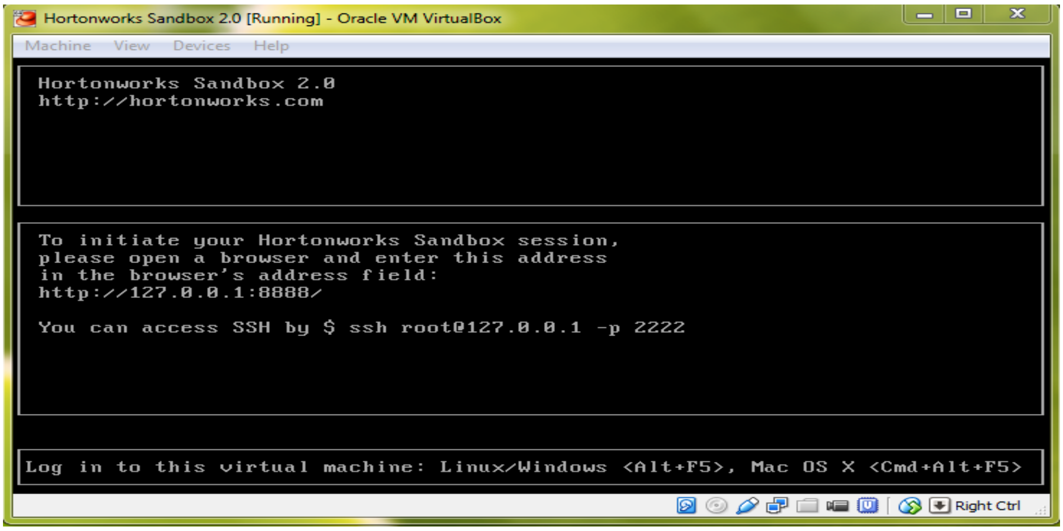
$$\text{Sensitivity (Recall)} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

IV. IMPLEMENTATION RESULTS

This proposed system is implemented in java in order to predict the frequent pattern from big data. Initially Oracle virtual box installed in windows 7 and Hortonworks is installed into oracle virtual box as multi node. In this Hortonworks, we can increase the number of processor to increase the performance speed of proposed system and we can increase the memory space of total number of processor. Putty and WinSCP are used to connect the Hortonworks and into file browser. Putty is network administering tool which is used to make the Linux run able environment is windows. WinSCP is security component protocol which is used to store the files. File browser is used to store to input into folder or as a file and it is used to download the output.

Figure 3 shows that starting of Hortonworks in order to produce the port and ip address to connect with putty and WinSCP. Figure 4 shows that configuration of putty, we need to enter port number and ip address which is created in Hortonworks. Figure 5 shows that contents displayed in WinSCP, we need to enter port number and ip address which created in Hortonworks. Username and password have entered for authentication purpose in order to connect with file browser. Figure 6 shows that running MapReduce program , Figure 7 shows that file browser, Figure 8 shows that Frequent pattern identification.



Hortonworks Sandbox 2.0 [Running] - Oracle VM VirtualBox
Machine View Devices Help
Right Ctrl

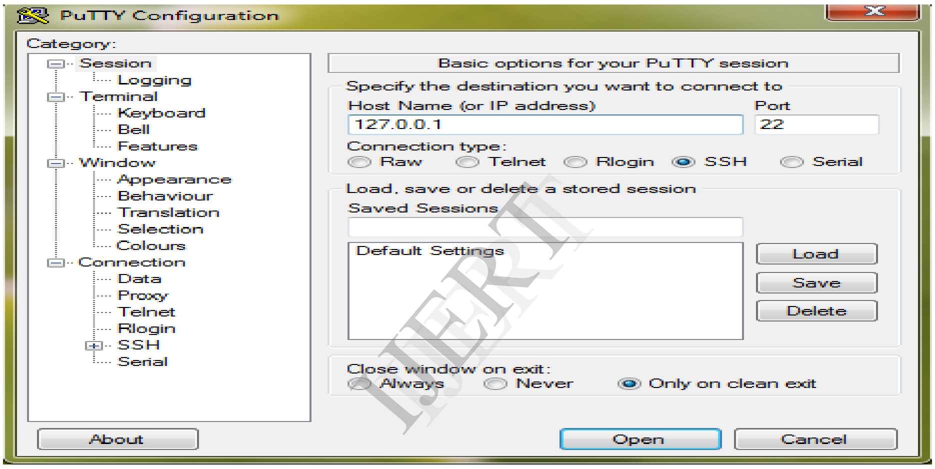
```
Hortonworks Sandbox 2.0
http://hortonworks.com

To initiate your Hortonworks Sandbox session,
please open a browser and enter this address
in the browser's address field:
http://127.0.0.1:8888/

You can access SSH by $ ssh root@127.0.0.1 -p 2222

Log in to this virtual machine: Linux/Windows <Alt+F5>, Mac OS X <Cmd+Alt+F5>
```

Fig. 3. Stating Hortonworks



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 127.0.0.1

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

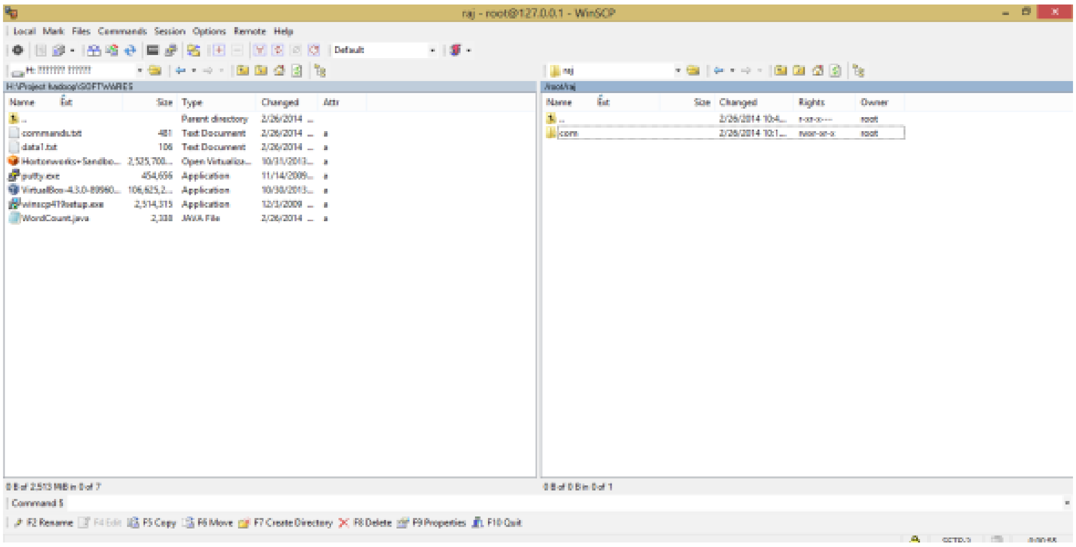
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Fig. 4. Putty configuration



H:\Project hadoop\SOFTWARES

Name Ext	Size	Type	Changed	Attr
..		Parent directory	2/26/2014 ...	
commands.txt	481	Text Document	2/26/2014 ...	a
data1.txt	106	Text Document	2/26/2014 ...	a
Hortonworks+Sandbo...	2,525,700...	Open Virtualiza...	10/31/2013...	a
putty.exe	454,656	Application	11/14/2009...	a
VirtualBox-4.3.0-89960...	106,625,2...	Application	10/30/2013...	a
winscp419setup.exe	2,514,315	Application	12/1/2009 ...	a
WordCount.java	2,338	JAVA File	2/26/2014 ...	a

/root/raj

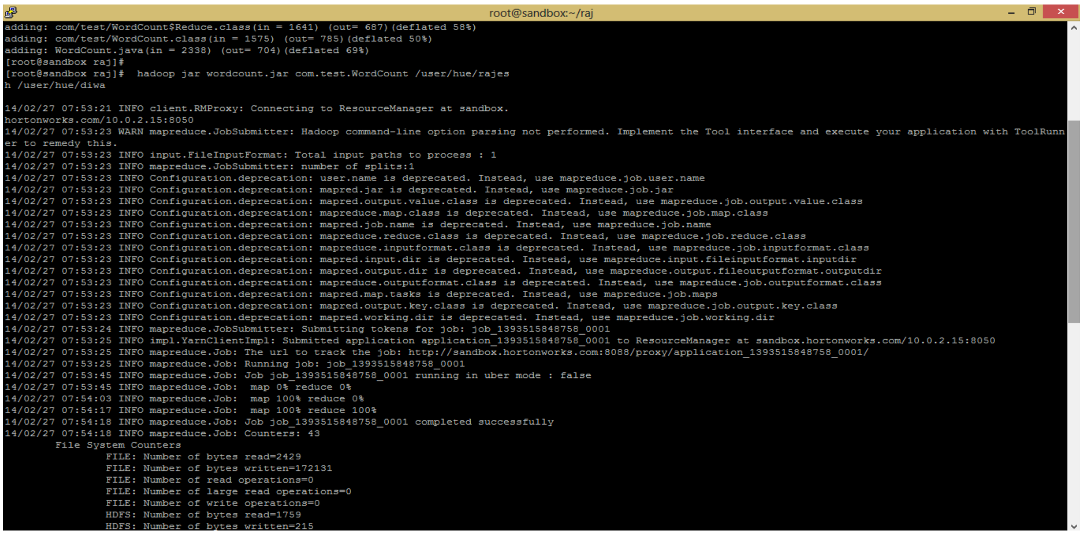
Name Ext	Size	Changed	Rights	Owner
..		2/26/2014 10:4...	r-xr-x---	root
com		2/26/2014 10:1...	rwxr-xr-x	root

0 B of 2,513 MB in 0 of 7

0 B of 0 B in 0 of 1

Command $

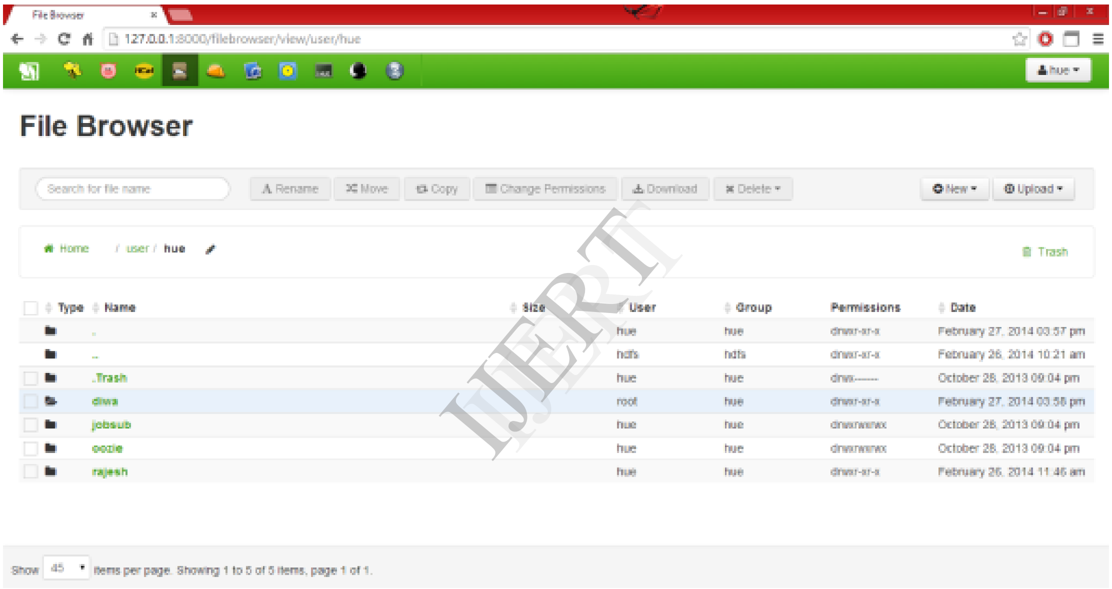
Fig. 6. Contents of the client system displayed by WinSCP



```
adding: com/test/WordCount$Reduce.class(in = 1641) (out= 687)(deflated 58%)
adding: com/test/WordCount.class(in = 1575) (out= 785)(deflated 50%)
adding: WordCount.java(in = 2338) (out= 704)(deflated 69%)
[root@sandbox raj]#
[root@sandbox raj]#  hadoop jar wordcount.jar com.test.WordCount /user/hue/rajes
h /user/hue/diwa

14/02/27 07:53:21 INFO client.RMProxy: Connecting to ResourceManager at sandbox.
hortonworks.com/10.0.2.15:8050
14/02/27 07:53:23 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunn
er to remedy this.
14/02/27 07:53:23 INFO input.FileInputFormat: Total input paths to process : 1
14/02/27 07:53:23 INFO mapreduce.JobSubmitter: number of splits:1
14/02/27 07:53:23 INFO Configuration.deprecation: user.name is deprecated. Instead, use mapreduce.job.user.name
14/02/27 07:53:23 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
14/02/27 07:53:23 INFO Configuration.deprecation: mapred.output.value.class is deprecated. Instead, use mapreduce.job.output.value.class
14/02/27 07:53:23 INFO Configuration.deprecation: mapreduce.map.class is deprecated. Instead, use mapreduce.job.map.class
14/02/27 07:53:23 INFO Configuration.deprecation: mapred.job.name is deprecated. Instead, use mapreduce.job.name
14/02/27 07:53:23 INFO Configuration.deprecation: mapreduce.reduce.class is deprecated. Instead, use mapreduce.job.reduce.class
14/02/27 07:53:23 INFO Configuration.deprecation: mapreduce.inputformat.class is deprecated. Instead, use mapreduce.job.inputformat.class
14/02/27 07:53:23 INFO Configuration.deprecation: mapred.input.dir is deprecated. Instead, use mapreduce.input.fileinputformat.inputdir
14/02/27 07:53:23 INFO Configuration.deprecation: mapred.output.dir is deprecated. Instead, use mapreduce.output.fileoutputformat.outputdir
14/02/27 07:53:23 INFO Configuration.deprecation: mapreduce.outputformat.class is deprecated. Instead, use mapreduce.job.outputformat.class
14/02/27 07:53:23 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
14/02/27 07:53:23 INFO Configuration.deprecation: mapred.output.key.class is deprecated. Instead, use mapreduce.job.output.key.class
14/02/27 07:53:23 INFO Configuration.deprecation: mapred.working.dir is deprecated. Instead, use mapreduce.job.working.dir
14/02/27 07:53:24 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1393515848758_0001
14/02/27 07:53:25 INFO impl.YarnClientImpl: Submitted application application_1393515848758_0001 to ResourceManager at sandbox.hortonworks.com/10.0.2.15:8050
14/02/27 07:53:25 INFO mapreduce.Job: The url to track the job: http://sandbox.hortonworks.com:8088/proxy/application_1393515848758_0001/
14/02/27 07:53:25 INFO mapreduce.Job: Running job: job_1393515848758_0001
14/02/27 07:53:45 INFO mapreduce.Job: Job job_1393515848758_0001 running in uber mode : false
14/02/27 07:53:45 INFO mapreduce.Job:  map 0% reduce 0%
14/02/27 07:54:03 INFO mapreduce.Job:  map 100% reduce 0%
14/02/27 07:54:17 INFO mapreduce.Job:  map 100% reduce 100%
14/02/27 07:54:18 INFO mapreduce.Job: Job job_1393515848758_0001 completed successfully
14/02/27 07:54:18 INFO mapreduce.Job: Counters: 43
        File System Counters
                FILE: Number of bytes read=2429
                FILE: Number of bytes written=172131
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1759
                HDFS: Number of bytes written=215
```

Fig. 6. Running MapReduce Job

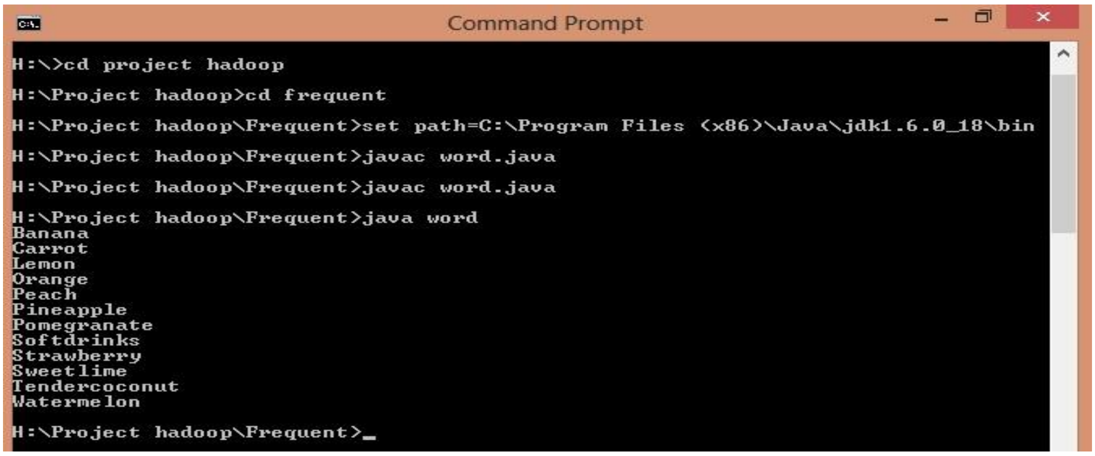


File Browser

	Type	Name	Size	User	Group	Permissions	Date
		.		hue	hue	drwxr-xr-x	February 27, 2014 03:57 pm
		..		hdfs	hdfs	drwxr-xr-x	February 26, 2014 10:21 am
□		.Trash		hue	hue	drwx------	October 28, 2013 09:04 pm
□		diwa		root	hue	drwxr-xr-x	February 27, 2014 03:58 pm
□		jobsub		hue	hue	drwxrwxrwx	October 28, 2013 09:04 pm
□		oozie		hue	hue	drwxrwxrwx	October 28, 2013 09:04 pm
□		rajesh		hue	hue	drwxr-xr-x	February 26, 2014 11:46 am

Show 45 items per page. Showing 1 to 5 of 5 items, page 1 of 1.

Fig. 7. File browser of Hortonworks



```
H:\>cd project hadoop

H:\Project hadoop>cd frequent

H:\Project hadoop\Frequent>set path=C:\Program Files (x86)\Java\jdk1.6.0_18\bin

H:\Project hadoop\Frequent>javac word.java

H:\Project hadoop\Frequent>javac word.java

H:\Project hadoop\Frequent>java word
Banana
Carrot
Lemon
Orange
Peach
Pineapple
Pomegranate
Softdrinks
Strawberry
Sweetlime
Tendercoconut
Watermelon

H:\Project hadoop\Frequent>_
```

Fig. 8. Frequent pattern identifiaction

Initially dataset is considered as input which has 100 transaction of juice shop. This dataset contains Item with different count which we cannot calculate manually. Table 2 shows that Item and Count of each of item. Initial dataset is pushed into file browser and count of each item produces using MapReduce paradigm programming concepts.

TABLE 2 Item and Count of each item

ITEM WITH COUNT			
Apple	4	Papaya	4
Banana	9	Peach	6
Carrot	6	Pineapple	7
Cranberry	5	Pomegranate	8
Grape	4	Softdrinks	17
Guava	4	Strawberry	10
Jackfruit	3	Sweetlime	8
Lemon	11	Tendercoconut	20
Mango	4	Tomato	4
Orange	17	Watermelon	8

The output of MapReduce program is item and its count should be analyzed in order to produce the frequent item form it. New token based approach is created to finding frequent item form intermediate output which contains item and count of each item.xThis token based approach takes input as file of intermediate output. Threshold value can fixed to eliminate the infrequent item. In this proposed system threshold value is 5.

TABLE 3 Frequent Item form above table

Frequent Item
Banana 9
Carrot 6
Lemon 11
Orange 17
Peach 6
Pineapple 7
Pomengranatte 8
Softdrinks 17
Strawberry 10
Sweetlime 8
Tendercoconut 20
Watermelon 8

TABLE 4 Confusion matrix values

		Predicted	
	CONFUSION MATRIX	Frequent	Infrequent
Actual	Frequent	**10**	**1**
	Infrequent	**0**	**9**

TABLE 5 Performance of proposed system

ACCURACY	95
SENSITIVITY	90.9090
PRECISION	100

5. CONCLUSION

Big Data is analyzed using Hortonworks by utilizing the concept of Hadoop distributed file system and it can be seen that it takes less time when comparing to existing system. Big Data is considered as data set which may be the sales sheet of shops. In order to find the frequent pattern of the dataset, Mapreduce paradigm and token based approach is proposed. The accuracy of the proposed system is 95% which is more sufficient to produce the accurate result and predict the frequent items to increase the profit. This system is more reliable and fault tolerant when compared to existing system.

REFERENCES

[1] Feng Xie., Zhen Chen., Hongfeng Xu., Xiwei Feng and Qi Hou (2013), ‘TST: Threshold Based Similarity Transitivity Method in Collaborative Filtering with Cloud Computing’, IEEE Transactions on Tsinghua Science and Technology, Vol. 18, No. 3, pp 318-327.

[2] Shvachko, K.; Hairong Kuang; Radia, S.; Chansler, R.; , "The Hadoop Distributed File System," Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on , vol., no., pp.1-10, 3-7 May 2010

[3] Jiong Xiea, FanJun Meng, HaiLong Wang, HongFang Pan, JinHong Cheng, Xiao Qina, (2013), Research on Scheduling Scheme for Hadoop clusters, International Conference on Computational Science, Vol. 18, pp. 2468 – 2471.

[4] M.H Nadimi-Shahraki and Norwati Mustapha, “ Efficient Candidacy Reduction for Frequent Pattern Mining”, International Journal of computer science and information security, Vol. 6, No. 3, 2009.

[5] B.Thirumala Rao and Dr. L.S.S.Reddy, “ Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments”, International Journal of Computer Applications, Vol. 34, No.9, 2011.

[6] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler, “The Hadoop Distributed File System”, IEEE,2010.

[7] Seyed Hamid Ghorashi, Roliana Ibrahim, Shirin Noekhah and Niloufar Salehi Dastjerdi, “A Frequent Pattern Mining Algorithm for Feature Extraction of Customer Reviews”, International Journal of Computer Science Issues, Vol. 9, Issue 4, No. 1,2012.

[8] Deepak Garg and Hemant Sharma, “ Comparative Analysis of Various Approaches Used in Frequent Pattern Mining”, International Journal of Advanced Computer Science and Applications,Special issue on Artificial Intelligence.