International Journal of Engineering Research & Management Technology ISSN: 2348-4039

November-2015 Volume 2, Issue-6

Email: editor@ijermt.org www.ijermt.org

Modeling Big Data Characteristics for Discovering Actionable Knowledge

Sasi Krishna .T, A. Sivva Kumar
Computer Science and Engineering
Jawaharlal Nehru Technological University
Hyderabad

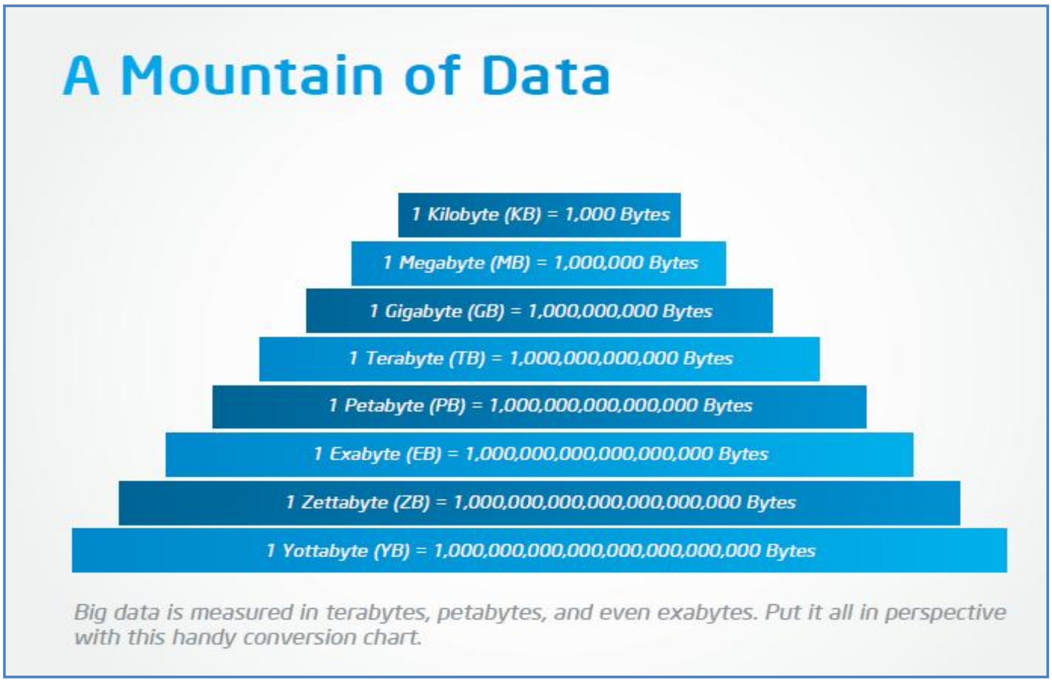
ABSTRACT:

Big data refers to the data is which is voluminous with other characteristics like velocity and variety. By mining big data it is possible to have comprehensive business intelligence that can be used to make well informed decisions. Enterprises in the real world have been using data mining. However, for full fledged business intelligence and accurate decision making, it is inevitable to make use of big data analytics. This will give them competitive advantages. Towards this end in this paper we make use of big data mining with Hadoop which is one of the distributed programming frameworks. We proposed a model that characterizes big data. We built a prototype application that demonstrates the mining of big data for actionable knowledge. The empirical results revealed that our application is useful for mining business intelligence from big data. However, the prototype is simple and it can be extended in future to support many algorithms that can be used to mine patterns that were not known earlier.

KEYWORD: Big Data, Data mining, Challenging issues, Datasets, Data Mining Algorithms

INTRODUCTION:

Big data can create value in terms of transparency, experimental analysis, business intelligence, real time analysis and decision making, and computer-assisted innovative solutions. Big data mining can improve prediction or forecast accuracy that adds big value to businesses. Internet communications, social networks, human digital universe, web search engines are some of the applications that produce Big data. The existing techniques such as machine learning techniques, unstructured data analytics, visualization, data mining, cloud computing, graph and mesh algorithms, and joining algorithms do not scale to processing Big data in Zettabytes. Big data can bring about many advantages such as increasing operational efficiency, strategic planning, better customer service, identifying customer needs, enhanced customer experience, identifying new markets, penetration into new markets, complying with regulations, and others. The knowledge discover process in big data mining includes phases such as data recording, data cleaning, dat analysis, data visualization and interpretation, and decision making. Big data techniques that bring abut these advantages can be classified into mathematical techniques, data analysis techniques, and Big data applications. With respect to stream processing Big data tools available are Storm, S4, SQL Stream Server, Splunk, Apache Kafka and SAP Hana. Figure 1 conceptually shows big data.


A Mountain of Data
1 Kilobyte (KB) = 1,000 Bytes
1 Megabyte (MB) = 1,000,000 Bytes
1 Gigabyte (GB) = 1,000,000,000 Bytes
1 Terabyte (TB) = 1,000,000,000,000 Bytes
1 Petabyte (PB) = 1,000,000,000,000,000 Bytes
1 Exabyte (EB) = 1,000,000,000,000,000,000 Bytes
1 Zettabyte (ZB) = 1,000,000,000,000,000,000,000 Bytes
1 Yottabyte (YB) = 1,000,000,000,000,000,000,000,000 Bytes

Big data is measured in terabytes, petabytes, and even exabytes. Put it all in perspective with this handy conversion chart.

Fig. 1 – Big data is measured in Peta bytes or higher

To process big data Hadoop is used. Hadoop is one of the distributed programming frameworks that are widely used in the real world. It is based on the new programming paradigm named “Map Reduce”. Bu et al. [9] studied distributed programming frameworks and proposed a new framework known as Haloop which a variant of Hadoop. Haloop extends the capabilities of Map Reduce with additional features like task scheduler, loop awareness and cashing. The frameworks like Hadoop & Haloop provide scalable solutions that help in processing Big data. Hadoop is being used by companies like Google, Yahoo and Facebook.

In Hadoop Map Reduce is used. Map Reduce is a new programming paradigm for big data mining. It has two phases. In the first phase, for each chunk a **Mapper** reads data, computes it and returns a list of key/value pairs. In the second phase of Map Reduce, a **Reducer** combines values pertaining to all distinct keys and the result is returned. In this context securing mappper in presence of untrusted mapper is an important approach to avoid security issues in big data processing. When mapper is compromised, it can produce unwanted output. Securing mapper is the challenging task that can help secure big data mining. Enterprise are producing huge amount of data every day. The data is grown exponentially and characterized by volume velocity and variety. Mining such data provides accurate business intelligence. When some part of data is mined, the knowledge obtained may not have complete ability to make decisions. The ability to make accurate decisions is possible when whole data is processed and come up with trends or patterns that show interesting facts useful for businesses. In this process there are security implications when subjected to Map Reduce programming. When mapper is compromised it is possible to get unrelated results. Adversaries can compromise mapper part of the new programming paradigm which is widely used in the real world now. When the opportunities and security implications are of big data mining are known it is possible to take well informed decisions. This is the motivation behind taking up this work.

In this paper we characterize big data and built a prototype application to demonstrate the mining of big data in order to discover actionable knowledge. The remainder of the paper is structured as follows. Section II provides a review of literature Section III presents the proposed implementation and results. Section IV concludes the paper.

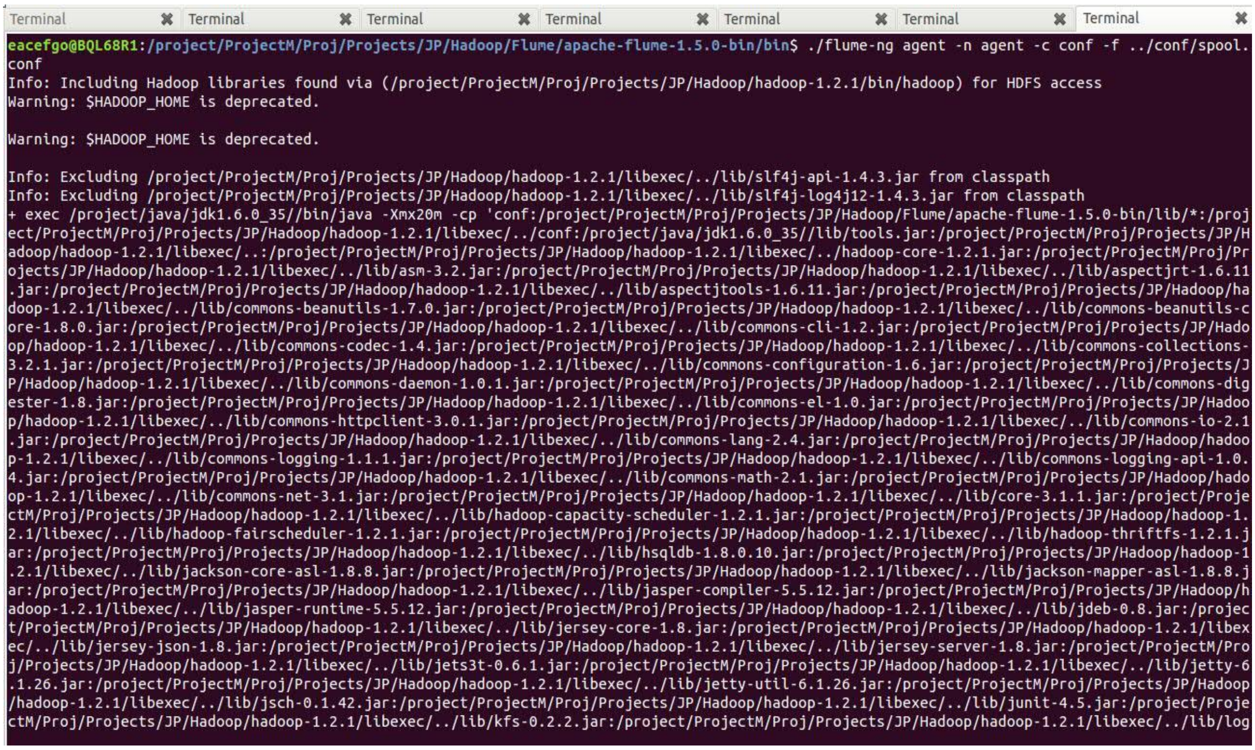
RELATED WORKS:

On the level of mining platform sector, at present, parallel programming models like Map Reduce are being used for the purpose of analysis and mining of data. Map Reduce is a batch-oriented parallel computing model. There is still a certain gap in performance with relational databases. Improving the performance of Map Reduce and enhancing the real-time nature of large-scale data processing have received a significant amount of attention, with Map Reduce parallel programming being applied to many machine learning and data mining algorithms. Data mining algorithms usually need to scan through the training data for obtaining the statistics to solve or optimize model.

For those people, who intend to hire a third party such as auditors to process their data, it is very important to have efficient and effective access to the data. In such cases, the privacy restrictions of user may be faces like no local copies or downloading allowed, etc. So there is privacy-preserving public auditing mechanism proposed for large scale data storage.[1] This public key-based mechanism is used to enable third-party auditing, so users can safely allow a third party to analyze their data without breaching the security settings or compromising the data privacy. In case of design of data mining algorithms, Knowledge evolution is a common phenomenon in real world systems. But as the problem statement differs, accordingly the knowledge will differ. For example, when we go to the doctor for the treatment, that doctor's treatment program continuously adjusts with the conditions of the patient. Similarly the knowledge For this, Wu [2] [3] [4] proposed and established the theory of local pattern analysis, which has laid a foundation for global knowledge discovery in multisource data mining. This theory provides a solution not only for the problem of full search, but also for finding global models that traditional mining methods cannot find.

IMPLEMENTATION:

We built a prototype application to demonstrate the concept of mining big data. We used Apache Hadoop to have a distributed programming framework with Hadoop Distributed File System (HDFS). Our application demonstrates clustering with big data that can be used to group similar objects.



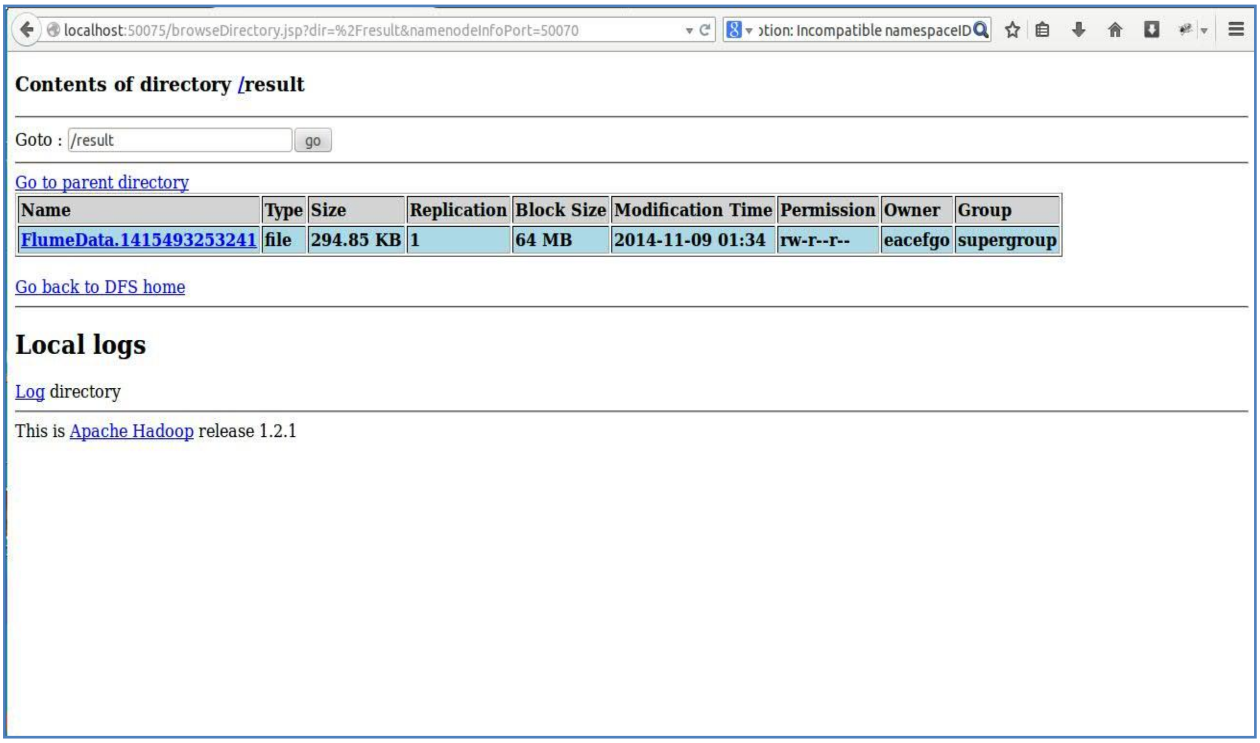
```
eacefgo@BQL68R1:/project/ProjectM/Proj/Projects/JP/Hadoop/Flume/apache-flume-1.5.0-bin/bin$ ./flume-ng agent -n agent -c conf -f ../conf/spool.
conf
Info: Including Hadoop libraries found via (/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/bin/hadoop) for HDFS access
Warning: $HADOOP_HOME is deprecated.

Warning: $HADOOP_HOME is deprecated.

Info: Excluding /project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/slf4j-api-1.4.3.jar from classpath
Info: Excluding /project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/slf4j-log4j12-1.4.3.jar from classpath
+ exec /project/java/jdk1.6.0_35//bin/java -Xmx20m -cp 'conf:/project/ProjectM/Proj/Projects/JP/Hadoop/Flume/apache-flume-1.5.0-bin/lib/*:/proj
ect/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../conf:/project/java/jdk1.6.0_35//lib/tools.jar:/project/ProjectM/Proj/Projects/JP/H
adoop/hadoop-1.2.1/libexec/..:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../hadoop-core-1.2.1.jar:/project/ProjectM/Proj/Pr
ojects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/asm-3.2.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/aspectjrt-1.6.11
.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/aspectjtools-1.6.11.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/ha
doop-1.2.1/libexec/../lib/commons-beanutils-1.7.0.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/commons-beanutils-c
ore-1.8.0.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/commons-cli-1.2.jar:/project/ProjectM/Proj/Projects/JP/Hado
op/hadoop-1.2.1/libexec/../lib/commons-codec-1.4.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/commons-collections-
3.2.1.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/commons-configuration-1.6.jar:/project/ProjectM/Proj/Projects/J
P/Hadoop/hadoop-1.2.1/libexec/../lib/commons-daemon-1.0.1.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/commons-dig
ester-1.8.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/commons-el-1.0.jar:/project/ProjectM/Proj/Projects/JP/Hadoo
p/hadoop-1.2.1/libexec/../lib/commons-httpclient-3.0.1.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/commons-io-2.1
.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/commons-lang-2.4.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoo
p-1.2.1/libexec/../lib/commons-logging-1.1.1.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/commons-logging-api-1.0.
4.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/commons-math-2.1.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hado
op-1.2.1/libexec/../lib/commons-net-3.1.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/core-3.1.1.jar:/project/Proje
ctM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/hadoop-capacity-scheduler-1.2.1.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.
2.1/libexec/../lib/hadoop-fairscheduler-1.2.1.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/hadoop-thriftfs-1.2.1.j
ar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/hsqldb-1.8.0.10.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1
.2.1/libexec/../lib/jackson-core-asl-1.8.8.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/jackson-mapper-asl-1.8.8.j
ar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/jasper-compiler-5.5.12.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/h
adoop-1.2.1/libexec/../lib/jasper-runtime-5.5.12.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/jdeb-0.8.jar:/projec
t/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/jersey-core-1.8.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libex
ec/../lib/jersey-json-1.8.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/jersey-server-1.8.jar:/project/ProjectM/Pro
j/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/jets3t-0.6.1.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/jetty-6
.1.26.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/jetty-util-6.1.26.jar:/project/ProjectM/Proj/Projects/JP/Hadoop
/hadoop-1.2.1/libexec/../lib/jsch-0.1.42.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/junit-4.5.jar:/project/Proje
ctM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/kfs-0.2.2.jar:/project/ProjectM/Proj/Projects/JP/Hadoop/hadoop-1.2.1/libexec/../lib/log
```

Figure 2 – Launching the distributed programming framework

As seen in Figure 2, it is evident that the Hadoop environment is being launched and the underlying programming paradism is "MapReduce".



localhost:50075/browseDirectory.jsp?dir=%2Fresult&namenodeInfoPort=50070

Contents of directory /result

Goto : /result go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
FlumeData.1415493253241	file	294.85 KB	1	64 MB	2014-11-09 01:34	rw-r--r--	eacefgo	supergroup

Go back to DFS home

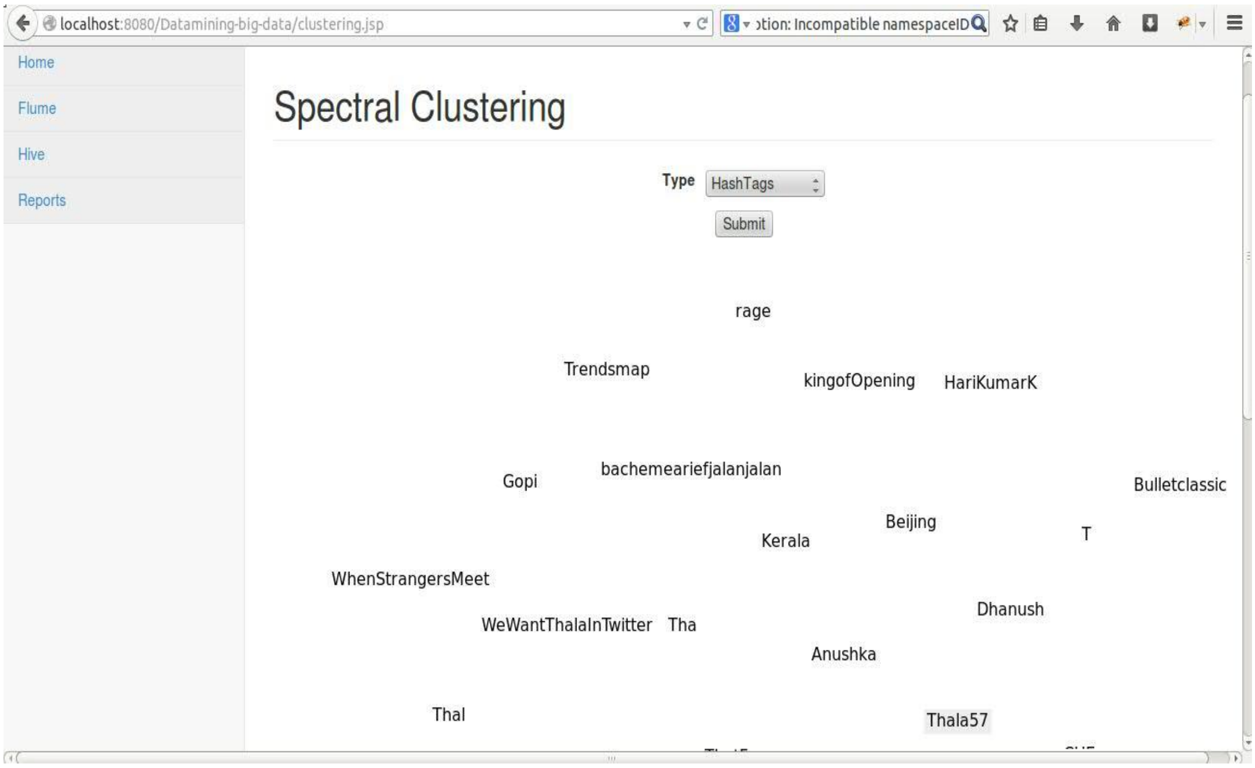
Local logs

Log directory

This is Apache Hadoop release 1.2.1

Figure 3 – Web intrface for hadoop

As shown in Figure 3, it is evident that the web based interface for Apache Hadoop. The interface can be used to view the infromation abou the framework and the undelrying file system.



localhost:8080/Datamining-big-data/clustering.jsp

Spectral Clustering

Type HashTags

Figure 4 – Web based inteface for spectral clustering

Spectral clustering is performed to mine data. Before actually making real clusters spectral clustering can help reducing dimensions in order to improve the performance of clustering The performance of the clustering is as shown in the graphs below.

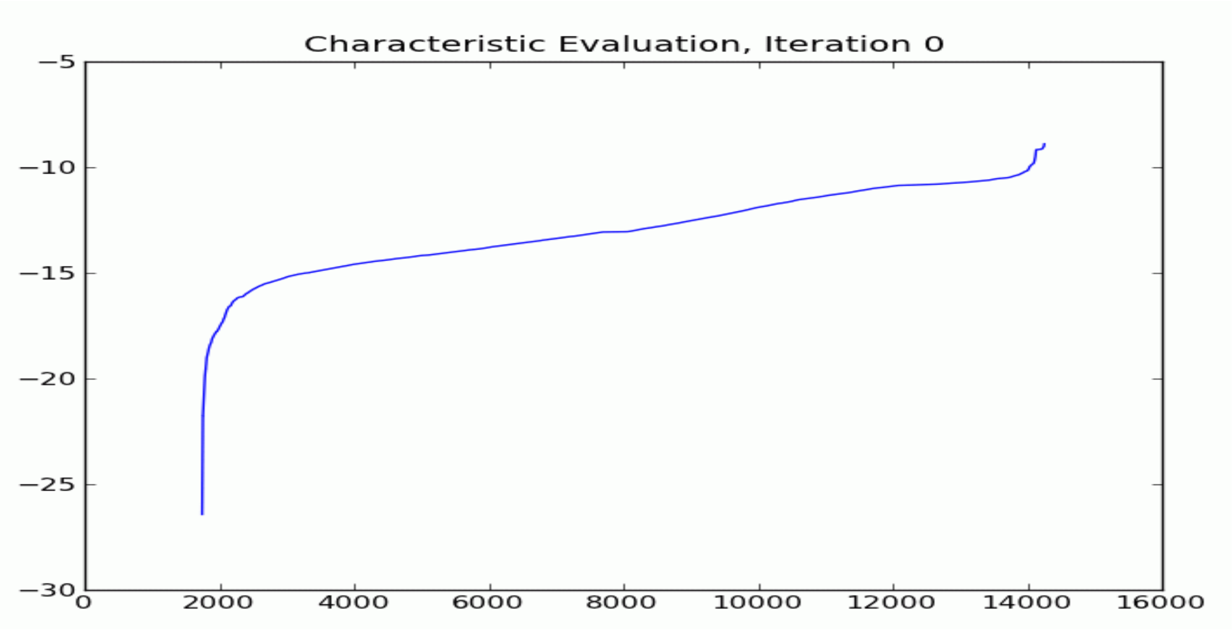


Figure 5 – Characteristic evaluation 0

As can be seen in Figure 5, it is evident that the iterations in the spectral clustering are evaluated. The graph here shows the evaluation of iteration 0.

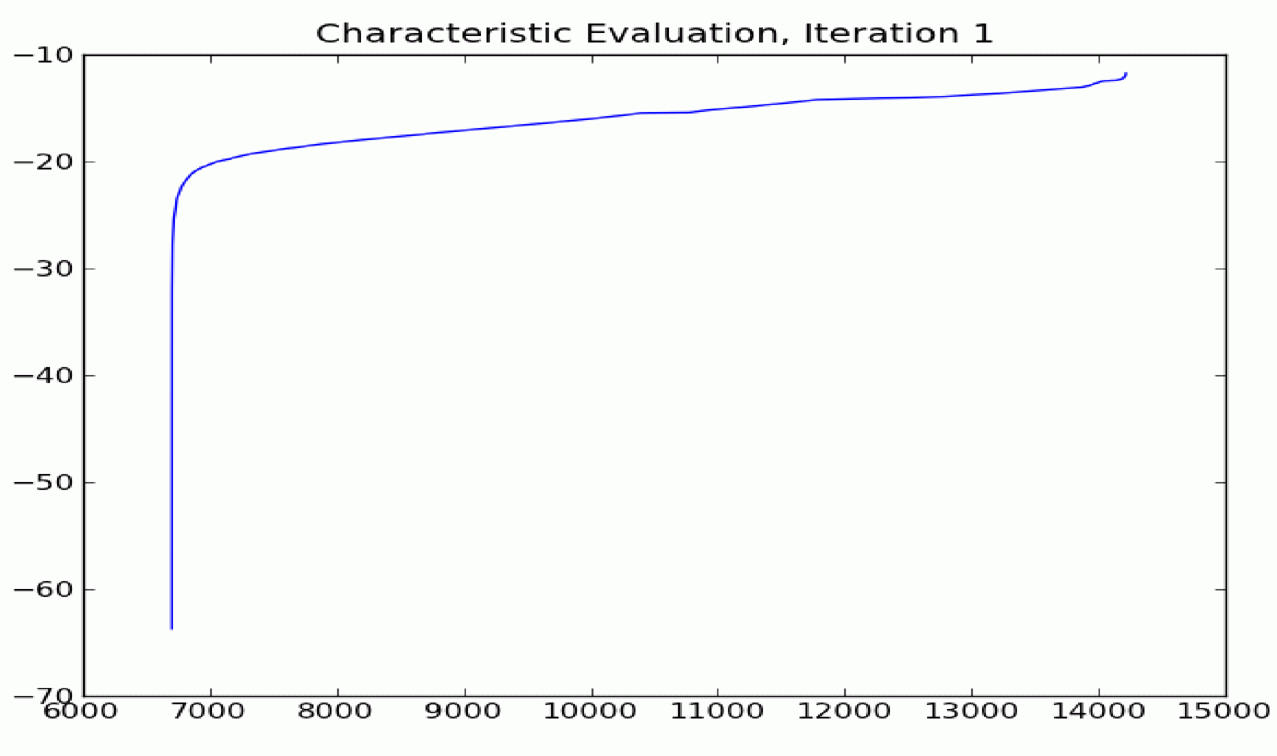


Figure 6 - Characteristic Evaluation Iteration 1

As can be seen in Figure 6, it is evident that the iterations in the spectral clustering are evaluated. The graph here shows the evaluation of iteration 1.

CONCLUSION:

Big data has become a buzzword these days. As data is growing exponentially big data concepts and big data analytics become very important for real world businesses. It is possible to have big data mined using distributed programming frameworks like Hadoop. The programming paradigm used for mining big data is Map Reduce. The file system used in Apache Hadoop is HDFS. The aim of this paper is to characterize big data and demonstrate the usefulness of mining such data. Towards this end, we built a prototype application that demonstrates the mining on big data in a distributed environment. The results revealed that the proposed approach is useful to extended it further to support more useful algorithms in future for big data analytics.

REFERENCES:

1. C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy- Preserving Public Auditing for Secure Cloud Storage” IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.
2. X. Wu and S. Zhang, “Synthesizing High-Frequency Rules from Different Data Sources,” IEEE Trans. Knowledge and Data Eng., vol. 15, no. 2, pp. 353-367, Mar./Apr. 2003.
3. X. Wu, C. Zhang, and S. Zhang, “Database Classification for Multi-Database Mining,” Information Systems, vol. 30, no. 1, pp. 71- 88, 2005
4. K. Su, H. Huang, X. Wu, and S. Zhang, “A Logical Framework for Identifying Quality Knowledge from Different Data Sources,” Decision Support Systems, vol. 42, no. 3, pp. 1673-1683, 2006.
5. E.Y. Chang, H. Bai, and K. Zhu, “Parallel Algorithms for Mining Large-Scale Rich-Media Data,” Proc. 17th ACM Int’l Conf. Multimedia, (MM ’09,) pp. 917-918, 2009.
6. D. Howe et al., “Big Data: The Future of Biocuration,” Nature, vol. 455, pp. 47-50, Sept. 2008.

a. Labrinidis and H. Jagadish, “Challenges and Opportunities with Big Data,” Proc. VLDB Endowment, vol. 5, no. 12, 2032-2033, 2012.

7. Y. Lindell and B. Pinkas, “Privacy Preserving Data Mining,” J. Cryptology, vol. 15, no. 3, pp. 177-206, 2002.
8. Tingyi Bu, Bill Howe, Magdalena Balazinska and Michael D. Ernst. 2010. *HaLoop: Efficient Iterative Data Processing on Large Clusters*. USA: IEEE. p1-12.