Overview of Big Data Analytics

Big Data Analysis

Understanding Big Data

The growing field of *big data* is among the most significant technology trends that are fundamentally transforming the way organizations operate and compete in the business world. As more and more companies collect large amounts of data through their daily operations, the ability to analyze and glean knowledge from big data has become an integral part of a successful business.

There are inherent complexities when dealing with data. At times, the data you are working with is stored and structured in relational databases. However, there is an increasing need to process unstructured data captured from documents, customer service records, social media sites, videos, and even machine-generated data from sensors. Organizations want to transform their massive amounts of data into knowledge so that they can enhance their customer experience and attain a distinctive competitive advantage.

Given the need to process an ever-growing volume of data, coupled with the increased velocity and variety of data, new ways of data management need to be considered. Oftentimes, data management is regarded from the "software" lens, yet it's better understood from a more holistic viewpoint. Data management must factor in technological advances in cloud computing, networking, storage, hardware, and virtualization.

Big Data Management

Big data management is the capacity and ability to handle massive amounts of disparate data in an efficient and timely manner so as to facilitate 'real-time' analysis and action. To equip organizations with the ability to analyze large amounts of data in real-time, companies must build out a successful big data management architecture.

A big data management architecture includes a set of foundational blocks that enables an organization to nimbly and effectively utilize a variety of data sources. The constituent components or services of the big data management architecture are:

1. A *data ingestion service* that loads data into big data repositories from a multitude of data sources
2. A *data orchestration service* that helps prepare the data to transform the data into an analysis-ready format

3. A *data discovery service* that enables the ability to access the data available for use
4. A *data access service* that facilitates access to the data so insights can be drawn from it
5. A *data management componen*t, interconnected to all the other components, that provides capabilities around security, governance, lineage, and meta-data management

Integral to the big data management architecture is the "processing and persistence" engine that binds the constituent components of the architecture together. Either Hadoop or Spark can serve as the data processing and persistence engine.

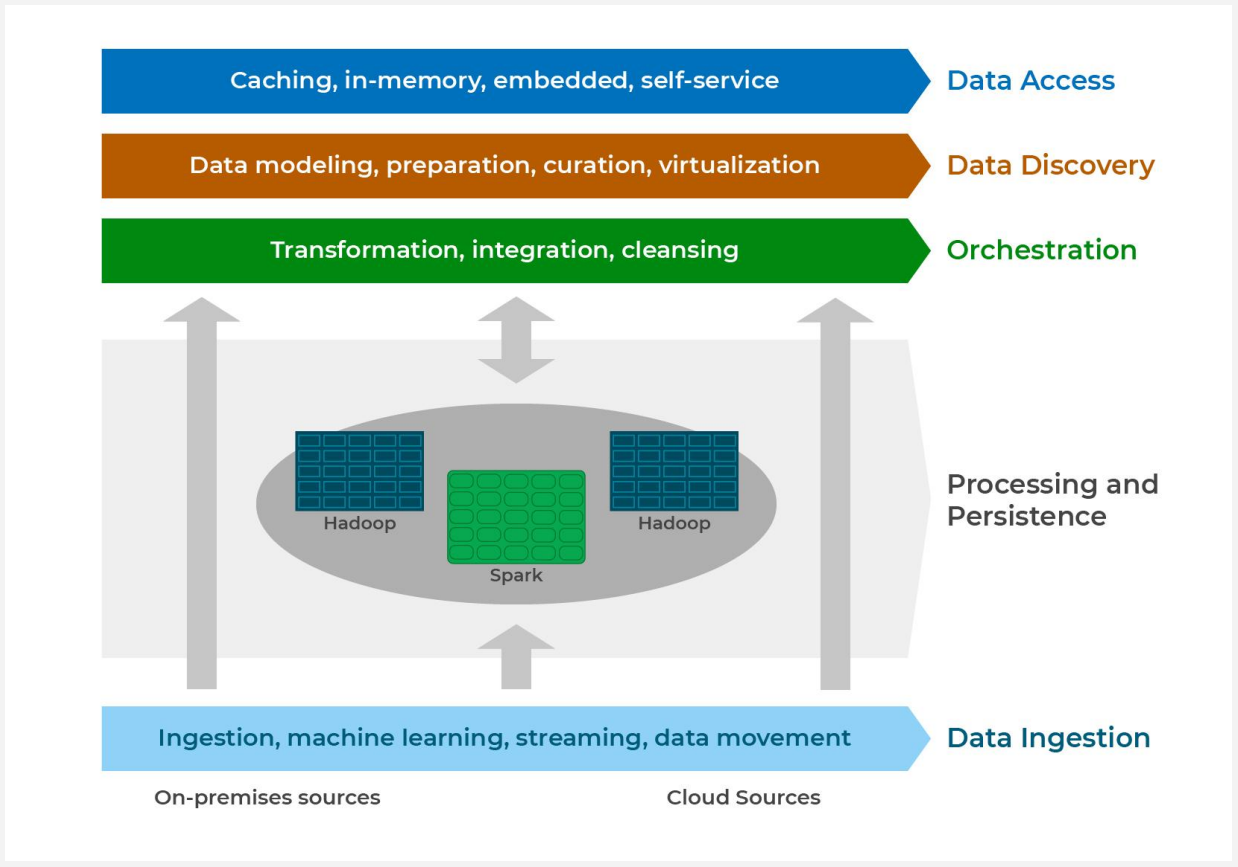

Figure 1: Big data fabric architecture

What is Hadoop?

Originally developed by Doug Cutting, a Yahoo! Engineer, *Hadoop* is now managed by the Apache Software Foundation as an open-source project. Hadoop processes large amounts of structured and unstructured data by parallelizing the processing of data across compute

nodes. Parallelization of data processing helps to speed up computation and eliminate latency. At its core, Hadoop is made up of two main components:

- **Hadoop Distributed File System (HDFS)**: A data storage cluster that enables the management of data (contained in files) across compute nodes, i.e., machines
- **MapReduce**: A *highly performant parallel* and distributed implementation of the MapReduce algorithm.

Hadoop Distributed File System

HDFS is essentially a data service that offers robust capabilities to manage data within a big data environment. Since it supports the "write once, read many times" paradigm (where once the data is written, it can be subsequently accessed/read many times), it is an excellent choice for supporting big data analysis.

The Hadoop data service includes a *NameNode* and multi *DataNode* running on a low-cost commodity hardware cluster. The NameNode orchestrates the execution of a data processing job across the cluster of data nodes, while the respective DataNodes execute on the processing tasks assigned to it by the NameNode (see Figure 2).

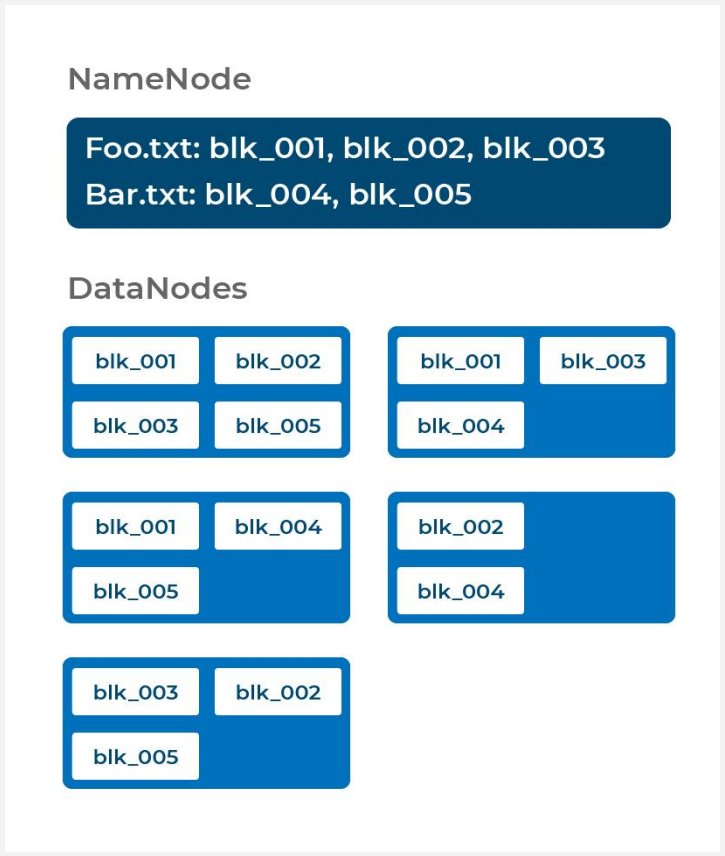

Figure 2: Illustration of NameNode and DataNode

Note that in the figure above, NameNode holds metadata for the two files (Foo.txt and Bar.txt). DataNodes holds the actual blocks. Each block is 64MB or 128MB in size and is replicated three times on the cluster.

Since the NameNode acts as the orchestrator of data processing jobs within HDFS, it monitors and tracks:

1. The way in which the files containing data are divided into data blocks
2. The data nodes which store those blocks
3. The overall health of the distributed file system

Given its role, the function of the NameNode is memory and I/O (input/output) intensive, and as such, the computer node hosting the NameNode does not store any user data or perform any computations.

The DataNodes act as soldiers performing the actual work within HDFS. When a file is read or written to the distributed file system, the file is divided into data blocks and the NameNode informs the client application (reading/writing the data) which DataNode each block resides in. The client application then communicates directly with the DataNode to process the data block assigned to that specific DataNode.

MapReduce

MapReduce is a data processing model that supports easy scaling of data processing over multiple compute nodes. The entire processing model includes several phases where each phase conducts an important series of operations to facilitate analysis on top of big data. The process is initiated through a MapReduce program (written in a client application) and continues until the results are written back into HDFS.

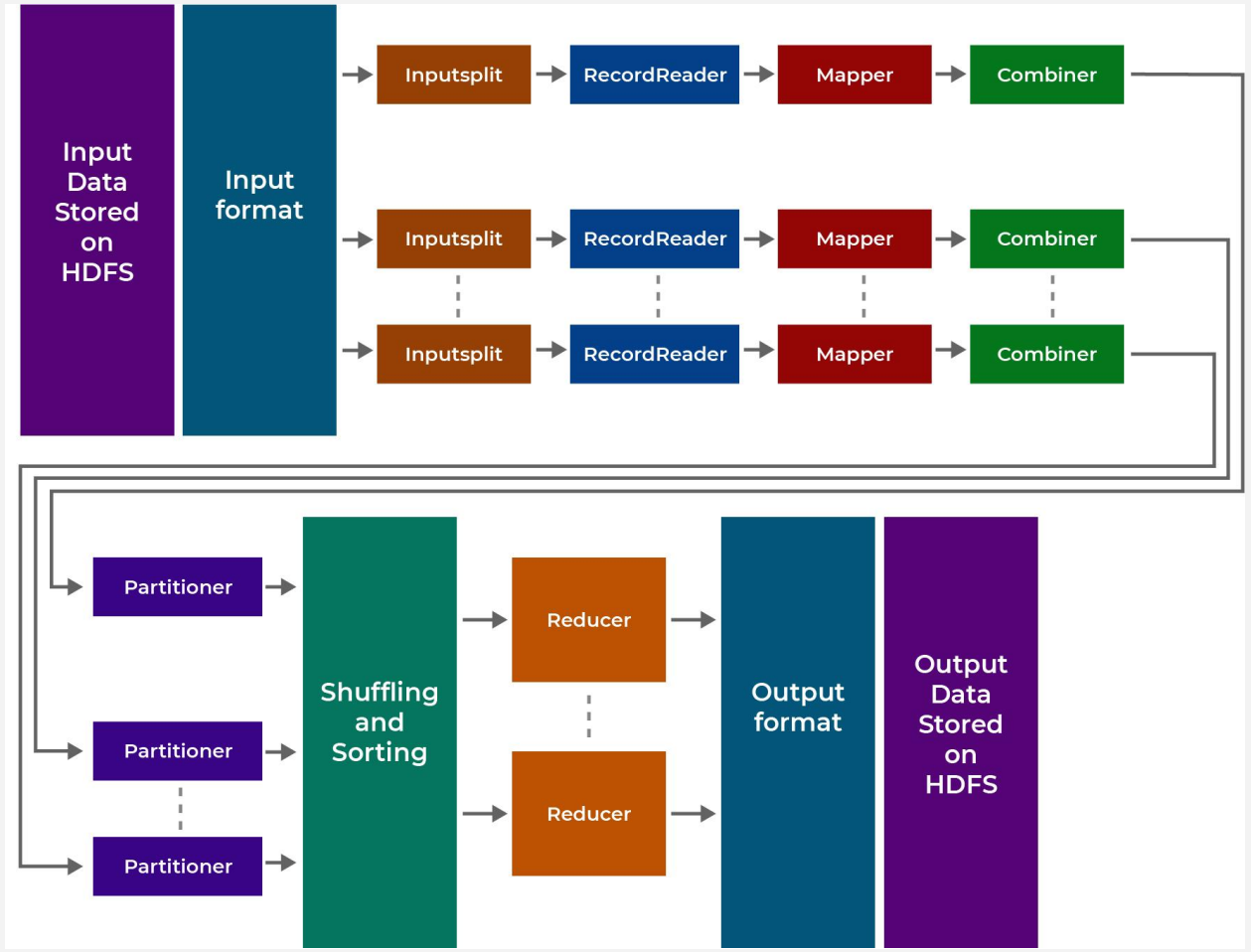

Image 3: Phases of the MapReduce model TBD

NOTE: Each of the phases depicted in Image 3 above will be covered in more detail in the next module.

What is Spark?

Apache Spark is the natural heir to MapReduce for general-purpose data processing. As Cloudera (2010-2021) so succinctly puts it, "Like MapReduce applications, each Spark application is a self-contained computation that runs user-specified code to compute a result" (p. 8).

A Spark application encompasses runtime entities such as driver, executor, job, task, and stage. Once it begins to run, a Spark application maps to a single driver process and a series of executor processes distributed across worker nodes in a cluster (see Figure 4).

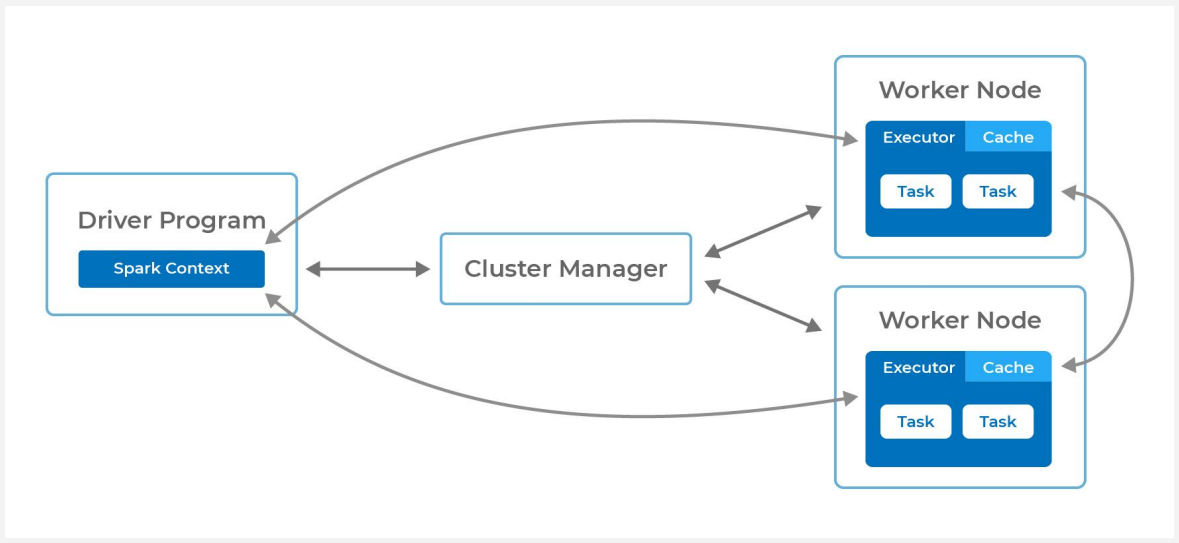

Figure 4: Spark flow

The driver process orchestrates the job flow, schedules tasks, and is available throughout the time the application is running. The executors are responsible for executing work in the form of tasks. An executor can execute many tasks which run concurrently during the lifetime of the application.

Spark is designed to cover a gamut of workloads including the following: batch applications, iterative ML algorithms, interactive SQL queries, and streaming. By supporting this wide range of disparate workloads in the same engine, Spark makes it simple to combine multiple distinct processing types as mentioned above. Moreover, Spark is highly scalable and affords APIs in Python, Scala, SQL, Java, and a host of built-in libraries.

Analytics and Big Data

Companies like Google, Facebook, Amazon, and Netflix (to name a few) are masters at analyzing big data. The ability to analyze big data provides unique opportunities for almost every organization that can do so. There are three different kinds of analysis you can do with big data (see Table 1).

Type of Analysis	Description of Analysis
Basic Analysis	Slicing and dicing of data, reporting, and visualizations
Advanced Analysis	Analysis based on statistical/predictive modeling techniques and text analytics
Operational Analysis	Analysis of business processes

Table 1: Types of analysis

Basic Analysis

Basic Analysis is usually leveraged to explore data and encompasses building out simple visualization or dashboards to gain relevant insights from the data. It is often used to examine large amounts of disparate data.

Advanced Analysis

Advanced Analysis entails utilizing algorithms for deep analysis of structured, semi-structured, or unstructured data. The analysis is based on sophisticated statistical or predictive models and advanced data-mining techniques. Nowadays, advanced analytics is becoming standard practice within organizations. Given the improvements in computational power, new algorithm design and developments, and the desire to obtain rich insights from large quantities of data, organizations are increasingly utilizing advanced analytics in their decision-making process to obtain a sustainable edge against the competition.

Predictive modeling is one of the predominant advanced analytical methods. This statistical modeling technique can help predict future outcomes on both structured and unstructured data. For example, banks might use predictive models to speculate on which customers are at risk of churn.

Unstructured data is an integral component of big data. Therefore, the process of analyzing unstructured text, transforming it into a structured format, and extracting relevant information from it, has become an important part of advanced analytics. *Text Analytics* is being used in a range of analyses, from predicting churn and detecting fraud, to deciphering the trending topics in social media analytics (Hurtiz, et. al, 2013).

Operational Analytics

Operational Analytics is when you weave analytics into the fabric of your business process. For example, a mortgage company may use a model to predict the creditworthiness of a mortgage loan application. If an application is submitted for a mortgage that does not adhere to certain credit/solvency standards, then the loan can be flagged for further review. In this case, the analysis automatically occurs within the context of a business process to optimize the workflow for evaluating loan applications.

Big Data Analytics Use Cases

Outlined below are a few use cases that illustrate how companies are leveraging big data to gain insights.

Customer 360 Views

Several organizations use big data to see a fuller or holistic view of a customer. These views compile data from a range of sources (both internal and external) to present pertinent customer-specific insights to different business functions that include customer service, sales, marketing, and other roles. This enables “at the right time” insights that can be leveraged to not only foster customer intimacy, but the information can also be utilized for up-sell/cross-sell opportunities to maximize customer share of wallet (SOW).

Fraud Detection

Fraud detection is a common use case of big data for credit card holders. Previously, credit card companies used rules-based algorithms to assist them in locating transactions that were potentially fraudulent. With the emergence of big data analytics and machine learning, the preventative systems for averting fraud that are in place today are much better than previous systems since they can now detect criminal activity and prevent false positives. For instance, a sophisticated fraud detection system may be able to decipher that a customer had recently purchased a cruise package and travel gear before a hotel was booked in a different state. Because of the detected purchase patterns, a predictive analytics workflow would be better able to determine that the hotel room reservation, albeit in a separate state, is less likely to be a fraudulent purchase.

Profit Optimization

Both service-based and product-centric organizations are utilizing big data analytics to charge customers optimum prices for business profit. The main objective of profit optimization is to set prices that maximize income. If prices are either too high or too low, companies run the risk of losing customers to competition or having diminished profit margins on sales. Therefore, big data analytics allows companies to dynamically gauge which price points have yielded the best results across varying market conditions, so the optimal price point can be presented to the customer at the right time and in the right context to drive a purchase.

Recommendation Engines

In today’s online economy, recommendation engines have become commonplace. When shopping for products or services online, websites suggest similar items that you may consider purchasing. The ability to offer relevant product recommendations is predicated on the use of big data analytics to analyze historical purchase patterns and current

clickstream, and then to surface pertinent options in real time based on what the user is exploring.

Preventive Maintenance

With the increasing utilization of the Internet of Things (IoT) in industrial settings, factories are using specialized sensors to monitor their expensive machinery and then transmit operational data about the machinery over the Internet. Big data analytics solutions can then be used to analyze that data in real time to diagnose that a problem might occur. If a potential problem is detected, agents can be proactive in administering preventive maintenance to help prevent accidents or expensive shutdowns.

References

Cloudera. (2010-2021). Spark guide.

Harvey, C. (2017). Big data use cases.

Hurwitz, J., Nugent, A. Halper, F. & Kaufman, M. (2013). Chapter 12: Defining Big Data Analytics. *Big Data for Dummies.*

Karau, H., Konwinski, A., Wendell P. & Zaharia, M. (2015). Chapter 1. Introduction to Data Analysis with Spark. *Learning Spark.*