OPEN ACCESS

International Journal Of Modern Engineering Research (IJMER)

Influence of Hadoop in Big Data Analysis and Its Aspects

Sumaiya Nazneen[1], Sara Siraj[2], Tabassum Sultana[3], Nabeelah Azam[4], Tayyiba Ambareen[5], Ethemaad Uddin Ahmed[6], Mohammed Salman Irshad[7]

[1,2,3,4,5,6,7] Student, Department of Information Technology, Muffakham Jah college of Engineering and Technology, Mount Pleasant, 8-2-249, Road Number 3, Banjara Hills, Hyderabad, Telangana, INDIA.

ABSTRACT: *This paper is an effort to present the basic understanding of BIG DATA and HADOOP and its usefulness to an organization from the performance perspective. Along-with the introduction of BIG DATA, the important parameters and attributes that make this emerging concept attractive to organizations has been highlighted. The paper also evaluates the difference in the challenges faced by a small organization as compared to a medium or large scale operation and therefore the differences in their approach and treatment of BIG DATA. As Hadoop is a Substantial scale, open source programming system committed to adaptable, disseminated, information concentrated processing. A number of application examples of implementation of BIG DATA across industries varying in strategy, product and processes have been presented. This paper also deals with the technology aspects of BIG DATA for its implementation in organizations. Since HADOOP has emerged as a popular tool for BIG DATA implementation. Map reduce is a programming structure for effectively composing requisitions which prepare boundless measures of information (multi-terabyte information sets) in- parallel on extensive bunches of merchandise fittings in a dependable, shortcoming tolerant way. A Map reduce skeleton comprises of two parts. They are "mapper" and "reducer" which have been examined in this paper. The paper deals with the overall architecture of HADOOP along with the details of its various components in Big Data.*

Keywords *– Big data, Hadoop, Analytic databases Framework, HDFS, Map Reduce.*

I. Introduction

Companies across the world have been using data since a long time to help them take better decisions in order to enhance their performances. It is the first decade of the 21st century that actually showcased a rapid shift in the availability of data and it's applicability for improving the overall effectiveness of the organization. This change that was to revolutionize the use of data.

Hadoop was designed especially for the analysis of large data sets to build scalable, distributed applications. To manage sizably voluminous data, Hadoop implements the paradigm called MapReduce defined by Google according to which the applications are divided into minute chunks of software, each of which can be run on a distinct node of all those who make up the system. Companies like Amazon, Cloudera, IBM, Intel, Twitter, Facebook and others are formulate their immensely enormous data message and providing insight into where the market is headed utilizing Apache Hadoop technology.

What is Big Data: Big data is the availability of a large amount of data which becomes difficult to store, process and mine using a traditional database primarily because of the data available is large, complex, unstructured and rapidly changing. This is probably one of the important reasons why the concept of big data was first embraced by online firms like Google, eBay, Facebook, LinkedIn etc.

Big Data in small v/s big companies: There is a specific reason as to why big data was first appreciated by the online firms and start-ups as mentioned above. These companies were built around the concept of using rapidly changing data and did not probably face the challenge of integrating the new and unstructured data with the already available ones. If we look at the challenges regarding big data being faced by the online firms and the start-ups we can highlight the following:

i. Volume: The largeness of the data available made it a challenge as it was neither possible nor efficient to handle such a large volume of data using traditional databases.

ii. Variety: As compared to the earlier versions, where data was available in one or two forms (possibly text and tables), and the current versions would mean data being available additionally in the form of pictures, videos, tweets etc.
iii. Velocity: Increasing use of the online space meant that the data that was available was rapidly changing and therefore had to be made available and used at the right time to be effective.

The MapReduce is a programming model designed for processing immensely colossal volumes of data in parallel by dividing the work into a set of independent tasks. MapReduce programs are invited in a particular style influenced by functional programming constructs, categorically idioms for processing lists of data. A MapReduce program is possessed of a "Map()" procedure that executes filtering and sorting (for example sorting people by first name into queues, one queue for each one name)and a "Reduce()" procedure that implements a synopsis operation. The "MapReduce Framework" (likewise called "infrastructure" or "framework") organizes by assembling the distributed servers, running the various tasks in parallel, controlling all communications and data transfers between the numerous parts of the framework, and accommodating for redundancy and fault tolerance. A prevalent open- source requisition is Apache Hadoop.

II. Big Data

2.1 The Challenges for Big Firms:

Big data may be new for startups and for online firms, but many large firms view it as something they have been wrestling with for a while. Some managers appreciate the innovative nature of big data, but more find it "business as usual" or part of a continuing evolution toward more data. They have been adding new forms of data to their systems and models for many years, and don't see anything revolutionary about big data. Put another way, many were pursuing big data before big data was big.

When these managers in large firms are impressed by big data, it's not the "bigness" that impresses them. Instead it's one of three other aspects of big data: the lack of structure, the opportunities presented, and low cost of the technologies involved. This is consistent with the results from a survey of more than fifty large companies by New Vantage Partners in2012. It found, according to the survey summary:

2.2 It's About Variety, Not Volume:

The survey indicates companies are focused on the variety of data, not its volume, both today and in three years. The most important goal and potential reward of Big Data initiatives is the ability to analyze diverse data sources.
Application areas and implementation examples:
1. Big Data for cost reduction: Some organizations that are pursuing big data believe strongly that for the storage of large data that is structured, Big data technologies like Hadoop clusters are very cost effective solutions that can be efficiently utilized for cost reduction. One company's cost comparison, for example, estimated that the cost of storing one terabyte for a year was $37,000 for a traditional relational database, $5,000 for a database appliance, and only $2,000 for a Hadoop cluster.1 Of course, these figures are not directly comparable, in that the more traditional technologies may be somewhat more reliable and easily managed. Data security approaches, for example, are not yet fully developed in the Hadoop cluster environment.

2.3 Big Data at Ups:

UPS is no stranger to big data, having begun to capture and track a variety of package movements and transactions as early as the 1980s. The company now tracks data on 16.3 million packages per day for 8.8 million customers, with an average of 39.5 million tracking requests from customers per day. The company stores over 16 peta-bytes of data.

Much of its recently acquired big data, however, comes from telematic sensors in over 46,000 vehicles. The data on UPS package cars (trucks), for example, includes their speed, direction, braking, and drive train performance. The data is not only used to monitor daily performance, but to drive a major redesign of UPS drivers' route structures. This initiative, called ORION (On-Road Integrated Optimization and Navigation), is arguably the world's largest operations research project. It also relies heavily on online map data, and will eventually reconfigure a driver's pickups and drop-offs in real time. The project has already led to savings in 2011 of more than 8.4 million gallons of fuel by cutting 85 million miles off of daily routes. UPS estimates that saving only one daily mile driven per driver saves the company $30 million, so the overall dollar savings are substantial. The company is also attempting to use data and analytics to optimize the efficiency of its 2000 aircraft flights per day.

2.4 Big Data for Time Reduction:

The second common objective of big data technologies and solutions is time reduction. Macy's merchandise pricing optimization application provides a classic example of reducing the cycle time for complex and large-scale analytical calculations from hours or even days to minutes or seconds. The department store chain has been able to reduce the time to optimize pricing of its 73 million items for sale from over 27 hours to just over 1 hour. Described by some as "big data analytics," this capability set obviously makes it possible for Macy's to re-price items much more frequently to adapt to changing conditions in the retail marketplace. This big data analytics application takes data out of a Hadoop cluster and puts it into other parallel computing and in-memory software architectures. Macy's also says it achieved 70% hardware cost reductions. Kerem Tomak, VP of Analytics at Macys.com, is using similar approaches to time reduction for marketing offers to Macy's customers. He notes that the company can run a lot more models with this time savings.

2.5 Big Data for Improving Process Efficiency:

Big data can be used for improving the process efficiency also. An excellent use of big data in this regard is cricket especially with the advent of the Indian Premier League (IPL). Not only are matches analyzed using the data available in order to formulate future strategies but even minute details like the performance of a bowler against a particular batsman and that too on a particular ground under certain conditions are being made available for the stakeholders to improve their efficiency.

For example, how will a batsman like Glenn Maxwell perform against a bowler like Sunil Narine at Eden Gardens or how different will it be at Mohali in Chandigarh is available to be used? Not only this but also data like how many balls has a particular batsman faced against a particular bowler the number of dot balls and the number of runs scored. Another example in this regard is the use of Big data to predict the probability (at any time during a match) of a team winning or losing in a match based on the extrapolation of the results in similar match situations.

III. Hadoop As An Open Source Tool For Big Data Analytics

Hadoop is a distributed software solution. It is a scalable fault tolerant distributed system for data storage and processing. There are two main components in Hadoop:

(i) HDFS (which is a storage)
(ii)Map Reduce (which is retrieval and processing): So HDFS is high bandwidth cluster storage and it of great use what is happening here is **(Fig. 1)**

We put a pent byte file on our Hadoop cluster, HDFS is going to breakup into blocks and then distributed it to across all of the nodes of our cluster and on top of that we have a fault tolerant concept what is done here is HDFS is configure Replication Factor (which is by default set to 3). What does this mean we put our file on hadoop it is going to make sure that it has 3 copy of every block that make up that file spread across all the node in our cluster .It is very useful and important because if we lose a node it has a self-feel what data was there on node and I am going to replicate that blocks that were on that node. The question arise how it does that for this It has a name node and a data node generally one name node per cluster but essentially name node is a meta data server it just hold in memory the location of every block and every node and even if you have multiple rack setup it will know where block exist and what racks across the cluster inside in your network that's the secret behind HDFS and we get data.

Map Reduce: Now how we get data is through Map Reduce as name implies it is a two-step process. There is a Mapper and Reducer programmers will write the mapper function which will go out and tell the cluster what data point we want to retrieve. The Reducer will then take all of the data and aggregate.

Hadoop is a batch processing here we are working on all the data on cluster, so we can say that Map Reduce is working on all of data inside our clusters. There is a myth that one need to be understand java to get completely out of clusters, in fact the engineers of Facebook built a subproject called HIVE which is sql interpreter. Facebook wants a lot of people to write adhoc jobs against their cluster and they are not forcing people to learn java that is why team of Facebook has built HIVE, now anybody who is familiar with sql can pull out data from cluster.

Pig is another one built by yahoo, it's a high level data flow language to pull data out of clusters and now Pig and hive are under the Hadoop Map Reduce job submitted to cluster. This the beauty of open source framework people can built, add and community keeps on growing in Hadoop more technologies and projects are added into Hadoop ecosystem.

3.1 MAPREDUCE FRAMEWORK

The MapReduce framework consists of two steps namely Map step and reduce step. Master node takes large problem input and slices it into smaller sub problems and distributes these to worker nodes. Worker node may do this again and leads to a multi-level tree structure .Worker processes smaller problem and hands back to master. In Reduce step Master node takes the answers to the sub problems and combines them in a predefined way to get the output/answer to original problem. The MapReduce framework is fault-tolerant because each node in the cluster is expected to report back periodically with completed work and status updates. If a node remains silent for longer than the expected interval, a master node makes note and re-assigns the work to other nodes.

3.2 WORKFLOW IN MAPREDUCE

The key to how MapReduce works is to take input as, conceptually, a list of records. The records are split among the different computers in the cluster by Map. The result of the Map computation is a list of key/value pairs. Reducer then takes each set of values that has the same key and combines them into a single value. So Map takes a set of data chunks and produces key/value pairs and Reduce merges things, so that instead of a set of key/value pair sets, you get one result. You can't tell whether the job was split into 100 pieces or 2 pieces. MapReduce isn't intended to replace relational databases. It’s intended is to provide a light weight way of programming things so that they can run fast by running in parallel on a lot of machines.

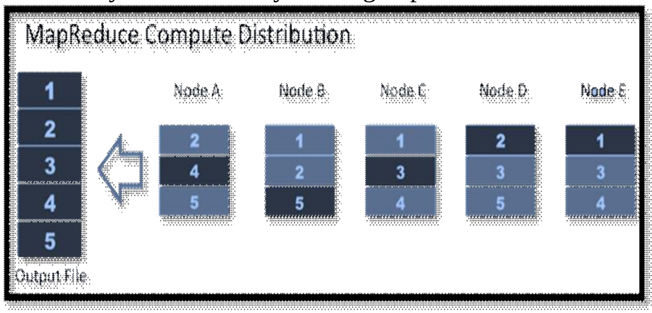

Fig. 1 Computation of MapReduce

MapReduce is important because it allows ordinary developers to use MapReduce library routines to create parallel programs without having to worry about programming for intra-cluster communication, task monitoring or failure handling. It is useful for tasks such as data mining, log file analysis, financial analysis and scientific simulations. Several implementations of MapReduce are available in a variety of programming languages, including Java, C++, Python, Perl, Ruby, and C. Typical Hadoop cluster integrates MapReduce and HFDS with master / slave architecture which consists of a Master node and multiple slave nodes. Master node contains Job tracker node (MapReduce layer), Task tracker node (MapReduce layer), Name node (HFDS layer),Data node (HFDS layer). Multiple slave nodes are Task tracker node (MapReduce layer) and Data node (HFDS layer). MapReduce layer has job and task tracker nodes while HFDS layer has name and data nodes.

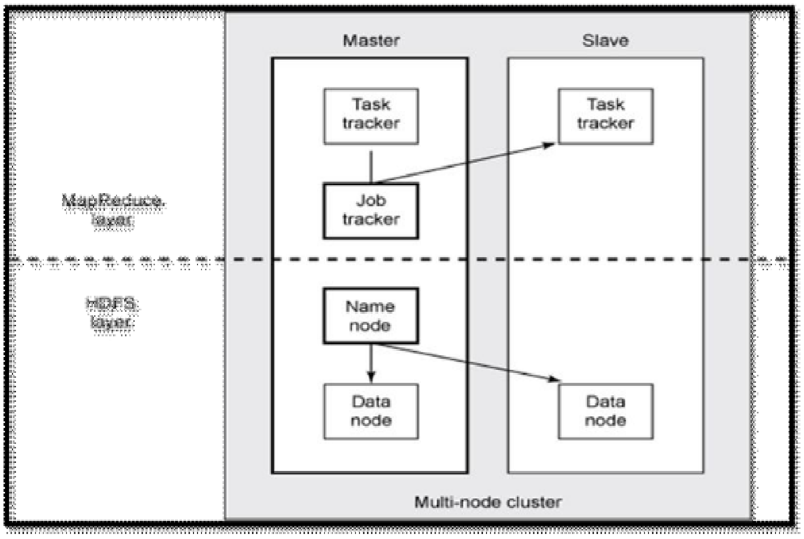

Fig. 2 Layers in MapReduce

Although the Hadoop framework is implemented in Java TM, MapReduce applications need not be written in Java. Hadoop Streaming is a utility which allows users to create and run jobs with any executable (e.g. Shell utilities) as the mapper and/or the reducer. Hadoop Pipes is a SWIG-compatible C++ API to implement MapReduce applications (non JNITM based).

3.4 DATA ORGANIZATION:

In many organizations, Hadoop and other MapReduce solutions are only the examples in the larger data analysis platform. Data will typically have to be translated in order to interface perfectly with the other organizations. Similarly, the data might have to be transmuted from its original state to a new state to clear analysis in MapReduce easier.

3.4.1 NATURE OF DATA:

MapReduce systems such as Hadoop aren’t being utilized exactly for text analysis anymore. An increasing number of users are deploying MapReduce jobs that analyze data once thought to be excessively difficult for the paradigm. One of the most obvious trends in the nature of data is the boost of image, audio, and video analysis. This kind of data is a serious prospect for a distributed system using MapReduce because these files are typically very big. Retailers want to examine their security video to detect what stores are most engaged. Medical imaging analysis is becoming harder with the astronomical resolutions of the image. Videos have colored pixels that change over time, laid out a flat grid. The data are analyzed in order by challenging to take a look at 10-pixel by 10-pixel by 5-second section of video and audio as a “record.” As multidimensional data increases in popularity, there are more patterns showing how to logically separate the data into records and input splits properly. For example, SciDB, an open- source analytical database, is specifically built to deal with multi-dimensional data. MapReduce is traditionally a batch analytics system, but streaming analytics feels like a natural onward motion. In many production MapReduce systems, data are always streaming in and then gets treated in batch on an interval. For instance, data from web server logs are streaming in, but the MapReduce job is only done every hour. This is inconvenient for a few reasons. First, processing an hour’s worth of data at once can strain resources. Novel systems that deal with streaming data in Hadoop have cropped up, most notably the commercial product like HStreaming and the open-source Storm platform, recently released by Twitter.

3.5 SOME ESSENTIAL HADOOP PROJECTS:

Data Access: The reason why we need more way to access data inside Hadoop is because not everyone is low level, Java, C or C++ programmer that can write Map Reduce Jobs to get the data and even if you are something what we do in SQL like grouping, aggregating, joining which a challenging job for anybody even if you are a professional. So we got some data access library. Pig is one among them. Pig is just a high level flow scripting language. It is really very easy to learn and hang-up. It does not have lot of keywords in it. It is getting a data, loading a data, filtering up, transforming the data and either returning and storing those results. There are 2 core components of PIG:
Pig Latin: This is a programming language.
Pig Runtime: which competes pig Latin and converts it into map reduce job to submit to cluster.
Hive: is another Data access project extremely popular like pig. Hive is a way to project structures on to data inside a cluster so it is really a database. Data-warehouse built on top of Hadoop and it contains a query language Hive QL like SQL to hive query language and it is extremely similar to SQL. Hive is similar thing like pig. It converts these queries into map reduce jobs that gets submitted to cluster.

Moving Down to Technology stack we have:
Data storage: Remember out of the box is a batch processing system. We put the data into HDFS system; once we read in many time or what if we needed to get specific data; What if we want to do real time processing system on top of Hadoop data and that’s why we have some of the column oriented database known as Hbase these are just Appache projects but there a buzz term for this NoSQL. Not once SQL that wants it stands for does not mean you can’t use sql like language to get data out. What it means the underlying structure of the database are not strict like they are in relational world very loose, very flexible which makes them very scalable : that’s what we need in the world of Big data and Hadoop, In fact those are lot of NoSQL database platform out here. One of the most popular is Mangodb.

Mangodb is extremely very popular, especially among programmers because it is really very easy to work with. It is document style storage model which means programmers can take data models and clone. We call objects in those applications and serialize them right into Mangodb and with the same ease can bring them back into application .

Hbase was based on Google Big table, which is a way we can create table which contains millions of rows and we can put indexes on them and can do serious data analysis and Hbase is data analysis we put indexing on them and go to high performance which seeks to find data which we are looking for and nice thing about Hbase is pig and hive natively agree with Hbase. So you write pig and hive queries against data sitting.

Cassandra is designed to handle large amount of data across many commodity servers, providing high availability with no single point of failure. Cassandra offers robust support for clusters spanning multiple data centers. It has its root in Amazon with more data storage table and it is designed for real time interactive transaction processing on top of our hadoop cluster. So both of them solve different problems but they both require seeking against our Hadoop data.

We also have random collection of projects that span of different categories some of these solve specific business problems and some of them are little more generic like.

Lucent: Lucent is there for full text searching an API loaded with algorithms to do things like standard full text searching, wild card searching, phrase searching, range searching kind of stuff.

Hama: Hama is there for BSP (Book Synchronous Processing). Here we need to work with large amount of scientific data.

Hcatalog: It is known as metadata table and storage management system. What does it mean it's a way to create a shared schema, it's a way for tools like Pig, Hive for interoperable also to have consistent view of data across those tools.

Avro: These are data serialization technology which is a way in which we can take data from application, package up into a format that we can either store in our disk or send across the wires so that another application can unpack it and desterilize into a format that they can understand. Avro is more generic

Thrift is more specific for creating flexible schemas that work with hadoop data. Its specific because it is meant for cross- language compatibility, so we build an application with hadoop data in java and if we want to use same object inside an application that you built on Ruby, Python or C++.

Crunch: Crunch is there for writing and testing and running map reduce pipeline. It essentially gives you full control to overall four phrases, which is going to be: 1. Map, 2. Reduce, 3. Shuffle, and 4. Combine. It is there to help in joining and aggregation that is very hard to do in low level map reduce, so Crunch is there to make you little more easier inside map reduce pipeline.

Data Intelligence: We also have data intelligence in the form of Drill and Mahout.
Drill: Drill is actually an incubator project and is designed to do interactive analysis on nested data.
Mahout: Mahout is a machine learning library that concurs the three Cs:

1. Recommendation (Collaborative Filtering)
2. Clustering (which is a way to group related text and documents)
3. Classification (which is a way to categorize related text and documents).

So Amazon uses all this stuff to a further recommendation like music sites uses to recommend songs you listen and also to do predictive analysis.

Sqoop: On the left side of Figure 2. We have Sqoop. It is a widely popular project because it is easy to integrate hadoop with relational systems. For instance, we have result of map reduce. Rather than taking these results and putting them on HDFS, and require Pig and Hive query, we can send those results to relational world so that a data professional can do their own analysis and become the part of process. So, Sqoop is popular for pushing hadoop data into relational world, but it is also popular for pushing data from relational world into hadoop, like archiving.

Flume and Chukwa: are real time log processing tools so that we can set up our frame-work where our applications, operating systems, services like web-services that generate mountains of log data. It's a way we can push real time data information right into hadoop and we can also do real time analysis.

Over the right hand side of **Figure 3.** We have a tool for managing, monitoring and orchestrating all the things that go in our cluster:

Oozie: It is a work flow library that allows us to play, and to connect lot of those essential projects for instances, Pig, Hive and Sqoop.
Zoo Keeper: It is a distributed service coordinate. So it's a way in which we keep our all services running across our entire cluster synchronous. So, it handles all synchronization and serialization. It also gives centralized management for these services.
Ambari: It allows you to provision a cluster which means that we can install services, so that we can pick Pig, Hive, Sqoop, Hbase and install it. It will go across all the nodes in cluster and also we can manage our services from one centralized location like starting up, stopping, reconfiguring and we can also monitor a lot of these projects from Ambari.

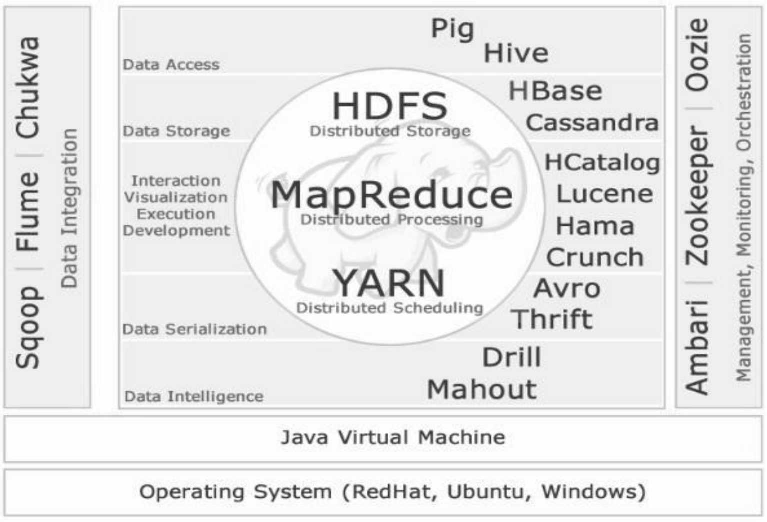

Figure 3. The image shows the hadoop technology stack. The hadoop core/common which consists of HDFS (Distributed Storage) which is a programmable interface to access stored data in cluster.

3.6 YARN (Yet another resource negotiation)

It's a Map Reduce version 2. This is future stuff. This is stuff which is currently alpha and yet to come. It is rewrite of Map
Reduce 1.

IV. Inputs And Outputs

The MapReduce framework operates exclusively on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs[15] and produces a output of the job as set of <key, value> pairs conceivably of distinct types. The key and value classes have to be serializable by the framework and hence need to implement the writable interface. Additionally, the key classes have to implement the Writable Comparable interface to facilitate sorting by the framework.

4.1 I/O TYPES OF A MAPREDUCE JOB:

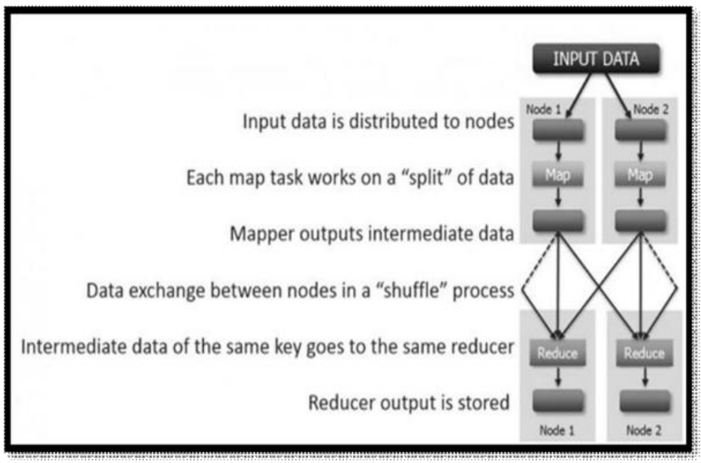
INPUT DATA
Node 1
Node 2
Input data is distributed to nodes
Each map task works on a "split" of data
Map
Map
Mapper outputs intermediate data
Data exchange between nodes in a "shuffle" process
Intermediate data of the same key goes to the same reducer
Reduce
Reduce
Reducer output is stored
Node 1
Node 2

Fig.4 Distribution of Input data

A simple MapReduce program can be written to determine how many times different words appear in a set of files for example if we the files like file1, file2, and file 3.

Input:

File1: Deer, Bear, River File2: Car, Car, River File3: Deer, Car, Bear

We can a write a program in map reduce by using three operations like map, combine, reduce to compute the output.

The first step is Map Step:

First Map	Second Map	Third Map
<Deer,1>	<Car,1>	<Deer,1>
<Bear,1>	<Car,1>	<Car,1>
<River,1>	<River,1>	<Bear,1>

The secondary step is Combine Step:

<Bear,1> <Car,1> <Deer,1> <River,1>
<Beer,1> <Car,1> <Deer,1> <River,1>
<Car,1>

The final step is Reduce Step:

<Beer,2> <Car,3> <Deer,2> <River,2>

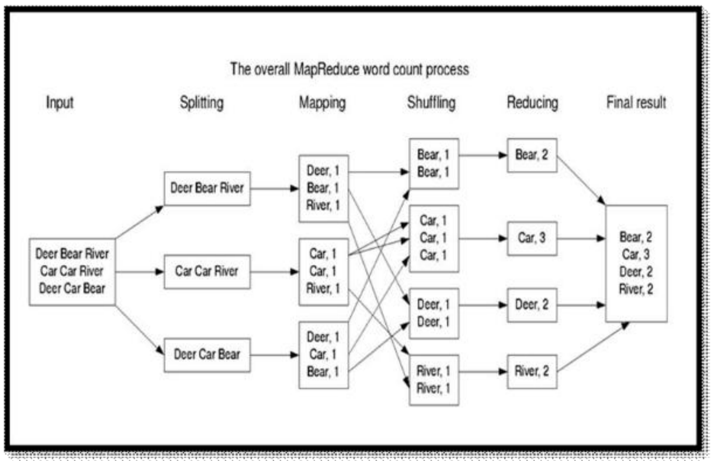

Fig.5 Example showing MapReduce Job

4.2 TASK EXECUTION AND ENVIRONMENT

Task Tracker executes Mapper/Reducer task as a child process in a separate JVM (Java Virtual Machine). The Child task inherits the environment of the parent Task Tracker. A User can specify environmental variables controlling memory, parallel computation settings, segment size. A requirement of applications using MapReduce specifies the Job configuration, input/output locations. It supply map and reduce functions via implementations of appropriate interfaces and/or abstract classes.

4.3 SCHEDULING

Usually, Hadoop uses FIFO to schedule jobs. The scheduler option depends on capacity and fair. Jobs are submitted to the queue according to their priority. Queues are allocated according to the resources capacity. Free resources are allocated to queues away from their total capacity.

V. Conclusion

From the topics discussed in detail we get to know the concept of Hadoop, Map Reduce and other variables in Big Data Analysis. Doug Cutting, Cloudera's chief architect, helped create Apache Hadoop out of necessity as data from the web exploded and grew far beyond the ability of traditional systems to handle it. Hadoop was initially inspired by papers published by Google outlining its approach to handling an avalanche of data, and has since become the standard for storing, processing and analyzing hundreds of terabytes and even petabytes of data. Hadoop MapReduce is a broad scale, open source software framework devoted to scalable, distributed, data-intensive computing. The MapReduce framework breaks up large data into smaller parallelizable chunks and handles scheduling. If you can rewrite algorithms into Maps and Reduces, and your problem can be broken up into small pieces solvable in parallel, then Hadoop's MapReduce is the way to go for a distributed problem solving approach to large datasets. Map reduce framework is Fault tolerant, decisive and supports thousands of nodes and petabytes of data. The future trend in big data is Apache Hadoop2. It is the second iteration of the Hadoop framework for distributed data processing. And in today's hyper-connected world where more and more data is being created every day, Hadoop's breakthrough advantages mean that businesses and organizations can now find value in data that was recently considered useless.

REFERENCES

[1] M. A. Beyer and D. Laney, "The importance of‟ big data‟ : A definition," Gartner, Tech. Rep., 2012.

[2] X. Wu, X. Zhu, G. Q. Wu, et al., "Data mining with big data," IEEE Trans. on Knowledge and Data Engineering, vol. 26, no. 1, pp. 97-107, January 2014.Rajaraman and J. D. Ullman, "Mining of massive datasets," Cambridge University Press, 2012.

[3] Z. Zheng, J. Zhu, M. R. Lyu. "Service-generated Big Data and Big Data-as-a-Service: An Overview," in Proc. IEEE BigData, pp. 403-410, October 2013. A . Bellogín, I. Cantador, F. Díez, et al., "An empirical comparison of social, collaborative filtering, and hybrid recommenders," ACM Trans. on Intelligent Systems and Technology, vol. 4, no. 1, pp. 1-37, January 2013.

[4] W. Zeng, M. S. Shang, Q. M. Zhang, et al., "Can Dissimilar Users Contribute to Accuracy and Diversity of Personalized Recommendation?," International Journal of Modern Physics C, vol. 21, no.10, pp. 1217- 1227, June 2010.

[5] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, "Fuzzy c-Means Algorithms for Very Large Data," IEEE Trans. on Fuzzy Systems, vol. 20, no.6, pp. 1130-1146, December 2012.

[6] "HDFS Users Guide - Rack Awareness". Hadoop.apache.org. Retrieved 2013-10 17.

[7] "Cloud analytics: Do we really need to reinvent the storage stack?". IBM. June 2009.

[8] "HADOOP-6330: Integrating IBM General Parallel File System implementation of Hadoop File system interface". IBM. 2009-10-23.

[9] "Refactor the scheduler out of the JobTracker". Hadoop Common. Apache Soft ware Foundation. Retrieved 9 June 2012.

[10] M. Tim Jones (6 December 2011). "Scheduling in Hadoop". ibm.com. IBM. Retrieved 20 November 2013.

[11] "Under the Hood: Hadoop Distributed File system reliability with Namenode and Avatarnode". Facebook. Retrieved 2012-09-13.

[12] "Under the Hood: Scheduling MapReduce jobs more efficiently with Corona". Facebook. Retrieved 2012-11-9.

[13] "Zettaset Launches Version 4 of Big Data Management Solution, Delivering New Stability for Hadoop Systems and Productivity Boosting Features | | Zettaset.comZettaset.com". Zettaset.com. 2011-12-06. Retrieved 2012-05-23.

[14] Curt Monash. "More patent nonsense — Google MapReduce". dbms2.com. Retrieved 2010-03-07.

[15] D. Wegener, M. Mock, D. Adranale, and S. Wrobel, "Toolkit-Based High-Performance Data Mining of Large Data on MapReduce Clusters," Proc. Int'l Conf. Data Mining Workshops (ICDMW '09), pp. 296-301, 2009

[16] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," in Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6, ser. OSDI'04.

[17] http://www. slideshare.net/mcsrivas/design-scale-and- performance-of-ma prs-distribution-for-hadoop

[18] Z. Liu, P. Li, Y. Zheng, et al., "Clustering to find exemplar terms for keyphrase extraction," in Proc. 2009 Conf. on Empirical Methods in Natural Language Processing, pp. 257-266, May 2009.

[19] X. Liu, G. Huang, and H. Mei, "Discovering homogeneous web service community in the user-centric web environment," IEEE Trans. on Services Computing, vol. 2, no. 2, pp. 167-181, April-June 2009.

[20] K. Zielinnski, T. Szydlo, R. Szymacha, et al., "Adaptive so a solution stack," IEEE Trans. on Services Computing, vol. 5, no. 2, pp. 149-163, April-June 2012.

[21] F. Chang, J. Dean, S. mawat, et al., "Bigtable: A distributed storage system for structured data," ACM Trans. on Computer Systems, vol. 26, no. 2, pp. 1-39, June 2008.

[22] Z. Zheng, H. Ma, M. R. Lyu, et al., "QoS-aware Web service recommendation by collaborative filtering," IEEE Trans. on Services Computing, vol. 4, no. 2, pp.140-152, February 2011.

[23] R. S. Sandeep, C. Vinay, S. M. Hemant, "Strength and Accuracy Analysis of Affix Removal Stemming Algorithms," International Journal of Computer Science and Information Technologies, vol. 4, no. 2, pp. 265-269, April 2013.

[24] V. Gupta, G. S. Lehal, "A Survey of Common Stemming Techniques and Existing Stemmers for Indian Languages," Journal of Emerging Technologies in Web Intelligence, vol. 5, no. 2, pp. 157-161, May 2013.A. Rodriguez, W. A. Chaovalitwongse, L. Zhe L, et al., "Master defect record retrieval using network-based feature association," IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 40, no. 3, pp. 319-329, October 2010.

[25] T. Niknam, E. Taherian Fard, N. Pourjafarian, et al., "An efficient algorithm based on modified imperialist competitive algorithm and K-means for data clustering," Engineering Applications of Artificial Intelligence, vol 24, no. 2, pp. 306-317, March 2011.

[26] M. J. Li, M. K. Ng, Y. M. Cheung, et al. "Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters," IEEE Trans. on Knowledge and Data Engineering, vol. 20, no. 11, pp. 1519-1534, November 2008.

[27] G. Thilagavathi, D. Srivaishnavi, N. Aparna, et al., "A Survey on Efficient Hierarchical Algorithm used in Clustering," International Journal of Engineering, vol. 2, no. 9, September 2013.

[28] C. Platzer, F. Rosenberg, and S. Dustdar, "Web service clustering using multidimensional angles as proximity measures," ACM Trans. on Internet Technology, vol. 9, no. 3, pp. 11:1-11:26, July, 2009.

[29] G. Adomavicius, and J. Zhang, "Stability of Recommendation Algorithms," ACM Trans. on Information Systems, vol. 30, no. 4, pp. 23:1-23:31, August 2012.

[30] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," Information retrieval, vol. 5, no. 4, pp. 287-310, October 2002.

[31] Yamashita, H. Kawamura and K. Suzuki, "Adaptive Fusion Method for User-based and Item-based Collaborative Filtering," Advances in Complex Systems, vol. 14, no. 2, pp. 133-149, May 2011.

[32] D. Julie and K. A. Kumar, "Optimal Web Service Selection Scheme With Dynamic QoS Property Assignment," International Journal of Advanced Research In Technology, vol. 2, no. 2, pp. 69-75, May2012.

[33] J. Wu, L. Chen, Y. Feng, et al., "Predicting quality of service for selection by neighborhood-based collaborative filtering," IEEE Trans. on Systems, Man, and Cybernetics: Systems, vol. 43, no. 2, pp. 428-439, March 2013.

[34] White, Tom (10 May 2012). Hadoop: The Definitive Guide. O'Reilly Media. p. 3. ISBN 978-1-4493-3877-0.

[35] "Applications and organizations using Hadoop". Wiki.apache.org. 2013-06-19. Retrieved 2013-10-17.

[36] "HDFS User Guide". Hadoop.apache.org. Retrieved 2012-05-23.

[37] "HDFS Architecture". Retrieved 1 September 2013.

[38] "Improving MapReduce performance through data placement in heterogeneous Hadoop Clusters" (PDF). Eng.auburn.ed. April 2010.

[39] Y. Zhao, G. Karypis, and U. Fayyad, "Hierarchical clustering algorithms for document datasets," Data Mining and Knowledge Discovery, vol. 10, no. 2, pp.141-168, November 2005.