Big Data Analytics Applications

IDA, Statistics and Data Mining, Linköping University

Ashraf Sarhan (ashsa762)

21 March 2017

Abstract

Big data analytics is the process of examining large data sets to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information. The analytical findings can lead to more effective marketing, new revenue opportunities, better customer service, improved operational efficiency, competitive advantages over rival organizations and other business benefits. Through this report, we will explore the most important applications in today's big data Analytics worlds.

Introduction

The primary goal of big data analytics is to help companies make more informed business decisions by enabling data scientists, predictive modelers and other analytics professionals to analyze large volumes of transaction data, as well as other forms of data that may be untapped by conventional business intelligence (BI) programs. That could include Web server logs and Internet clickstream data, social media content and social network activity reports, text from customer emails and survey responses, mobile-phone call detail records and machine data captured by sensors connected to the Internet of Things (IoT).

However, semi-structured and unstructured data may not fit well in traditional data warehouses based on relational databases. Furthermore, data warehouses may not be able to handle the processing demands posed by sets of big data that need to be updated frequently or even continually. For example, real-time data on the performance of mobile applications or high frequency trading (HFT). As a result, many organizations looking to collect, process and analyze big data have turned to a newer class of applications that includes Hadoop and related tools such as YARN, MapReduce, Spark, Hive and Pig as well as NoSQL databases. Those applications form the core of an open source software framework that supports the processing of large and diverse data sets across clustered systems.

Hadoop clusters and NoSQL systems are being used as landing pads and staging areas for data before it gets loaded into a data warehouse for analysis, often in a summarized form that is more conducive to relational structures. Increasingly though, big data vendors are pushing the concept of a Hadoop data lake[1] that serves as the central repository for an organization's incoming streams of raw data. In such architectures, subsets of the data can then be filtered for analysis in data warehouses and analytical databases (OLAP), or it can be analyzed directly in Hadoop using batch query tools, stream processing software and SQL on Hadoop technologies (Hive) that run interactive, ad hoc queries written in SQL.

Hadoop

Hadoop is the ground worker for most the big data analytics applications, which makes it possible to run applications on systems with thousands of commodity hardware nodes, and to handle thousands of terabytes of data. Its distributed file system (HDFS) facilitates rapid data transfer rates among nodes and allows the system to continue operating in case of a node failure. This approach lowers the risk of catastrophic system failure and unexpected data loss, even if a significant number of nodes become inoperative. Consequently, Hadoop quickly emerged as a foundation for big data processing tasks, such as scientific analytics, business and sales planning, and processing enormous volumes of sensor data, including from internet of things sensors.

[1] Data lake https://en.wikipedia.org/wiki/Data_lake

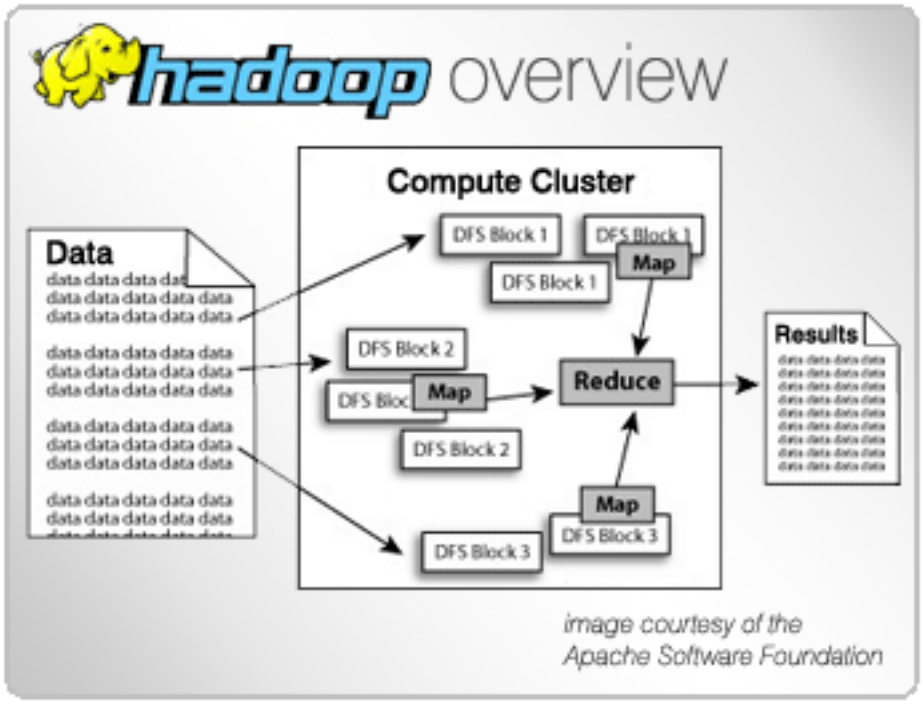

Figure 1: Hadoop (How it works)

As a software framework, Hadoop is composed of numerous functional modules. At a minimum, Hadoop uses Hadoop Common as a kernel to provide the framework's essential libraries. Other components include[2]:

- Hadoop Distributed File System (HDFS), which is capable of storing data across thousands of commodity servers to achieve high bandwidth between nodes.
- Hadoop Yet Another Resource Negotiator (YARN), which provides resource management and scheduling for user applications.
- Hadoop MapReduce, which provides the programming model used to tackle large distributed data processing, mapping data and reducing it to a result.

Some of the reasons organizations use Hadoop is its' ability to store, manage and analyze vast amounts of structured and unstructured data quickly, reliably, flexibly and at low-cost.

- *Scalability and Performance* – distributed processing of data local to each node in a cluster enables Hadoop to store, manage, process and analyze data at petabyte scale.
- *Reliability* – large computing clusters are prone to failure of individual nodes in the cluster. Hadoop is fundamentally resilient – when a node fails processing is re-directed to the remaining nodes in the cluster and data is automatically re-replicated in preparation for future node failures.
- *Flexibility* – unlike traditional relational database management systems, you don't have to created structured schemas before storing data. You can store data in any format, including semi-structured or unstructured formats, and then parse and apply schema to the data when read.
- *Low Cost* – unlike proprietary software, Hadoop is open source and runs on low-cost commodity hardware.
- *More Agility* – For unstructured data, relational databases lack the agility and scalability that is needed. Apache Hadoop makes it possible to cheaply process and analyze huge amounts of both structured and unstructured data together, and to process data without defining all structure ahead of time.

[2]Apache Hadoop http://hadoop.apache.org/

Conslusion

The Hadoop software stack introduces entirely new economics for storing and processing data at scale. It allows organizations unparalleled flexibility in how they're able to leverage data of all shapes and sizes to uncover insights about their business. Users can now deploy the complete hardware and software stack including the OS and Hadoop software across the entire cluster and manage the full cluster through a single management interface.

Apache Hadoop includes a Distributed File System (HDFS), which breaks up input data and stores data on the compute nodes. This makes it possible for data to be processed in parallel using all of the machines in the cluster. The Apache Hadoop Distributed File System is written in Java and runs on different operating systems.

Hadoop was designed from the beginning to accommodate multiple file system implementations and there are a number available. HDFS and the Amazon S3 file system are probably the most widely used, but many others are available, including the MapR File System.