

An Emergence Techniques In Big Data Mining

Hemant N. Randhir

Department of Information, RCPIT Shirpur, Maharashtra, India

ABSTRACT

The term 'Big Data' describes innovative techniques and technologies to capture, store, distribute, manage and analyze petabyte- or larger-sized datasets with high-velocity and different structures. Big data can be structured, unstructured or semi-structured, resulting in incapability of conventional data management methods. Data is generated from various different sources and can arrive in the system at various rates. In order to process these large amounts of data in an inexpensive and efficient way, parallelism is used. Big Data is a data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it. Hadoop is the core platform for structuring Big Data, and solves the problem of making it useful for analytics purposes. Hadoop is an open source software project that enables the distributed processing of large data sets across clusters of commodity servers. It is designed to scale up from a single server to thousands of machines, with a very high degree of fault tolerance.

Keywords: *Big Data, Hadoop, Map Reduce, HDFS, Hadoop Components*

I. INTRODUCTION

A. Big Data: Definition

Big data is a term that refers to data sets or combinations of data sets whose size (volume), complexity (variability), and rate of growth (velocity) make them difficult to be captured, managed, processed or analyzed by conventional technologies and tools, such as relational databases and desktop statistics or visualization packages, within the time necessary to make them useful. While the size used to determine whether a particular data set is considered big data is not firmly defined and continues to change over time, most analysts and practitioners currently refer to data sets from 30-50 terabytes(10¹² or 1000 gigabytes per terabyte) to multiple petabytes (10¹⁵ or 1000 terabytes per petabyte) as big data. Figure No. 1.1 gives Layered Architecture of Big Data System. It can be decomposed into three layers, including Infrastructure Layer, Computing Layer, and Application Layer from top to bottom.

B. 3 Vs of Big Data

Volume of data: Volume refers to amount of data. Volume of data stored in enterprise repositories have grown from megabytes and gigabytes to petabytes.

Variety of data: Different types of data and sources of data. Data variety exploded from structured and legacy data stored in enterprise repositories to unstructured, semi structured, audio, video, XML etc.

Velocity of data: Velocity refers to the speed of data processing. For time-sensitive processes such as catching fraud, big data must be used as it streams into your enterprise in order to maximize its value.

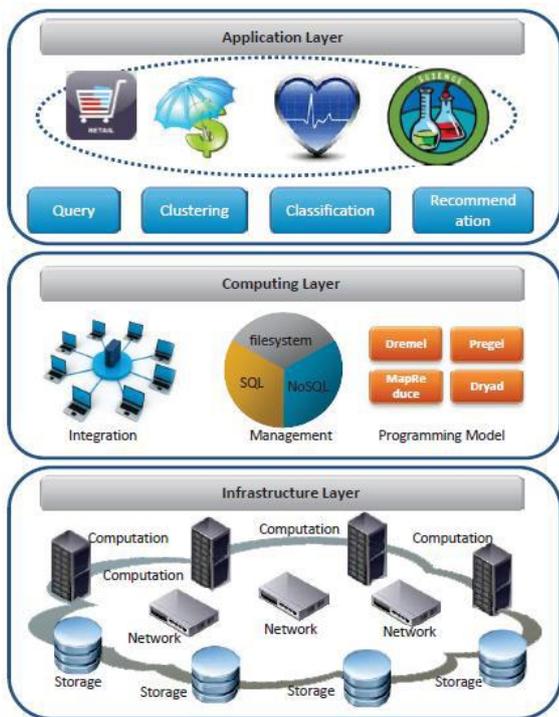


Figure 1: Layered Architecture of Big Data System

C. Problem with Big Data Processing

i. Heterogeneity and Incompleteness

When humans consume information, a great deal of heterogeneity is comfortably tolerated. In fact, the nuance and richness of natural language can provide valuable depth. However, machine analysis algorithms expect homogeneous data, and cannot understand nuance. In consequence, data must be carefully structured as a first step in (or prior to) data analysis. Computer systems work most efficiently if they can store multiple items that are all identical in size and structure. Efficient representation, access, and analysis of semi-structured data require further work.

ii. Scale

Of course, the first thing anyone thinks of with Big Data is its size. After all, the word “big” is there in the very name. Managing large and rapidly increasing volumes of data has been a challenging issue for many decades. In the past, this challenge was mitigated by processors getting faster, following Moore’s law, to provide us with the resources needed to cope with increasing volumes of data. But, there is a fundamental shift underway now: data volume is scaling faster than compute resources, and CPU speeds are static.

iii. Timeliness

The flip side of size is speed. The larger the data set to be processed, the longer it will take to analyze. The design of a system that effectively deals with size is likely also to result in a system that can process a given size of data set faster. However, it is not just this speed that is usually meant when one speaks of Velocity in the context of Big Data. Rather, there is an acquisition rate challenge

iv. Privacy

The privacy of data is another huge concern, and one that increases in the context of Big Data. For electronic health records, there are strict laws governing what can and cannot be done. For other data, regulations, particularly in the US, are less forceful. However, there is great public fear regarding the inappropriate use of personal data, particularly through linking of data from multiple sources. Managing privacy is effectively both a technical and a sociological problem, which must be addressed jointly from both perspectives to realize the promise of big data.

v. Human Collaboration

In spite of the tremendous advances made in computational analysis, there remain many patterns that humans can easily detect but computer algorithms have a hard time finding. Ideally, analytics for Big Data will not be all computational rather it will be designed explicitly to have a human in the loop. The new sub-field of visual analytics is attempting to do this, at least with respect to the modeling and analysis phase in the pipeline. In today’s complex world, it often takes multiple experts from different domains to really understand what is going on. A Big Data analysis system must support input from multiple human experts, and shared exploration of results. These multiple experts may be separated in space and time when it is too expensive to assemble an entire team together in one room. The data system has to accept this distributed expert input, and support their collaboration.

2. Hadoop: Solution for Big Data Processing

Hadoop is a Programming framework used to support the processing of large data sets in a distributed

computing environment. Hadoop was developed by Google's MapReduce that is a software framework where an application break down into various parts. The Current Apache Hadoop ecosystem consists of the Hadoop Kernel, MapReduce, HDFS and numbers of various components like Apache Hive, Base and Zookeeper. HDFS and MapReduce are explained in following points.

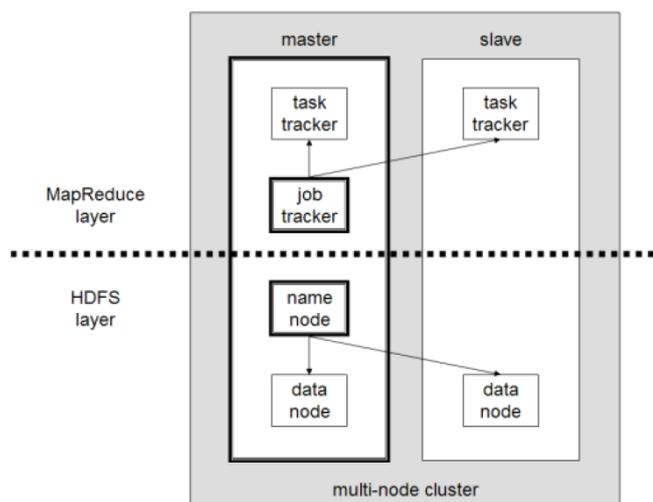


Figure 2: Hadoop Architecture

A. HDFS Architecture

Hadoop includes a fault-tolerant storage system called the Hadoop Distributed File System, or HDFS. HDFS is able to store huge amounts of information, scale up incrementally and survive the failure of significant parts of the storage infrastructure without losing data. Hadoop creates clusters of machines and coordinates work among them. Clusters can be built with inexpensive computers. If one fails, Hadoop continues to operate the cluster without losing data or interrupting work, by shifting work to the remaining machines in the cluster. HDFS manages storage on the cluster by breaking incoming files into pieces, called “blocks,” and storing each of the blocks redundantly across the pool of servers. In the common case, HDFS stores three complete copies of each file by copying each piece to three different servers.

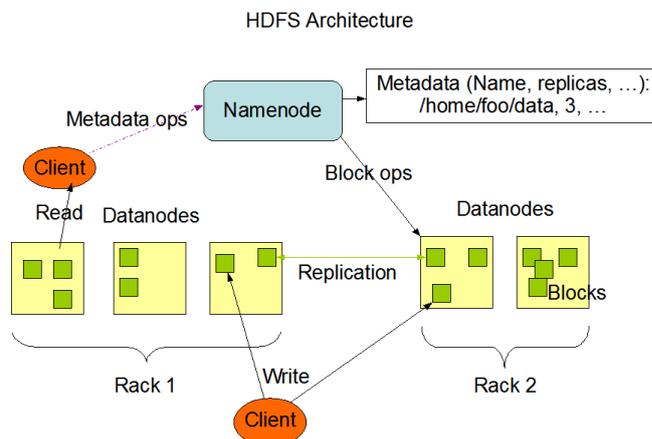


Figure 3: HDFS Architecture

B. MapReduce Architecture

The processing pillar in the Hadoop ecosystem is the MapReduce framework. The framework allows the specification of an operation to be applied to a huge data set, divide the problem and data, and run it in parallel. From an analyst's point of view, this can occur on multiple dimensions. For example, a very large dataset can be reduced into a smaller subset where analytics can be applied. In a traditional data warehousing scenario, this might entail applying an ETL operation on the data to produce something usable by the analyst. In Hadoop, these kinds of operations are written as MapReduce jobs in Java. There are a number of higher level languages like Hive and Pig that make writing these programs easier. The outputs of these jobs can be written back to either HDFS or placed in a traditional data warehouse. There are two functions in MapReduce as follows:

- map** – the function takes key/value pairs as input and generates an intermediate set of key/value pairs
- reduce** – the function which merges all the intermediate values associated with the same intermediate key

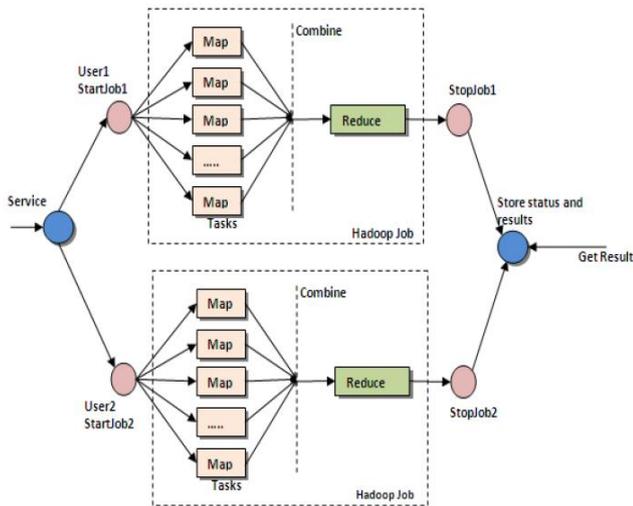


Figure 4: MapReduce Architecture

II. LITERATURE REVIEW

S. Vikram Phaneendra & E. Madhusudhan Reddy

et.al. Illustrated that in olden days the data was less and easily handled by RDBMS but recently it is difficult to handle huge data through RDBMS tools, which is preferred as “big data”. In this they told that big data differs from other data in 5 dimensions such as volume, velocity, variety, value and complexity. They illustrated the hadoop architecture consisting of name node, data node, edge node, HDFS to handle big data systems. Hadoop architecture handle large data sets, scalable algorithm does log management application of big data can be found out in financial, retail industry, health-care, mobility, insurance. The authors also focused on the challenges that need to be faced by enterprises when handling big data: - data privacy, search analysis, etc [1].

Kiran kumara Reddi & Dnysl Indira et.al. Enhanced us with the knowledge that Big Data is combination of structured , semi-structured ,unstructured homogenous and heterogeneous data .The author suggested to use nice model to handle transfer of huge amount of data over the network .Under this model, these transfers are relegated to low demand periods where there is ample ,idle bandwidth available . This bandwidth can then be repurposed for big data transmission without impacting other users in system. The Nice model uses a store –and-forward approach by utilizing staging servers. The model is able to accommodate differences in time zones and variations in bandwidth. They suggested that new algorithms are required to transfer

big data and to solve issues like security, compression, routing algorithms [2].

Jimmy Lin et.al. used Hadoop which is currently the large –scale data analysis “ hammer” of choice, but there exists classes of algorithms that aren’t “ nails” in the sense that they are not particularly amenable to the MapReduce programming model . He focuses on the simple solution to find alternative non-iterative algorithms that solves the same problem. The standard MapReduce is well known and described in many places .Each iteration of the pagerank corresponds to the MapReduce job. The author suggested iterative graph, gradient descent & EM iteration which is typically implemented as Hadoop job with driven set up iteration &Check for convergences. The author suggests that if all you have is a hammer, throw away everything that’s not a nail [3].

Wei Fan & Albert Bifet et.al. Introduced Big Data Mining as the capability of extracting Useful information from these large datasets or streams of data that due to its Volume, variability and velocity it was not possible before to do it. The author also started that there are certain controversy about Big Data. There certain tools for processes. Big Data as such hadoop, strom, apache S4. Specific tools for big graph mining were PEGASUS & Graph. There are certain Challenges that need to death with as such compression, visualization etc.[4].

Albert Bifet et.al. Stated that streaming data analysis in real time is becoming the fastest and most efficient way to obtain useful knowledge, allowing organizations to react quickly when problem appear or detect to improve performance. Huge amount of data is created everyday termed as “ big data”. The tools used for mining big data are apache hadoop, apache big, cascading, scribe, storm, apache hbase, apache mahout, MOA, R, etc. Thus, he instructed that our ability to handle many exabytes of data mainly dependent on existence of rich variety dataset, technique, software framework [5].

Bernice Purcell et.al. Started that Big Data is comprised of large data sets that can’t be handle by traditional systems. Big data includes structured data, semi-structured and unstructured data. The data storage technique used for big data includes multiple clustered network attached storage (NAS) and object based

storage. The Hadoop architecture is used to process unstructured and semi-structured using map reduce to locate all relevant data then select only the data directly answering the query. The advent of Big Data has posed opportunities as well challenges to business [6].

Sameer Agarwal et.al. Presents a BlinkDB, a approximate query engine for running interactive SQL queries on large volume of data which is massively parallel. BlinkDB uses two key ideas: (1) an adaptive optimization framework that builds and maintains a set of multi-dimensional stratified samples from original data over time, and (2) A dynamic sample selection strategy that selects an appropriately sized sample based on a query's accuracy or response time requirements [7].

Yingyi Bu et.al. Used a new technique called as HaLoop which is modified version of Hadoop MapReduce Framework, as Map Reduce lacks built-in-support for iterative programs HaLoop allows iterative applications to be assembled from existing Hadoop programs without modification, and significantly improves their efficiency by providing inter- iteration caching mechanisms and a loop-aware scheduler to exploit these caches. He presents the design, implementation, and evaluation of HaLoop, a novel parallel and distributed system that supports large-scale iterative data analysis applications. HaLoop is built on top of Hadoop and extends it with a new programming model and several important optimizations that include (1) a loop-aware task scheduler, (2) loop-invariant data caching, and (3) caching for efficient fix point verification [8].

Shadi Ibrahim et.al. Project says presence of partitioning skew causes a huge amount of data transfer during the shuffle phase and leads to significant unfairness on the reduce input among different data nodes In this paper, author develop a novel algorithm named LEEN for locality aware and fairness-aware key partitioning in MapReduce. LEEN embraces an asynchronous map and reduce scheme. Author has integrated LEEN into Hadoop. His experiments demonstrate that LEEN can efficiently achieve higher locality and reduce the amount of shuffled data. More importantly, LEEN guarantees fair distribution of the reduce inputs. As a result, LEEN achieves a performance improvement of up to 45% on different workloads. To tackle all this he presents a

present a technique for Handling Partitioning Skew in MapReduce using LEEN [9].

Kenn Slagter et.al. Proposes an improved partitioning algorithm that improves load balancing and memory consumption. This is done via an improved sampling algorithm and partitioner. To evaluate the proposed algorithm, its performance was compared against a state of the art partitioning mechanism employed by Tera Sort as the performance of MapReduce strongly depends on how evenly it distributes this workload. This can be a challenge, especially in the advent of data skew. In MapReduce, workload distribution depends on the algorithm that partitions the data. One way to avoid problems inherent from data skew is to use data sampling. How evenly the partitioner distributes the data depends on how large and representative the sample is and on how well the samples are analyzed by the partitioning mechanism. He uses an improved partitioning mechanism for optimizing massive data analysis using MapReduce for evenly distribution of workload [10].

III. CONCLUSION

We have entered an era of Big Data. The paper describes the concept of Big Data along with 3 Vs, Volume, Velocity and variety of Big Data. The paper also focuses on Big Data processing problems. These technical challenges must be addressed for efficient and fast processing of Big Data. The challenges include not just the obvious issues of scale, but also heterogeneity, lack of structure, error-handling, privacy, timeliness, provenance, and visualization, at all stages of the analysis pipeline from data acquisition to result interpretation. These technical challenges are common across a large variety of application domains, and therefore not cost-effective to address in the context of one domain alone. The paper describes Hadoop which is an open source software used for processing of Big Data.

IV. REFERENCES

- [1]. S.Vikram Phaneendra & E.Madhusudhan Reddy "Big Data- solutions for RDBMS problems- A survey" In 12th IEEE/IFIP Network Operations & Management Symposium (NOMS 2010) (Osaka, Japan, Apr 19{23 2013).

- [2]. Kiran kumara Reddi & Dnvsl Indira “Different Technique to Transfer Big Data : survey” IEEE Transactions on 52(8) (Aug.2013) 2348 { 2355 }
- [3]. Jimmy Lin “MapReduce Is Good Enough?” The control project. IEEE Computer 32 (2013).
- [4]. Umasri.M.L, Shyamalagowri.D ,Suresh Kumar.S “Mining Big Data:- Current status and forecast to the future” Volume 4, Issue 1, January 2014 ISSN: 2277 128X
- [5]. Albert Bifet “Mining Big Data In Real Time” Informatica 37 (2013) 15–20 DEC 2012
- [6]. Bernice Purcell “The emergence of “big data” technology and analytics” Journal of Technology Research 2013.
- [7]. Sameer Agarwal†, Barzan MozafariX, Aurojit Panda†, Henry Milner†, Samuel MaddenX, Ion Stoica “BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data” Copyright © 2013 ACM 978-1-4503-1994 2/13/04
- [8]. Yingyi Bu _ Bill Howe _ Magdalena Balazinska _ Michael D. Ernst “The HaLoop Approach to Large-Scale Iterative Data Analysis” VLDB 2010 paper “HaLoop: Efficient Iterative Data Processing on Large Clusters.
- [9]. Shadi Ibrahim* _ Hai Jin _ Lu Lu “Handling Partitioning Skew in MapReduce using LEEN” ACM 51 (2008) 107–113
- [10]. Kenn Slagter · Ching-Hsien Hsu “An improved partitioning mechanism for optimizing massive data analysis using MapReduce” Published online: 11 April 2013