(Big) Data Engineering In Depth

From Beginner to Professional

Moustafa Alaa

Senior Data Engineer at Onfido, London, UK

The Definitive Guide to Big Data Engineering Tasks

Previous video recap!

Section: Introduction To Hadoop

Lecture Objectives

- Why do we need Hadoop?

Lecture Objectives

- Why do we need Hadoop?
- Hadoop Distributed File System (HDFS) concepts.

Lecture Objectives

- Why do we need Hadoop?
- Hadoop Distributed File System (HDFS) concepts.
- Go dive into MapReduce.

Lecture Objectives

- Why do we need Hadoop?
- Hadoop Distributed File System (HDFS) concepts.
- Go dive into MapReduce.
- Hadoop architecture and its echosystems.

Lecture Objectives

- Why do we need Hadoop?
- Hadoop Distributed File System (HDFS) concepts.
- Go dive into MapReduce.
- Hadoop architecture and its echosystems.
- How does Hadoop store, distribute, and process the data?

Introduction to Hadoop

- Apache Hadoop's MapReduce and HDFS components were inspired by Google papers on MapReduce and Google File System

---

[1] From Wikipedia https://en.wikipedia.org/wiki/Apache_Hadoop

[2] Google File System http://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf

[3] MapReduce: Simplifed Data Processing on Large Clusters https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf

Introduction to Hadoop

Is it already dead?

What is Hadoop?

- A distributed software framework to store, process, and analzing ”Large Scale of Data AKA. Big Data”

What is Hadoop?

- A distributed software framework to store, process, and analzing ”Large Scale of Data AKA. Big Data”
- It is open source!

What is Hadoop?

- A distributed software framework to store, process, and analzing ”Large Scale of Data AKA. Big Data”
- It is open source!
- It runs on commodity (standard) hardware.

What is Hadoop?

- A distributed software framework to store, process, and analzing ”Large Scale of Data AKA. Big Data”
- It is open source!
- It runs on commodity (standard) hardware.
- Hadoop architecture and its echosystems.

Hadoop Core Components

- Hadoop HDFS: Data Storage Layer (File System).

Hadoop Core Components

- Hadoop HDFS: Data Storage Layer (File System).
- Hadoop MapReduce: The processing engine (compute paradigm) in Hadoop.

Hadoop Core Components

- Hadoop HDFS: Data Storage Layer (File System).
- Hadoop MapReduce: The processing engine (compute paradigm) in Hadoop.
- Hadoop YARN: The resource manager in Hadoop.

Introduction to Hadoop

What are the alternatives?

Hadoop Core Components

- ~~Hadoop HDFS~~ S3/GFS: Data Storage Layer (File System).
- ~~Hadoop MapReduce~~ Spark/Flink: The processing engine (compute paradigm) in Hadoop.
- ~~Hadoop YARN~~ Kubernetes: The resource manager in Hadoop.

Introduction to Hadoop

Can we use these alternatives on prem?

Hadoop Echosystem

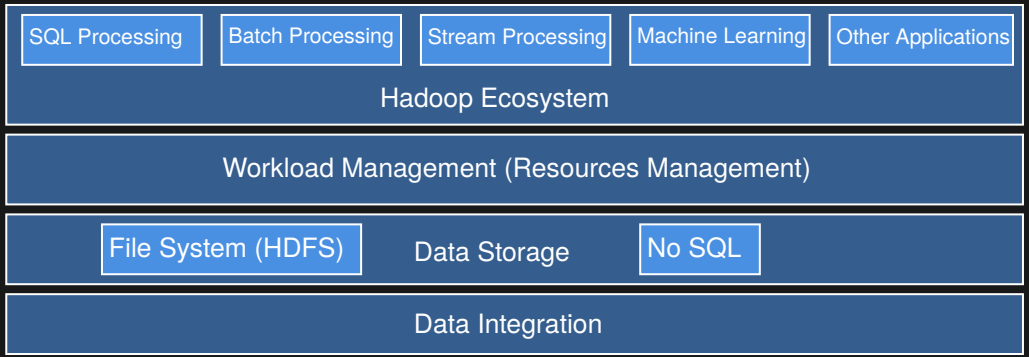

Figure: Hadoop Architecture

Hadoop Motivation

Hadoop Motivation

Hadoop Motivation

Processing:

- Traditional Computation was depending on bigger computers to deal with bigger data.

Hadoop Motivation

Processing:

- Traditional Computation was depending on bigger computers to deal with bigger data.
- This method has a bottleneck in the computation (Moore's Law), but this couldn't keep up.

Hadoop Motivation

Processing:

- Traditional Computation was depending on bigger computers to deal with bigger data.
- This method has a bottleneck in the computation (Moore's Law), but this couldn't keep up.
- The better solution requires more computers (distributed computing framework).

Hadoop Motivation

Storage:

- Traditional Computation store the data in a central unit.

Hadoop Motivation

Storage:

- Traditional Computation store the data in a central unit.
- Data was copied (moved) to the computation nodes, for example, IBM Data stage or Talend.

Hadoop Motivation

Storage:

- Traditional Computation store the data in a central unit.
- Data was copied (moved) to the computation nodes, for example, IBM Data stage or Talend.
- The process of copying or moving the data was fine when we move a small amount of data, but the big data will cause lots of problems, especially in the network bandwidth, and data moving will be costly.

Requirements for The New Approach

- Fault Tolerance.

Requirements for The New Approach

- Fault Tolerance.
- High Availability.

Requirements for The New Approach

- Fault Tolerance.
- High Availability.
- Reliability.

Requirements for The New Approach

- Fault Tolerance.
- High Availability.
- Reliability.
- Scalability.

Requirements for The New Approach

- Fault Tolerance.
- High Availability.
- Reliability.
- Scalability.
- Consistency.

Requirements for The New Approach

- Fault Tolerance.
- High Availability.
- Reliability.
- Scalability.
- Consistency.
- Data Locality.

Requirements for The New Approach

- Fault Tolerance.
- High Availability.
- Reliability.
- Scalability.
- Consistency.
- Data Locality.
- Economic.

Requirements for The New Approach

Economic

- It uses commodity (Standard/Economic) hardware.

Requirements for The New Approach

Data Locality

- It brings the program to the data rather than the data to the program. It runs the computation where the data reside.

Requirements for The New Approach

Data Locality

- It brings the program to the data rather than the data to the program. It runs the computation where the data reside.

- HDFS is strongly consistent.

Requirements for The New Approach

Fault Tolerance

- It is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components.

[1] From Wikipedia https://en.wikipedia.org/wiki/Fault_tolerance

Requirements for The New Approach

Fault Tolerance

- It is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components.

- The ability of maintaining functionality when portions of a system break down is referred to as graceful degradation.

[1] From Wikipedia https://en.wikipedia.org/wiki/Fault_tolerance

Requirements for The New Approach

Fault Tolerance

- It is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components.
- The ability of maintaining functionality when portions of a system break down is referred to as graceful degradation.
- A fault-tolerant design enables a system to continue its intended operation, possibly at a reduced level, rather than failing completely, when some part of the system fails.

[1] From Wikipedia https://en.wikipedia.org/wiki/Fault_tolerance

Requirements for The New Approach

High Availability

- High availability (HA) is a characteristic of a system which aims to ensure an agreed level of operational performance, usually uptime, for a higher than normal period.

[1] From Wikipedia https://en.wikipedia.org/wiki/High_availability

Requirements for The New Approach

High Availability

- High availability (HA) is a characteristic of a system which aims to ensure an agreed level of operational performance, usually uptime, for a higher than normal period.

- The availability of the cluster (system) to operate without any downtime despite any hardware failure. The data or the system should be available and accessed from any alternative way.

[1] From Wikipedia https://en.wikipedia.org/wiki/High_availability

Requirements for The New Approach

Reliability

- The data reliably stored on the cluster of machine despite machine failures.

Requirements for The New Approach

Scalability

- The system must be highly scalable in both vertical and horizontal. This means we can add a new node to an existing cluster easily or add new hardware to an existing node.

Requirements for The New Approach

Consistency

- Any failure during the execution job shouldn’t affect the outcome of the job.

Requirements for The New Approach

Consistency

- Any failure during the execution job shouldn’t affect the outcome of the job.
- HDFS is strongly consistent.

Hadoop Core Concepts

Hadoop Core Concepts

- Hadoop is scalable and fault-tolerant.

Hadoop Core Concepts

- Hadoop is scalable and fault-tolerant.
- Hadoop replicates the data to increase the availability and reliability.

Hadoop Core Concepts

- Hadoop is scalable and fault-tolerant.
- Hadoop replicates the data to increase the availability and reliability.
- Hadoop brings the program to the data.

Hadoop Core Concepts

- Hadoop is scalable and fault-tolerant.
- Hadoop replicates the data to increase the availability and reliability.
- Hadoop brings the program to the data.
- Applications are written in high-level code.

Hadoop Core Concepts

- Hadoop is scalable and fault-tolerant.
- Hadoop replicates the data to increase the availability and reliability.
- Hadoop brings the program to the data.
- Applications are written in high-level code.
- Hadoop reduces the data movement (shuffle) between the nodes.

Thank you for watching!

See you in the next video ☺