Big Data An Overview

K.R. Dabhade

Abstract: The Data sets that are too large and complex to manipulate or interrogate with standard methods or tools so it cannot be processed using some conventional methods. Now a days social networks, mobile phones, sensors and science contribute to pet bytes of data created daily. Creators of web search engines were among the first to confront this problem. We've all heard a lot about "big data," but "big" is really a red herring. Companies like telecommunication, and other data-centric industries have had huge datasets for a long time. The storage capacity continues to expand, today's "big" is certainly tomorrow's "medium" and next week's"small." or it can be defined as "big data" is when the size of the data itself becomes part of the problem. The process of research into massive amounts of data to reveal hidden patterns and secret correlations named as big data analytic. We're discussing data problems ranging from gigabytes to petabytes of data. These useful informations for companies or organizations with the help of gaining richer and deeper insights and getting an advantage over the competition. Hence big data implementations need to be analyzed and executed as accurately as possible. At some point, traditional techniques for working with data run out of steam. The information platforms are similar to traditional data warehouses, but different. Some rich APIs, are designed for exploring and understanding the data rather than for traditional analysis and reporting.

I. Introduction

In few years, there has been tremendous amount of data explosion that's available. Big data can be analyzed in many ways like storing acquiring, processing of data. Big data characteristics depends on three V like velocity, volume and veracity of the data. When big data is processed and stored the additional dimensions come into play, such as governance, policies and security. Choosing an architecture and building an appropriate big data solution is challenging because so many factors have to be considered. The data may be as of web server logs, streams, online transaction records, The data from government data sensors or some other source, here we need figure out what to do with such a data. And not just companies using own data or the data contributed by their users. Now a days it is common to mash-up data from a number of sources.

II. Necessity

The organizations now a days which have built data platforms have found necessary to go beyond the relational database model. The database systems stop being effective at this scale as like traditional relational database. Hence Managing sharing and replication across a horde of database servers is difficult and slow. Hence the need to define a schema in advance conflicts with reality of multiple and unstructured data sources in which you may not know what's important until after you've analyzed the data. The Relational databases are designed for consistency to support complex transactions that can easily be rolled back if any one of a complex set of operations fails. So as to store huge datasets effectively, These are frequently called NoSQL databases, or Non-Relational databases, neither term is very useful. Many of these databases are the logical descendants of Amazon's Dynamo and Google's BigTable and are designed to be distributed across many nodes so as to provide the eventual consistency but not absolute consistency to have very flexible schema.

- *Prof. K.R. Dabhade , Information Technology Department, P.E.S. College of Engineering, Aurangabad, India. Dabhade.karan871@gmail.com*
- *Prof. , Information Technology Department, P.E.S. College of Engineering, Aurangabad, India.*

III. Objectives

Every data is useful only if we can do something of it, and the enormous datasets present some computational problems. Google had popularized the MapReduce approach, basically a divide-and-conquer strategy for distributing an large problem across an extremely large computing cluster. In the initial "map" stage a programming task is divided into a number of identical parts or subtasks these subtasks are then distributed across many processors the obtained intermediate results are then combined by a single reduce task. MapReduce is a programming model and an associated implementation for processing and generating large data sets. It seems an obvious solution to biggest problem of creating large searches in Google's. As it is easy to distribute a search across thousands of processors and then combine the results into a single set. The less obvious is that MapReduce has proven to be widely applicable to many large data problems ranging from search to machine learning. Architecturally, the reason we are able to ask complicated computational questions is because we got all of these processors which are working in parallel, harnessed together and the reason we are able to deal with lots of data is because Hadoop spreads it out. Our goal is to examine a broad range of applications we have participant from well established industries as well as new companies whose concept of business is to analyze data.

IV. Theme

Using of data effectively requires something different from traditional statistics. Developing new software platforms for storing and processing massive amounts of data and for applying analytics beyond what conventional relational systems can do. We see a "sea change" happening as analysis moves from the simple SQL aggregation capabilities to much more complex routines to perform data clustering predictive modeling and complex statistics. we focused on building array-oriented DBMSes because relational systems are not good at these linear algebra operations as they are specified on arrays not tables. The things that differentiates data science from statistics is that data science is a holistic approach here we are increasingly finding data in the wild and data scientists are involved with gathering data and massaging it into a tractable form making it tell its story, and presenting that story to others. To meet the challenge of processing such large data sets, Google created Map-Reduce. Google's work and Yahoo's

creation of the Hadoop MapReduce implementation has spawned an ecosystem of big data processing tools. We're building data processing systems that facilitate rapid processing data. Big Data is a "Big Velocity" problem that requires data conditioning at high rates including the abilities to aggregate data at high speeds and load it into database management systems.

a. Literature Survey

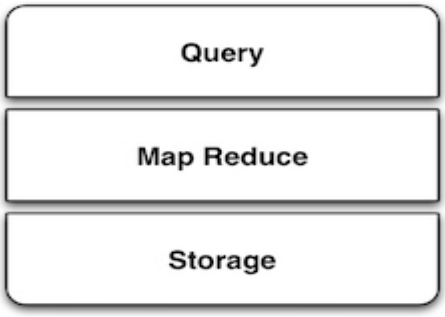
Now a days data is everywhere like government, web server, business partners. We are finding that almost everything can (or has) been instrumented. The critical issue about the Big data is the privacy and security. Big data samples describe the review about the atmosphere, biological science and research.

- Storage Map Reduce Big data is data that becomes large enough that it cannot be processed using conventional methods. So creators of data such as social networks, sensors data, mobile networks and science contribute to petabytes of data created daily.

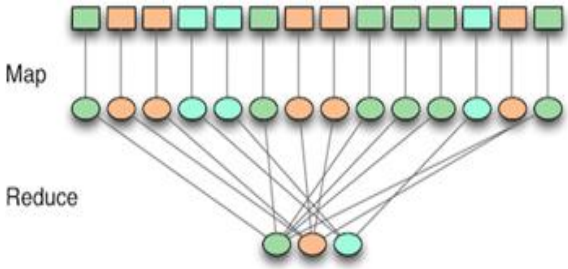
- Processing such large data sets the Google created Map-Reduce.The togethere work of Google's and Yahoo's creation of the Hadoop MapReduce implementation has spawned an ecosystem of big data processing tools.

- A stack for big data systems has emerged as Map Reduce has grown in popularity which comprising layers of Storage, Map Reduce and Query (SMAQ).

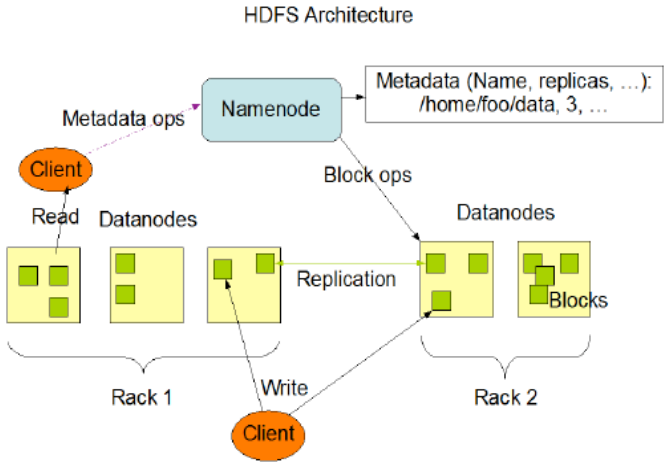
- The SMAQ systems are typically open source, distributed, and run on commodity hardware.



- The Map Reduce is created at google in response to the problem of creating web search indexes the Map Reduce framework is the powerhouse behind most of today's big data processing.
- The innovation of MapReduce is the ability to take a query over a data set divide it and run it in parallel over many nodes.



- Loading the data—This operation is more properly called Extract Transform Load (ETL) in data warehousing terminology. Data must be extracted from its source & structured to make it ready for processing.
- **MapReduce**—This phase will retrieve data from storage, process it, and return the results to the storage.
- **Extracting the result**—Once processing is complete, for the result to be useful to humans, it must be retrieved from the storage and presented.
- Many SMAQ systems have features designed to simplify the operation of each of these stages.
- **Storage-**MapReduce requires storage from which to fetch data and in which to store the results of the computation. The data expected by MapReduce is not relational data, as used by conventional databases. Instead, data is consumed in chunks, which are then divided among nodes and fed to the map phase as key value pairs. This data does not require a schema, and may be unstructured.
- **Hadoop** is dominant open source map reduce implementation funded by yahoo emerged in 2006 creator is "Doug cutting" it is now hosted by apache architecture



To communicate between node in 2nd generation uses replication factor Hadoop and HDFS utilize a master-slave architecture. HDFS is written in Java, with an HDFS cluster consisting of a primary Name Node a master server that manages the file system namespace and also regulates access to data by clients . An optional secondary Name Node for fail over purposes also may be configured. Consecutively. HDFS has many goals. Here are some of the most notable: The Fault tolerance by detecting faults and applying quick automatic recovery.

- Accessing data via Map Reduce streaming The Processing logic is close to the data rather than the data close to the processing Logic.

VI. COMPARISION

Big data is the data which becomes large enough that it is difficult or impossible to processed by using conventional methods. So creators of web search engines were among

the first to confront this problem. Now a days social networks , sensors , mobile phones and science contribute to petabytes of data created daily and to meet the challenge of processing such large data sets the google created Map-Reduce. Google's work and Yahoo's creation of the Hadoop MapReduce implementation has spawned an ecosystem of big data processing tools. The MapReduce has grown in popularity as stack for big data systems has emerged in comprising layers of Storage of Map Reduce and Query (SMAQ).The SMAQ systems are typically open source and run on commodity Hardware and strategy for distributing an extremely large problem across an extremely computing clusters. In map stage an task is divided into a no of identical sub tasks which are distributed across many processors. The intermediate result then combined by single reduced task Hadoop is designed to run on a large number of machines that don't share any memory or disks. That means we can buy a whole bunch of commodity servers, put them in a rack and run the Hadoop software on each one of them. When we want to load all of your organization's data into Hadoop what the software bust that data into pieces that it then spreads across your different servers. There's no one place where you go to talk to all of your data. Hadoop keeps track of where the data resides and because there are multiple copy stores. The data stored on a server that goes offline or even dies can be automatically replicated from a known copy. In a centralized database system we have got one big disk connected to four or eight or 16 big processors. In a Hadoop cluster every one of those servers has two or four or eight CPUs. You can run your indexing job by sending your code to each of the dozens of servers in your cluster, and each server operates on its own little piece of the data. The Results are then delivered back to you in a unified whole.

REFERENCES

[1] Apache HBase http://hbase.apache.org

[2] Apache Accumulo http://accumulo.apache.org

[3] J. Kepner and S. Ahalt, "MatlabMPI," Journal of Parallel and Distributed Computing, vol. 64, issue 8, August, 2004.

[4] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A.D. Joseph, R.Katz, S. Shenker and I. Stoica, "Mesos: A Platform for Fine-Grained.

[5] N. Bliss, R. Bond, H. Kim, A. Reuther, and J.Kepner, "Interactive grid computing at Lincoln Laboratory," Lincoln Laboratory Journal, vol. 16,no. 1, 2006.

[6] J. Kepneretal.,"Dynamic distributed dimensional data model (D4M) database and computation system,"37th IEEE International 1989.