INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

BIG DATA ANALYSIS: A SURVEY

SUVARNA S. BHUSNAR, SWATI S. SANDEKAR, SUSHIL S.CHAVHAN

Department of Information technology YCCE, Nagpur.

Accepted Date: 15/03/2016; Published Date: 01/05/2016

Abstract: We live in on-demand, on-command Digital universe with data prolifering by Institutions, Individuals and Machines at a very high rate. This data is categories as "Big Data" due to its sheer Volume, Variety, Velocity and Veracity. Most of this data is unstructured, structured or semi structured and it is heterogeneous in nature. The volume and the heterogeneity of data with the speed it is generated, makes it difficult for the present computing infrastructure to manage Big Data. Traditional data management, warehousing and analysis systems fall short of tools to analyze this data. Due to its specific nature of Big Data, it is stored in distributed file system architectures. Hadoop and HDFS by Apache is widely used for storing and managing Big Data. Analyzing Big Data is a challenging task as it involves large distributed file systems which should be fault tolerant, flexible and scalable. Map Reduce is widely been used for the efficient analysis of Big Data. In this research paper the authors suggest various methods for catering to the problems in hand through Map Reduce framework over Hadoop Distributed File System (HDFS). Map Reduce is a Minimization technique which makes use of file indexing with mapping, sorting, shuffling and finally reducing. Map Reduce techniques have been studied at in this paper which is implemented for Big Data analysis using HDFS.

Keywords: Big Data Analysis, Big Data Management, Map Reduce, HDFS

Corresponding Author: MS. SUVARNA S. BHUSNAR

Access Online On:

www.ijpret.com

How to Cite This Article:

Suvarna S. Bhusnar, IJPRET, 2016; Volume 4 (9): 1066-1075

PAPER-QR CODE

INTRODUCTION

Big Data encompasses everything from click stream data from the web to genomic and proteomic data from biological research and medicines. Big Data is a heterogeneous mix of data both structured (traditional datasets –in rows and columns like DBMS tables, CSV's and XLS's) and unstructured data like e-mail attachments, manuals, images, PDF documents, medical records such as x-rays, ECG and MRI images, forms, rich media like graphics, video and audio, contacts, forms and documents. Businesses are primarily concerned with managing unstructured data, because over 80 percent of enterprise data is unstructured and require significant storage space and effort to manage. "Big data" refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyse.

Big data analyticsis the area where advanced analytic techniques operate on big data sets. It is really about two things,Big data and Analytics and how the two have teamed up to create one of the most profound trends in business intelligence. Map Reduce by itself is capable for analysing largedistributed data sets; but due to the heterogeneity, velocity and volume of Big Data, it is a challenge for traditional data analysis and management tools . A problem with Big Data is that they use NoSQL and has no Data Description Language (DDL) and it supports transaction processing. Also, web-scale data is not universal and it is heterogeneous. For analysis of Big Data, database integration and cleaning is much harder than the traditional mining approaches. Parallel processing and distributed computing is becoming a standard procedure which are nearly non-existent in RDBMS. Map Reduce has following characteristics ; it supports Parallel and distributed processing, it is simple and its architecture is shared-nothing which has commodity diverse hardware (big cluster).Its functions are programmed in a highlevel programming language (e.g. Java, Python) and it is flexible. Query processing is done through NoSQL integrated in HDFS as Hive tool.

Analytics helps to discover what has changed and the possible solutions. Second, advanced analytics is the best way to discover more business opportunities, new customer segments, identify the best suppliers, associate products of affinity, understand sales seasonality etc. Traditional experience in data warehousing, reporting, and online analytic processing (OLAP) is different for advanced forms of analytics. Organizations are implementing specific forms of analytics, particularly called advanced analytics. These are an collection of related techniques and tool types, usually including predictive analytics, data mining, statistical analysis, complex SQL, data visualization, artificial intelligence, natural language processing. Database analytics

platforms such as MapReduce, in-database analytics, in- memory databases, and columnar data stores are used for standardizing them. With big data analytics, the user is trying to discover new business facts that no one in the enterprise knew before, a better term would be "discovery analytics. To do that, the analyst needs large volumes of data with plenty of detail. This is often data that the enterprise has not yet tapped for analytics example, the log data. The analyst might mix that data with historic data from a data warehouse and would discover for example, new change behaviour in a subset of the customer base. The discovery would lead to a metric, report, analytic model, or some other product of BI, through which the company could track and predict the new form of customer behavioural change.

II. BIG DATA: OPPORTUNITIES AND CHALLENGES

In the distributed systems world, "Big Data" started to become a major issue in the late 1990‟s due to the impact of the world-wide Web and a resulting need to index and query its rapidly mushrooming content. Database technology (including parallel databases) was considered for the task.

The turn of the millennium then brought further challenges as companies began to use information such as the topology of the Web and users‟ search histories in order to provide increasingly useful search results, as well as more effectively-targeted advertising to display alongside and fund those results. Google‟s technical response to the challenges of Web-scale data management and analysis was simple, by database standards, but kicked off what has become the modern "Big Data" revolution in the systems world . To handle the challenge of Web-scale storage, the Google File System (GFS) was created . GFS provides clients with the familiar OSlevel byte-stream abstraction, but it does so for extremely large files whose content can span hundreds of machines in shared nothing clusters created using inexpensive commodity hardware. To handle the challenge of processing the data in such largefiles, Google pioneered its Map Reduce programming model and platform . This model, characterized by some as "parallel programming for dummies", enabled Google‟s developers to process large collections of data by writing two user-defined functions, map and reduce, that the Map Reduce framework applies to the instances (map) and sorted groups of instances that share a common key (reduce) – similar to the sort of partitioned parallelism utilized in shared- nothing parallel query processing. Driven by very similar requirements, software developers at Yahoo!, Facebook, and other large Web companies followed suit. Taking Google‟s GFS and Map Reduce papers as rough technical specifications, open-source equivalents were developed, and the Apache Hadoop Map Reduce platform and its underlying file system (HDFS, the Hadoop

Distributed File System) were born . The Hadoop system has quickly gained traction, and it is now widely used for use cases including click stream and log analysis, and certain large-scale information extraction and machine learning tasks.. Pig is relational-algebra-like in nature, and is reportedly used for over 60% of Yahoo!‟s MapReduce use cases; Hive is SQL-inspired and reported to be used for over 90% of the Facebook Map Reduce use cases. Microsoft‟s technologies include a parallel runtime system called Dryad and two higher-level programming models, Dryad LINQ and the SQLlike SCOPE language , which utilizes Dryad under the covers. Interestingly, Microsoft has also recently announced that its future “Big Data” strategy includes support for Hadoop.

III. WHAT IS BIG DATA? Big Data concept means a datasets which continues to grow so much that it becomes difficult to manage it using existing database management concepts & tools. The difficulty can be related to data capture, storage, search, sharing, analytics and visualization etc. The Big Data spans across three dimensions: Volume, Velocity and Variety.

Volume – The size of data is very large and in terabytes and petabytes.

Velocity – It should be used when streaming in to the enterprise in order to maximize its value to the business. The role of time is very critical here.

Variety – It extends beyond the structured data, including unstructured data of all varieties: text, audio, video, posts, log files etc.

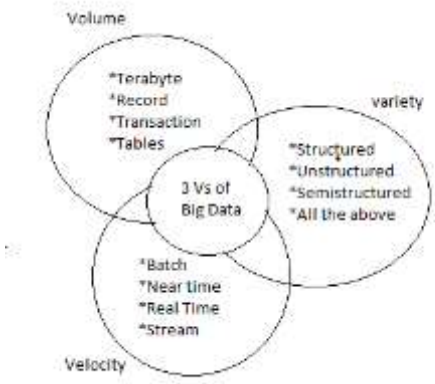
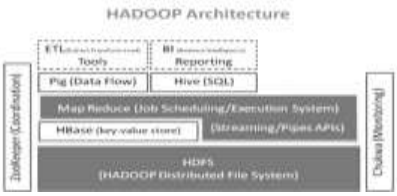


Figure 1. The three Vs of Big Data

IV. HADOOP AND HDFS

Hadoop is a scalable, open source, fault-tolerant Virtual Grid operating system architecture for data storage and processing. It runs on commodity hardware, it uses HDFS which is fault-tolerant high-bandwidth clustered storage architecture. It runs MapReduce for distributed data processing and is works with structured and unstructured data.

Figure2 Illustrates the layers found in the software architecture of aHadoop stack . At the bottom of the Hadoop software stack is HDFS, a distributed file system in which each file appears as a (very large) contiguous and randomly addressable sequence of bytes. For batch analytics, the middle layer of the stack is the Hadoop Map Reduce system, which applies map operations to the data in partitions of an HDFS file, sorts and redistributes the results based on key values in the output data, and then performs reduce operations on the groups of output data items with matching keys from the map phase of the job. For applications just needing basic key-based record management operations, the HBase store (layered on top of HDFS) is available as a key-value layer in the Hadoop stack. As indicated in the figure, the contents of HBase can either be directly accessed and manipulated by a client application or accessed via Hadoop for analytical needs. Many users of the Hadoop stack prefer the use of a declarative language over the bare MapReduce programming model. High-level language compilers (Pig and Hive) are thus the topmost layer in the Hadoop software stack for such clients.



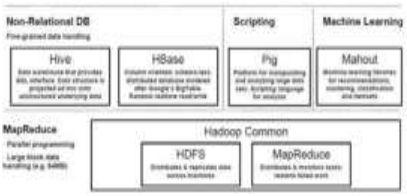


Fig 2.Hadoop Architecture Layers Fig 3.Hadoop Architecture Tools and usage

Figure3 shows the relevancy between the traditional experience in data warehousing, reporting, and online analytic processing (OLAP) and advanced analytics with collection of related techniques like data mining with DBMS, artificial intelligence, machine learning, and database analytics platforms such as MapReduce and Hadoop over HDFS .

V. BIG DATA ANALYSIS

Heterogeneity, scale, timeliness, complexity, and privacy problems with Big Data hamper the progress at all phases of the process that can create value from data. Much data today is not natively in structured format; for example, tweets and blogs are weakly structured pieces of text, while images and video are structured for storage and display, but not for semantic content and search: transforming such content into a structured format for later analysis is a major challenge . The value of data enhances when it can be linked with other data, thus data integration is a major creator of value. Since most data is directly generated in digital format today, we have the opportunity and the challenge both to influence the creation to facilitate later linkage and to automatically link previously created data. Data analysis, organization, retrieval, and modelling are other foundational challenges. Big Data analysis is a clear bottleneck in many applications, both due to lack of scalability of the underlying algorithms and due to the complexity of the data that needs to be analysed. Finally, presentation of the results and its interpretation by non- technical domain experts is crucial to extracting actionable knowledge as most of the BI related jobs are handled by statisticians and not software experts.

VI. MAP REDUCE

MapReduce [1-2] is a programming model for processing large-scale datasets in computer clusters. The MapReduce programming model consists of two functions, map() and reduce(). Users can implement their own processing logic by specifying a customized map() and reduce() function. The map() function takes an input key/value pair and produces a list of intermediate key/value pairs. The MapReduce runtime system groups together all intermediate pairs based on the intermediate keys and passes them to reduce() function for producing the final results. Map (in_key, in_value) ---

>list(out_key,intermediate_value) Reduce (out_key,list(intermediate_value)) -- ->list(out_value) The signatures of map() and reduce() are as follows : map (k1,v1)

! list(k2,v2)and reduce (k2,list(v2)) ! list(v2)

A MapReduce cluster employs a master-slave architecture where one master node manages a number of slave nodes . In the Hadoop, the master node is called JobTracker and the slave node is called TaskTracker as shown in the figure 4. Hadoop launches a MapReduce job by first splitting the input dataset into even-sized data blocks. Each data block is then scheduled to one TaskTracker node and is processed by a map task. The TaskTracker node notifies the JobTracker

when it is idle. The scheduler then assigns new tasks to it. The scheduler takes data locality into account when it disseminates data blocks.

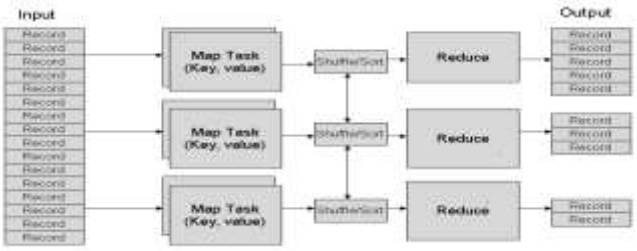


Figure 4. Map Reduce Architecture and Working

It always tries to assign a local data block to a TaskTracker. If the attempt fails, the scheduler will assign a rack-local or random data block to the TaskTracker instead. When map () functions complete, the runtime system groups all intermediate pairs and launches a set of reduce tasks to produce the final results. Large scale data processing is a difficult task, managing hundreds or thousands of processors and managing parallelization and distributed environments makes is more difficult. Map Reduce provides solution to the mentioned issues, as is supports distributed and parallel I/O scheduling, it is fault tolerant and supports scalability and i has inbuilt processes for status and monitoring of heterogeneous and large datasets as in Big Data.

A. Map Reduce Components

1. Name Node – manages HDFS metadata, doesn‟t deal with files directly

2. Data Node – stores blocks of HDFS – default replication level for each block: 3

3. Job Tracker – schedules, allocates and monitors job execution on slaves – Task Trackers

4. Task Tracker – runs Map Reduce operations

B. Map Reduce techniques

Combiners provide a general mechanism within the MapReduce framework to reduce the amount of intermediate data generated by the mappers. They can be understood as "mini-reducers" that process the output of mappers. The combiner's aggregate term counts across the documents processed by each map task. This result in a reduction in the number of intermediate key-value pairs that need to be shuffled across the network, from the order of total number of terms in the collection to the order of the number of unique terms in the

collection. They reduce the result size of map functions and perform reduce-like function in each machine which decreases the shuffling cost.

Inverse indexing is a technique in which the keywords of the documents are mapped according to the document keys in which they are residing. For example

Doc1: IMF, Financial Economics Crisis

Doc2: IMF, Financial Crisis

Doc3: Harry Economics

Doc4: Financial Harry Potter Film

Doc5: Harry Potter Crisis

The following is the inverted index of the above data IMF -> Doc1:1,

Doc2:1

Financial -> Doc1:6, Doc2:6, Doc4:1

Economics -> Doc1:16, Doc3:7

Crisis -> Doc1:26, Doc2:16, Doc5:14

Harry -> Doc3:1, Doc4:11, Doc5:1

Potter -> Doc4:17, Doc5:7

Film -> Doc4:24

Shuffling is the procedure of mixing the indexes of the files and their keys, so that a heterogeneous mix of dataset can be obtained. If the dataset is shuffled, then there are better chances that the resultant query processing will yield near accurate results. We can relate the shuffling process with the population generating by crossover in the GA algorithms. The processes are different in nature, but their purpose is similar.

It is a term used to distribute the Mappers in the HDFS architecture. Sharding refers to the groupings or documents which are done so that the MapReduce jobs are done parallel in a distributed environment.

Join is a RDBMS term; it refers to combining two or more discrete datasets to get Cartesian product of data of all the possible combinations. Map Reduce does not have its own

Join techniques, but RDBMS techniques are tweaked and used to get the maximum possible combinations.The join techniques which are adopted for Map Reduce are Equi Join, Self Join, Repartition Join and Theta Join. & Classification

They are Data Analysis term, used mainly in Data Mining. In Map Reduce it is achieved through K means clustering . Here, iterative working improves partitioning of data into k clusters. After the clustering, the data sorted are grouped together based upon rules to be formed into classes.

The steps for clustering in Map Reduce are; Step1: Do

Step2: Map

Step3: Input is a data point and k centres are broadcasted

Step4: Finds the closest centre among k centres for the input point

Step5: Reduce

Step6: Input is one of k centres and all data points having this centre as their closest centre

Step7: Calculates the new centre using data points

Step 8: Repeat 1-7, until all of new centres are not changed.

VII. CONCLUSION

The need to process enormous quantities of data has never been greater. Not only are terabyte- and petabyte-scale datasets rapidly becoming commonplace, but there is consensus that great value lies buried in them, waiting to be unlocked by the right computational tools. In the commercial sphere, business intelligence, driven by the ability to gather data from a dizzying array of sources. Big Data analysis tools like Map Reduce over Hadoop and HDFS, promises to help organizations better understand their customers and the marketplace, hopefully leading to better business decisions and competitive advantages [6]. For engineers building information processing tools and applications, large and heterogeneous datasets which are generating continuous flow of data, lead to more effective algorithms for a wide range of tasks, from machine translation to spam detection. In the natural and physical sciences, the

ability to analyse massive amounts of data may provide the key to unlocking the secrets of the cosmos or the mysteries of life. MapReduce can be exploited to solve a variety of problems related to text processing at scales that would have been unthinkable a few years ago .

REFERENCES

1. Jefry Dean and Sanjay Ghemwat, MapReduce:A Flexible Data Processing Tool, Communications of the ACM, Volume 53, Issuse.1,January 2010, pp 72-77.
2. Jefry Dean and Sanjay Ghemwat,.MapReduce: Simplified data processing on large clusters, Communications of the ACM, Volume 51 pp. 107–113, 2008
3. Brad Brown, Michael Chui, and James Manyika, Are you ready for the era of „big data"?,McKinseyQuaterly,Mckinsey Global Institute, October 2011.
4. DunrenChe, MejdlSafran, and ZhiyongPeng, From Big Data to Big Data Mining: Challenges, Issues, and Opportunities, DASFAA Workshops 2013, LNCS 7827, pp. 1–15, 2013.
5. MarcinJedyk, MAKING BIG DATA, SMALL, Using distributed systems for processing, analysing and managing large huge data sets, Software Professional"s Network, Cheshire Data systems Ltd.
6. OnurSavas, YalinSagduyu, Julia Deng, and Jason Li,Tactical Big Data Analytics: Challenges, Use Cases and Solutions, Big Data Analytics Workshop in conjunction with ACM Sigmetrics 2013,June 21, 2013.
7. Kyuseok Shim, MapReduce Algorithms for Big Data Analysis, DNIS 2013, LNCS 7813, pp. 44–48, 2013.
8. Raja. Appuswamy,ChristosGkantsidis,DushyanthNarayanan,OrionHo dson,AntonyRowstron, Nobody ever got fired for buying a cluster, Microsoft Research, Cambridge, UK, Technical Report, MSR-TR- 2013-2
9. Carlos Ordonez, Algorithms and Optimizations for Big Data Analytics: Cubes, Tech Talks, University of Houston, USA. [10] Spyros Blanas, Jignesh M. Patel,VukErcegovac, Jun Rao,Eugene J. Shekita, YuanyuanTian, A Comparison of Join Algorithms for Log Processing in MapReduce, SIGMOD"10, June 6–11, 2010, Indianapolis, Indiana, USA.
10. J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in USENIXSymposium on Operating Systems Design and Implementation, San Francisco, CA, Dec. 2004, pp. 137–150.
11. S. Ghemawat, H. Gobioff, and S. Leung, "The Google File System." in ACM Symposium on Operating Systems Principles, Lake George, NY, Oct 2003, pp. 29 – 43.