Overview of Big Data & Study of Processing Techniques for Big Data

Ovass Shafi
Assistant Professor,
Deptt.of Computer Applications,
Amar Singh College,
Googji Bagh, Srinagar,
J&K,

Ab. Qayoom Sofi
Assistant Professor
Deptt.of Computer Applications,
Government Degree College,
Pattan, J&K,

Rashid Ashraf Malik
Assistant Professor,
Deptt.of Computer Applications,
Amar Singh College,
Googji Bagh, Srinagar,
J&K,

Asif Iqbal Kawoosa
Assistant Professor,
Deptt.of Computer Applications,
Amar Singh College,
Googji Bagh, Srinagar,
J&K,

Syed Ishfaq Manzoor
Assistant Professor,
Deptt.of Computer Applications,
Amar Singh College,
Googji Bagh, Srinagar,
J&K,

Abstract

A data set (or dataset) is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set.

Big data is a term for data sets that are so large or complex that traditional data processing application software is inadequate to deal with them. Challenges include capture, storage, analysis, data curation, search, sharing, transfer, visualization, querying, updating and information privacy. The term "big data" often refers simply to the use of predictive analytics, user behaviour analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set. "There is little doubt that the quantities of data now available are indeed large, but that’s not the most relevant characteristic of this new data ecosystem. Analysis of data sets can find new correlations to "spot business trends, prevent diseases, combat crime and so on. Data sets grow rapidly because they are increasingly gathered by cheap and numerous information-sensing Internet of things devices such as mobile devices, aerial (remote sensing), software logs, cameras, microphones, radio-frequency identification (RFID) readers and wireless sensor networks. The world's technological per-capita capacity to store information has roughly doubled every 40 months since the 1980s. As of 2012, every day 2.5 exabytes (2.5×1018) of data are generated. One question for large enterprises is determining who should own big-data initiatives that affect the entire organization.

Relational database management systems and desktop statistics and visualization-packages often have difficulty handling big data. The work may require "massively parallel software running on tens, hundreds, or even thousands of servers". What counts as "big data" varies depending on the capabilities of the users and their tools, and expanding capabilities make big data a moving target. "For some organizations, facing hundreds of gigabytes of data for the first time may trigger a need to reconsider data management options. For others, it may take tens or hundreds of terabytes before data size becomes a significant consideration."

Keywords: Data Set, Big Data, Data Analysis, MapReduce, Hadoop, Apache Storm, Apache Flink, Apache Spark, Big Data Sources.

Introduction

A data set (or dataset) is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set.

Traditional data processing applications are not applicable for processing of large datasets. In modern days, the data source from which data generates are unlimited in number, and the amount of data generated from these sources is very huge. It is not possible to use the traditional data processing applications to process such enormous quantity of data.

Aim of the Study

The aim of the study is to provide knowledge about Big Data, Big Data Sets, and Sources of Big Data. This paper studies the various methods used for processing of Big Data along with their strengths and advantages. This paper also compares various Big Data Processing methods with each other and provides information regarding their merits and demerits.

Big Data

Big data is a term for data sets that are so large or complex that traditional data processing application software is inadequate to deal with them. Challenges include capture, storage, analysis, data curation, search, sharing, transfer, visualization, querying, updating and information privacy. The term "big data" often refers simply to the use of predictive analytics, user behaviour analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set. "There is little doubt that the quantities of data now available are indeed large, but that's not the most relevant characteristic of this new data ecosystem. Analysis of data sets can find new correlations to "spot business trends, prevent diseases, combat crime and so on. Data sets grow rapidly because they are increasingly gathered by cheap and numerous information-sensing Internet of things devices such as mobile devices, aerial (remote sensing), software logs, cameras, microphones, radio-frequency identification (RFID) readers and wireless sensor networks. The world's technological per-capita capacity to store information has roughly doubled every 40 months since the 1980s. As of 2012, every day 2.5 exabytes (2.5×1018) of data are generated. One question for large enterprises is determining who should own big-data initiatives that affect the entire organization

Relational database management systems and desktop statistics and visualization-packages often have difficulty handling big data. The work may require "massively parallel software running on tens, hundreds, or even thousands of servers". What counts as "big data" varies depending on the capabilities of the users and their tools, and expanding capabilities make big data a moving target. "For some organizations, facing hundreds of gigabytes of data for the first time may trigger a need to reconsider data management options. For others, it may take tens or hundreds of terabytes before data size becomes a significant consideration."

Source of Big Data

The sources that generate big datasets include:

1. Sensor Data
2. Machine Log Data
3. Public web
4. Social Media
5. Business Apps
6. Media
7. Docs
8. Archives

Archived data is the internal data behind the firewalls which is unstructured without any APIs. The data can be digitized in order to structure it.

Docs include both the data that resides internal and external to the organization. Such data is also without any APIs.

Media Includes data in the form of text, images, audio or video forms. The data may exist in-and-out of the organization, connected with APIs and is structured to some extent.

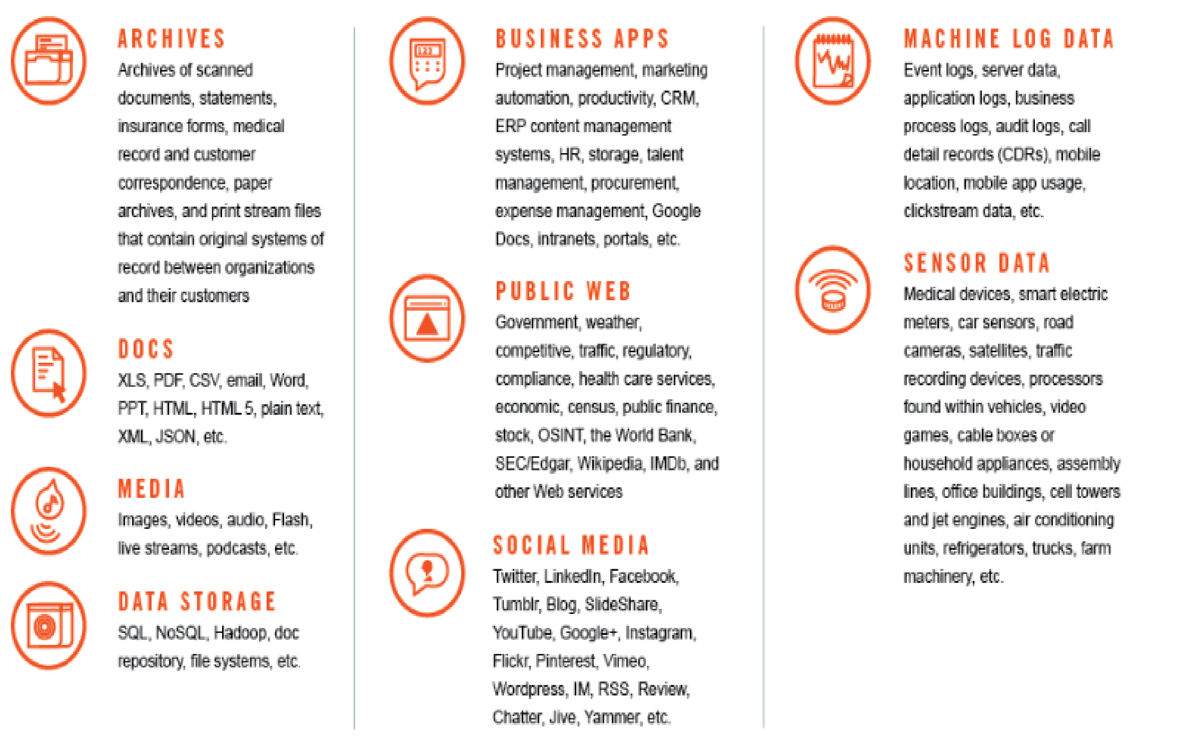
Business apps includes data collected by commercial organizations for using APIs and is mostly structured and can be collected both internal as well as outside of the organization, for example collection of customer details by ecommerce sites for efficient response to their customer in future.

Public web includes data that is mostly external to organization, For example, effect of introduction of new products by competitors on your business.

Social media generates high velocity, high volume data that is used to detect trends, analyze sentiment about your brand, customer service and competitors etc.

Machine log data is used through Web analytics, where the data regarding activities done by users on web are recorded, analysed and to better identify, target and convert visitors.

Sensor data is high velocity, high volume, and variety data that can be used correctly to understand user context and predict behaviour. Sensors for geolocation, temperature, noise, attention, engagement, biometrics, and more can collect reams of data that is useful for better purchase and ownership experiences in a variety of industries.



ARCHIVES
Archives of scanned documents, statements, insurance forms, medical record and customer correspondence, paper archives, and print stream files that contain original systems of record between organizations and their customers

DOCS
XLS, PDF, CSV, email, Word, PPT, HTML, HTML 5, plain text, XML, JSON, etc.

MEDIA
Images, videos, audio, Flash, live streams, podcasts, etc.

DATA STORAGE
SQL, NoSQL, Hadoop, doc repository, file systems, etc.

BUSINESS APPS
Project management, marketing automation, productivity, CRM, ERP content management systems, HR, storage, talent management, procurement, expense management, Google Docs, intranets, portals, etc.

PUBLIC WEB
Government, weather, competitive, traffic, regulatory, compliance, health care services, economic, census, public finance, stock, OSINT, the World Bank, SEC/Edgar, Wikipedia, IMDb, and other Web services

SOCIAL MEDIA
Twitter, LinkedIn, Facebook, Tumblr, Blog, SlideShare, YouTube, Google+, Instagram, Flickr, Pinterest, Vimeo, Wordpress, IM, RSS, Review, Chatter, Jive, Yammer, etc.

MACHINE LOG DATA
Event logs, server data, application logs, business process logs, audit logs, call detail records (CDRs), mobile location, mobile app usage, clickstream data, etc.

SENSOR DATA
Medical devices, smart electric meters, car sensors, road cameras, satellites, traffic recording devices, processors found within vehicles, video games, cable boxes or household appliances, assembly lines, office buildings, cell towers and jet engines, air conditioning units, refrigerators, trucks, farm machinery, etc.

What Kinds of Datasets are Considered Big Data?

The big data as the name implies is large in volume. In the same way, the uses of big data are also varied. The various sources of big data are listed above, but the most prominent sources of big data are transactional data, including everything from stock prices to bank data to individual merchants purchase histories; and sensor data, much of it coming from what is commonly referred to as the Internet of Things (IoT). This sensor data might be anything from measurements taken from robots on the manufacturing line of an auto maker, to location data on a cell phone network, to instantaneous electrical usage in homes and businesses, to passenger boarding information taken on a transit system.

By properly processing and analysing this data (data mining), the organizations are able to deduct various patterns and trends from the data and provide more customized service and increased efficiencies in whatever industry the data is collected from. For example the customers will be provided by the choices they are interested in by the system based on their previous surfing experiences.

Processing of Big Data

The data generated by various technologies like WWW, scientific applications, engineering applications, networks, business enterprise services, cloud computing, distributed computing etc., are so large in volume that it is not possible to process them using traditional data processing techniques, and lead to the development of data sciences or data mining techniques.

Various efforts have been made to bring the distributed technologies and standard users closer by hiding the technical differences present in distributing environment. Not only the complex designs are required to create and maintain big data processing techniques but big data platforms also require additional algorithms that give rise to additional tasks like big data pre-processing and analysis.

MapReuce & Apache Hadoop

The first technique for processing of large-scale datasets was MapReduce. This tool was designed to process and generate huge datasets automatically and in distributed way. The MapReduce technique actually combines two operations, Map and Reduce which enables the user to use a scalable and distributed tool without worrying about technical difference, such as failure recovery, data partitioning and communication. Apache Hadoop is the most popular open-source implementation of MapReduce that maintains all the above features. The drawback of MapReduce and Hadoop is that it lacks to scale well when dealing with iterative and online processes, typical in machine learning and stream analytics.

Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

Typically the compute nodes and the storage nodes are the same, that is, the MapReduce framework and the Hadoop Distributed File System are running on the same set of nodes. This configuration allows the framework to effectively schedule tasks on the nodes where data is already present, resulting in very high aggregate bandwidth across the cluster.

The MapReduce framework consists of a single master Resource Manager, one slave NodeManager per cluster-node, and MRAppMaster per application. Minimally, applications specify the input/output locations and supply map and reduce functions via implementations of appropriate interfaces. These, and other job parameters, comprise the job configuration.

The Hadoop job client then submits the job (jar/executable etc.) and configuration to the Resource Manager which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

Although the Hadoop framework is implemented in Java, MapReduce applications need not be written in Java.

1. Hadoop Streaming is a utility which allows users to create and run jobs with any executables (e.g. shell utilities) as the mapper and/or the reducer.
2. Hadoop Pipes is a SWIG-compatible C++ API to implement MapReduce applications.

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 and produces a set of <key, value> pairs as the output of the job, conceivably of different 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.

Apache Spark

Another tool for processing of big data as an alternative to Hadoop is Spark from Apache that provides the capability of performing faster distributed computing by using in-memory primitives. It has the feature to load data into the memory and re-using it repeatedly and thus overcomes the problem of iterative and online processing presented by MapReduce. Besides Spark is a general purpose Framework and allows to implement several distributed programming models on top of itlike Pregel or Hadoop. Apache Spark is built on top of a new abstraction model known as Resilient Distributed Datasets(RDDs). The Spark allows controlling the persistence and managing the partitioning of data along with other features.

Spark was introduced by Apache Software Foundation for speeding up the Hadoop computational computing software process. Spark is not a modified version of Hadoop and is not, really, dependent on Hadoop because it has its own cluster management. Hadoop is just one of the ways to implement Spark. Spark uses Hadoop in two ways – one is storage and second is processing. Since Spark has its own cluster management computation, it uses Hadoop for storage purpose only.

Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. It is based on Hadoop MapReduce and it extends the MapReduce model to efficiently use it for more types of computations, which includes interactive queries and stream processing. The main feature of Spark is its in-memory cluster computing that increases the processing speed of an application.

Spark is designed to cover a wide range of workloads such as batch applications, iterative algorithms, interactive queries and streaming. Apart from supporting all these workload in a respective system, it reduces the management burden of maintaining separate tools.

Features of Apache Spark

Apache Spark has following features.

1. **Speed**

   Spark helps to run an application in Hadoop cluster, up to 100 times faster in memory, and 10 times faster when running on disk. This is possible by reducing number of read/write operations to disk. It stores the intermediate processing data in memory.
2. **Supports Multiple Languages**

   Spark provides built-in APIs in Java, Scala, or Python. Therefore, you can write applications in different languages. Spark comes up with 80 high-level operators for interactive querying.
3. **Advanced Analytics**

   Spark not only supports ‘Map’ and ‘reduce’. It also supports SQL queries, Streaming data, Machine learning (ML), and Graph algorithms.

Apache Storm

Apache Storm is another open source distributed real-time processing platform, capable of processing millions of records per second in a fault tolerant way.

Apache Storm is a distributed real-time big data-processing system. Storm is designed to process vast amount of data in a fault-tolerant and horizontal scalable method. It is a streaming data framework that has the capability of highest ingestion rates. Though Storm is stateless, it manages distributed environment and cluster state via Apache ZooKeeper. It is simple and you can execute all kinds of manipulations on real-time data in parallel.

Apache Storm is continuing to be a leader in real-time data analytics. Storm is easy to setup, operate and it guarantees that every message will be processed through the topology at least once.

Apache Storm vs Hadoop

Basically Hadoop and Storm frameworks are used for analyzing big data. Both of them complement each other and differ in some aspects. Apache Storm does all the operations except persistency, while Hadoop is good at everything but lags in real-time computation. The following table compares the attributes of Storm and Hadoop.

Storm	Hadoop
Real-time stream processing	Batch processing
Stateless	Stateful
Master/Slave architecture with ZooKeeper based coordination. The master node is called as nimbus and slaves are supervisors.	Master-slave architecture with/without ZooKeeper based coordination. Master node is job tracker and slave node is task tracker.
A Storm streaming process can access tens of thousands messages per second on cluster.	Hadoop Distributed File System (HDFS) uses MapReduce framework to process vast amount of data that takes minutes or hours.
Storm topology runs until shutdown by the user or an unexpected unrecoverable failure.	MapReduce jobs are executed in a sequential order and completed eventually.
If nimbus / supervisor dies, restarting makes it continue from where it stopped, hence nothing gets affected.	If the JobTracker dies, all the running jobs are lost.

Benefits of Apache Storm

The Apache Storm provides following benefits:

1. Storm is open source, robust, and user friendly. It could be utilized in small companies as well as large corporations.
2. Storm is fault tolerant, flexible, reliable, and supports any programming language.
3. Allows real-time stream processing.
4. Storm is unbelievably fast because it has enormous power of processing the data.
5. Storm can keep up the performance even under increasing load by adding resources linearly. It is highly scalable.
6. Storm performs data refresh and end-to-end delivery response in seconds or minutes depends upon the problem. It has very low latency.
7. Storm has operational intelligence.
8. Storm provides guaranteed data processing even if any of the connected nodes in the cluster die or messages are lost.

Apache Flink

Flink is a recent top-level Apache project designed for distributed stream and batch data processing that try to fill the online gap left by spark.

Apache Flink is an Apache project for Big Data processing. Although it looks like Apache Spark, there are a lot of differences in both their architecture and ideas. The defining hallmark of Apache Flink is the ability to process streaming data in real time. Apache Spark is considered to be the pioneer in real-time processing with proven capabilities, but its micro-batching architecture supports a Near Real Time (NRT) scenario.

The primitive concept of Apache Flink is the high-throughput and low-latency stream processing framework which also supports batch processing. The architecture is a flip of the other Big Data processing architectures where the primary notion was the batch processing framework.. The prospect of Apache Flink seems to be significant and looks like the goal for stream processing.

Conclusion

The paper starts with the introduction of Data set and defines what the data set is in terms of database management system. The discussion then moves over to big data and explains the concept of big data in detail. The next section deals with the sources of big data that includes the details regarding sources that produces large volumes of data. The research paper also throws light on what kinds of data sets can be considered big. In modern days number of datasets that produces data are large, but all datasets doesn’t produce so much volume of data to be considered them big. Finally the processing techniques of big data are introduces along with detailed concept and with their merits and demerits. The comparison among various techniques is also provided in the research paper.

References

1. *https://datafloq.com/read/understanding-sources-big-data-infographic/338*
2. *https://en.wikipedia.org/wiki/Big_data*
3. *http://www.hadoopadmin.co.in/sources-of-bigdata/*
4. *https://hadoop.apache.org/docs/r2.7.2/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html*
5. *https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.htm*
6. *https://www.tutorialspoint.com/apache_storm/apache_storm_introduction.htm*
7. *https://dzone.com/articles/apache-flink-the-4g-of-big-data*