Simple Analytics in Retail Sales Data Using Hadoop

R.R. Karthikeyan [1], Dr. B Raghu [2]
[1] *Research-Scholar, Bharath University, Chennai-600126,Tamil Nadu, India*
[2]*Co-Author & Supervisor: Dr.B.Raghu, Principal, SVS Groups of Institutions, Warangal-506 015, Telangana, India*

Abstract:

Data is collected from point of sale transactions, inventory status and pricing, competitive intelligence, social media, weather, and customers (scrubbed of personal identification) and then pulled together on the Hadoop Platform, allowing for a centralized analysis of correlations and patterns that are relevant to improving business.
In-store and online purchases, Twitter trends, local sports events, and weather buying patterns are analyzed by big data algorithms to build innovative applications that personalize customer experience while increasing the efficiency of logistics. Point of sale transactions are analyzed to provide product recommendations or discounts, based on which products were bought together or before another product. Predictive analytics is used to know what products sell more on particular days in certain kinds of stores, to reduce overstock and to remain properly stocked on the most in-demand products, helping to optimize the supply chain.
Key Words – *Big Data Analytics, Hadoop,Retail analytics.*

I. Hadoop:

The Hadoop is distributed file system which is highly effective for process large amounts data using effective distributed techniques which takes use of the commodity instead of using the super computers to process the large amounts of data. The Hadoop is now the *de facto* standard for Big-Data Analysis.

Hadoop is the open source software which is developed and maintained by theApache software foundation.The creation of the Hadoop is from the Google's white paper on the Map Reduce architecture with the distributed computing technology embedded to it. This made the founders of the Hadoop to invent it and now it's the revolution with the three important "*V*" the volume, variety and velocity. Now-a-days there are of options and more functionality in the Hadoop to use it with the various set of tools to take the advantage of the Hadoop File System.

II. HDFS (Hadoop File System)

HDFS is the distributed file system for the Hadoop ecosystem it takes the take and distributes across it nodes for distributed computing access, the Hadoop is an analytical tool but not the dynamic storage database we load the large amount it, but not advisable to the small volumes of data into which makes it less efficient since relation database system quiet effective in the small amounts of data under Gigabytes.

HDFS uses block replication to reliably store very large files across multiple machines with two distinct pieces of software: The NameNode server, which manages the file system namespace and client access, and the DataNodes, responsible for serving read and write requests, as well as block creation, deletion, and replication. A basic understanding of the replication pattern can be helpful for developers and cluster administrators since, while it generally works well, imbalances in the distribution of data can impact cluster performance and require tuning. HDFS stores each file as a sequence of blocks, each identically sized except for the final one.

By default, blocks are replicated three times, but both the size of the blocks and the number of replicas can be configured on a per-file basis. Files are write-once and have only a single writer at any time in order to enable high throughput data access and simplify data coherency issues.

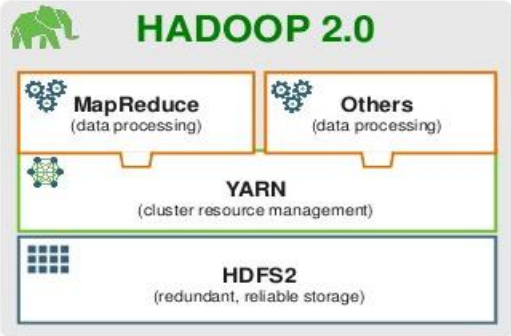
The NameNode makes all decisions about block replication based on the heartbeat and block reports it receives from each DataNode in the cluster. The heartbeat signals that the DataNode is healthy, and the block report provides a list of all blocks on the

DataNode. When a new block is created, HDFS places the first replica on the node where the writer is running. A second replica is written on a randomly chosen node in any rack except the rack where the first replica was written.

Then the third replica is placed on a randomly chosen machine in this second rack. If more than the default of three replicas is specified in the configuration, the remaining replicas are placed randomly, with the restriction that no more than one replica is placed on any one node and no more than two replicas are placed on the same rack.

Important commands in HDFS:

Pushing Files into Hadoop file system



$ hadoop fs –get filename

$ hadoop fs –put filename

To retrieve it back from the HDFS

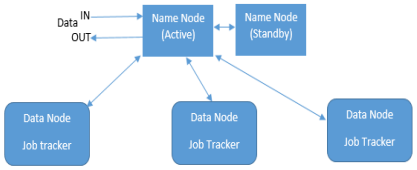
To list the files in the HDFS

$ hadoop fs –ls

Fig.1 Hadoop 2.0 Architecture

Imaginethe file size is 100 MB then its broken into the 64MB blocks and then replicated at least thrice onto different data nodes in a cluster and all the information is stored into, so the 100 MB would be 64 MB plus reaming 46 MB block written into the different data node and then replicated 3 times the same blocks and then registered into the name node.

Fig.2HDFC System Diagram



I. Map Reduce for Retail Analytics

Map Reduce is the programing model to extract the useful inform and necessary information from the datasets present in the HDFS, first the data is mapped to required fields and removing the unwanted information and then its sorted to align then this data is feed into the reducer which then reduces this sorted redundant information or unique set of values to monitor the outcome of it.

Map Reduce is the Googles programming model to extract the necessary and useful data out of the large files which are stored in the HDFS, it contacts the YARN (Yet another resource manager) to get the data out of the HDFS in Hadoop 1 the MapReduce is directly coupled to the HDFS so there the usage of the resources where constrained and we were restricted to use the Java language, with the YARN we possible integrate defined or custom data processing software's to have more useful data.Let's take a look at the sample dataset for how the map reduce is used to achieve meaningful analysis.The below listed is the very less amount of data from the sample, from this the San Jose is repeated twice since they have two orders from San Jose. The sample set consists of the date, time, area, type of purchase, and amount and payment type in the same order, from the list let's get the total purchases by the area name to understand the purchases by the area. Implementing Map and reduce to this, we want to know to values area name and the amount. Let's get those form the sample date the name and price are the fields for mapper, after that output data is sorted and then passed into the reducer which then joins all the amounts by the area name. this is the functionality implemented for this sample. This is the output for the MapReduce when implemented for the less amount of the data set its is faster but for the large

amount it takes days to process it, for that we need distributed system which has the distributed jobs running capacity.

II. **TABLE I: Sample data: Part of the voluminous data**

Date	Time	Area Name	Type of Purchase	Price	Purchase Type
2012-01-01	09:00	San Jose	Men's Clothing	214.05	Amex
2012-01-01	09:00	Fort Worth	Women's Clothing	153.57	Visa
2012-01-01	09:00	San Diego	Music	66.08	Cash
2012-01-01	09:00	Pittsburgh	Pet Supplies	493.51	Discover
2012-01-01	09:00	Omaha	Children's Clothing	235.63	MasterCard
2012-01-01	09:00	Stockton	Men's Clothing	247.18	MasterCard
2012-01-01	09:00	Austin	Cameras	379.6	Visa
2012-01-01	09:00	New York	Consumer Electronics	296.8	Cash
2012-01-01	09:00	Corpus Christi	Toys	25.38	Discover
2012-01-01	09:00	Fort Worth	Toys	213.88	Visa
2012-01-01	09:00	Las Vegas	Video Games	53.26	Visa
2012-01-01	09:00	San Jose	Women's Clothing	215.82	Cash

III. Component of Hadoop :

Components of Hadoop: HBase: It is open source, distributed and Non-relational database system implemented in Java. It runs above the layer of HDFS. It can serve the input and output for the Map Reduce in well-mannered structure.

Oozie: Oozie is a web-application that runs in a java servlet. Oozie use the database to gather the information of Workflow which is a collection of actions. It manages the Hadoop jobs in a mannered way.

Sqoop: Sqoop is a command-line interface application that provides platform which is used for converting data from relational databases and Hadoop or vice versa. Avro: It is a system that provides functionality of data serialization and service of data exchange. It is basically used in Apache Hadoop.

These services can be used together as well as independently according the data records. Chukwa: Chukwa is a framework that is used for data collection and analysis to process and analyze the massive amount of logs. It is built on the upper layer of the HDFS and Map Reduce framework.

Pig: Pig is high-level platform where the MapReduce framework is created which is used with Hadoop platform. It is a high level data processing system where the data records are analyzed that occurs in high level language.

Zookeeper: It is a centralization based service that provides distributed synchronization and provides group services along with maintenance of the configuration information and records.

Hive: It is application developed for data warehouse that provides the SQL interface as well as relational model. Hive infrastructure is built on the top layer of Hadoop that help in providing conclusion, and analysis for respective queries.

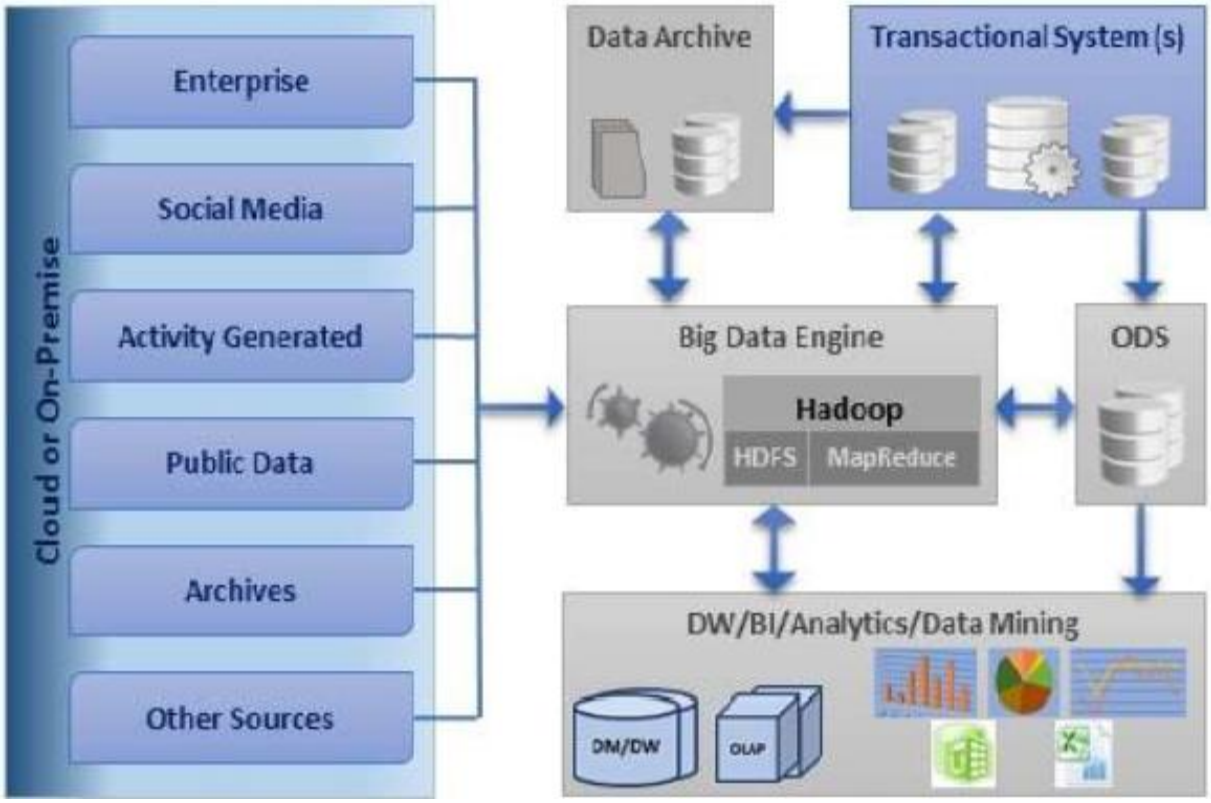


Fig.3 Modern Big Data System

IV. Steps to run the Jobs in Hadoop:

1. Implementing the Jobs of MapReduce in HDFS:
   Pushing data to be analyzed to the HDFS
   $ hadoop –push data.file

2. Run the Map Reduce jobs into the HDFS system by the
$ hadoop jar "path of the hadoop job runner" –mapper mapper.py –reducer reducer.py –file mapper.py –file reducer.py –input "input file name" –c output "output dir name"

We have set the above command to alias of "*hs*"
$ hs mapper.py reducer.py "inputfilename" "outputdirname"

After running it for sample dataset the observed output is as below:

[training@localhost code]$ hs mapper.py reducer.py testfile joboutput
packageJobJar: [mapper.py, reducer.py, /tmp/hadoop-training/hadoop-unjar8616757019681653336/] [] /tmp/streamjob538467600025788142.jar tmpDir=null
17/12/21 06:20:36 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
17/12/21 06:20:36 WARN snappy.LoadSnappy: Snappy native library is available
17/12/21 06:20:36 INFO snappy.LoadSnappy: Snappy native library loaded
17/12/21 06:20:36 INFO mapred.FileInputFormat: Total input paths to process : 1
17/12/21 06:20:37 INFO streaming.StreamJob: getLocalDirs(): [/var/lib/hadoop-hdfs/cache/training/mapred/local]
17/12/21 06:20:37 INFO streaming.StreamJob: Running job: job_201712181131_0002
17/12/21 06:20:37 INFO streaming.StreamJob: To kill this job, run:
17/12/21 06:20:37 INFO streaming.StreamJob: UNDEF/bin/hadoop job -Dmapred.job.tracker=0.0.0.0:8021 -kill job_201712181131_0002
17/12/21 06:20:37 INFO streaming.StreamJob: Tracking URL:

http://0.0.0.0:50030/jobdetails.jsp?jobid=job_201712181131_0002
17/12/21 06:20:38 INFO streaming.StreamJob: map 0% reduce 0%
17/12/21 06:20:50 INFO streaming.StreamJob: map 100% reduce 0%
17/12/21 06:21:01 INFO streaming.StreamJob: map 100% reduce 100%
17/12/21 06:21:06 INFO streaming.StreamJob: Job complete: job_201712181131_0002
17/12/21 06:21:06 INFO streaming.StreamJob: Output: joboutput

V. Web View for the Hadoop

The web view gives us the detailed information about the Hadoop process running and the time it took it to complete the process, and detailed information regarding it in the same page to keep track of the jobs and to debug if any issue occurs.

Hadoop job_201712181131_0001 on 0

User: training
Job Name: streamjob3134222797528906952.jar
Job File: hdfs://0.0.0.0:8020/var/lib/hadoop-hdfs/cache/mapred/mapred/staging/training/.staging/job_201712181131_0001/job.
Submit Host: localhost.localdomain
Submit Host Address: 127.0.0.1
Job-ACLs: All users are allowed
Job Setup: Successful
Status: Succeeded
Started at: Thu Dec 21 06:09:16 EST 2017
Finished at: Thu Dec 21 06:09:34 EST 2017
Finished in: 17sec
Job Cleanup: Successful

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	1	0	0	1	0	0 / 0
reduce	100.00%	1	0	0	1	0	0 / 0

Fig.4 Web view for the Hadoopprocessing

After opening the joboutput as specified in the job execution command,
[training@localhost code]$ hadoop fs -ls joboutput
Found 3 items
-rw-r--r-- 1 training supergroup 0 2017-12-21 06:21 joboutput/_SUCCESS
drwxr-xr-x - training supergroup 0 2017-12-21 06:20 joboutput/_logs
-rw-r--r-- 1 training supergroup 1081 2017-12-21 06:21 joboutput/part-00000

The part-00000 consists the result processed by the reducer at last. It shows area vice total sales.

[training@localhost code]$ hadoop fs -cat joboutput/part-00000

Albuquerque	440.7
Anchorage	1086.22
Atlanta	254.62
Aurora	117.81
Austin	1304.78
Boise	481.09
Durham	557.76
Fort Wayne	370.55
Fort Worth	367.45
Fremont	465.66
Fresno	663.47
Garland	134.33
Greensboro	443.72
Honolulu	345.18
Houston	309.16
Indianapolis	135.96
Kansas City	302.69
Las Vegas	146.65
Lexington	359.29
Lincoln	136.9
Lubbock	452.07
Madison	16.78
Memphis	9.47
Milwaukee	418.91
Minneapolis	182.05
Nashville	297.43
New Orleans	399.39
New York	468.91
Newark	39.75
Norfolk	669.25
Oakland	155.33
Omaha	879.06
Philadelphia	683.79
Boston	418.94
Buffalo	483.82
Chandler	758.17
Charlotte	440.11
Chicago	47.23
Cincinnati	299.55
Phoenix	371.21
Pittsburgh	1015.76
Portland	108.69
Reno	168.71
Riverside	1021.61
Saint Paul	33.13
San Antonio	125.35
San Bernardino	270.58
San Diego	66.08
San Francisco	260.65
SanJose	429.87

Santa Ana	402.92
Spokane	448.
ledo	59.08
Tucson	367.76
Tulsa	225.13
Virginia Beach	647.67
Washington	392.81
Wichita	460.38
Jacksonville	359.65

VI. Conclusion:

Purposes of Implementing the job on the above data set is to get the understanding of what's is amount spend by the people for the particular region of the stores , consider this example in san Jose we have 5 purchases so get the total amount we sum up all the five and result is the region name and the total amount spend from that region, the result very good insights about the expenditure of the customers based on the expenditure, based on this we can have improve and get more goods for that store and making it more profitable to improve the customer satisfaction for the every store, if the expenditure is less then we have look at what the customers are looking for, like are products available for the customer's? if yes what's stooping them from buying, are prices too high, are the all the payment types supported for that regions, etc. , by the above data implications, the good insights for selling more products with high customer satisfaction, this will lead to success for the business

References:

[1] Berkovich, S., Liao, D.: On Clusterization of big data Streams. In: 3rd International Conference

[2] on Computing for Geospatial Research and Applications, article no. 26. ACM Press,New York (2012).

[3] Beyer, M.A., Laney, D.: The Importance of 'Big Data': A Definition. Gartner (2012)

[4] Madden, S.: From Databases to big data. IEEE Internet Computing 16(3), 4–6 (2012)Chen, H., Chiang, R., & Storey, V. (2012). Business Intelligence and Analytics: from Big Data to BigImpact. MIS Quarterly, 36 (4), 1165-1188.

[5] Cheng, Y., Qin, C., & Rusu, F. (2012). GLADE: Big Data Analytics Made Easy. SIGMOD (pp. 697-700). AR: ACM.Cuzzocrea, a., Song.

[6] K. Ashton, That ''Internet of Things'' thing, RFiD Journal (2009).

[7] H. Sundmaeker, P. Guillemin, P. Friess, S. Woelfflé, Vision and challenges for realising the Internet of Things, Cluster of European Research Projects on the Internet of Things—CERP IoT, 2010. [3] J. Buckley (Ed.)

[8] The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems, Auerbach Publications, New York, 2006.