# Map Reduce | Meerza Ahmed

Tuesday, November 14, 2023 1:00 PM

Lecture 13: Link | Syllabus

**Class Week 13/15*

**Database Systems by Coronel & Morris**

## Chapter 14: Big Data Analytics & NoSQL

### * 14.2 *Hadoop*

Scaling out into clusters based on low-cost, commodity servers is the dominant approach that organizations are currently pursuing for Big Data management. As a result, new technologies not based on the relational model have been developed. Hadoop is a Java-based framework for distributing and processing very large data sets across clusters of computers, it has become the de facto standard for most Big Data storage and processing While the Hadoop framework includes many parts, the two most important components are the Hadoop Distributed File System (HDFS) and MapReduce.

* *Impala* was the first SQL-on-Hadoop application. It was produced by Cloudera as a query engine that supports SQL queries that pull data directly from HDFS. Prior to Impala, if an organization needed to make data from Hadoop available to analysts through an SQL interface, data would be extracted from HDFS and imported into a relational database. With Impala, analysts can write SQL queries directly against the data while it is still in HDFS. Impala makes heavy use of in-memory caching on data nodes. It is generally considered an appropriate tool for processing large amounts of data into a relatively small result set.

### * 14.2-a *HDFS*

Hadoop Distributed File System (HDFS) is a low-level distributed file processing system, this means that it is used for data storage.

(HDFS) approach to distributing data is based on several key assumptions :

- High Volume - The volume of data in Big Data applications is expected to be in tera bytes, petabytes, or larger. Hadoop assumes that files in the HDFS will be extremely large. Data in the HDFS is organized into physical blocks, just as in other file storage.
- Write - Once, read - many - Using a write-once, read-many model simplifies concurrency issues and improves overall data throughput. Using this model, a file is created, written to the file system, and then closed. Once the file is closed, changes cannot be made to its contents. This improves overall system performance and works well for the types of tasks performed by many Big Data applications
- Streaming access - Unlike transaction processing systems where queries often retrieve small pieces of data from several different tables, Big Data applications typically process entire files. Instead of optimizing the file system to randomly access individual data elements, Hadoop is optimized for batch processing of entire files as a continuous stream of data.
- Fault tolerance - Hadoop is designed to be distributed across thousands of low-cost, commodity computers. It is assumed that with thousands of such devices, at any point in time, some will experience hardware errors. Therefore, the HDFS is designed to replicate data across many different devices so that when one device fails, the data is still available from another device.

### * 14.2-b *MapReduce*

MapReduce provides data processing to complement data storage of HDFS, MapReduce is the computing framework used to process large data sets across clusters. *Conceptually,* MapReduce follows the principle of divide and conquer. MapReduce takes a complex task, breaks it down into a collection of smaller subtasks, performs the subtasks all at the same time, and then combines the result of each subtask to produce a final result for the original task.

* A map function takes a collection of data and sorts and filters the data into a set of key-value pairs, it is performed by a program called a mapper. A reduce function takes a collection of key-value pairs, all with the same key value, and summarizes them into a single result, it is performed by a program called a reducer.

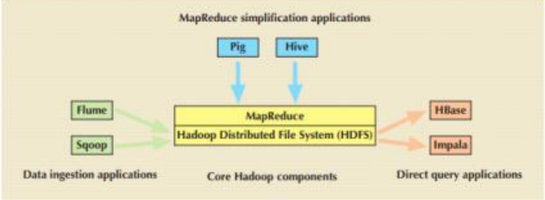
* MapReduce works like this :

1. Big data is split into file segments, held in a compute cluster made up of nodes (aka partitions)
2. A mapper task is run in parallel on all the segments (ie. in each node/partition, in each of its segments); each mapper produces output in the form of multiple (key,value) pairs
3. Key/value output pairs from all mappers are forwarded to a shuffler, which consolidates each key's values into a list (and associates it with that key)
4. The shuffler forwards keys and their value lists, to multiple reducer tasks; each reducer processes incoming key-value lists, and emits a single value for each key

Optionally, before forwarding to shufflers, a 'combiner' operation in each node can be set up to perform a local per-key reduction - if specified, this would be 'step 1.5', in the above workflow.

The cluster user (programmer) only needs to supply a mapper task and a reducer task, the rest is automatically handled!

FIGURE 14.6 A SAMPLE OF THE HADOOP ECOSYSTEM


MapReduce simplification applications
Pig
Hive
Flume
Sqoop
MapReduce
Hadoop Distributed File System (HDFS)
HBase
Impala
Data ingestion applications
Core Hadoop components
Direct query applications

*** 14.4-a** *Data Mining*

Data mining refers to analyzing massive amounts of data to uncover hidden trends, patterns, and relationships; to form computer models to simulate and explain the findings; and then to use such models to support business decision making. In other words, data mining focuses on the discovery and explanation stages of knowledge acquisition.

*How is ML different from DM?*

Machine learning is the process of TRAINING an algorithm on an EXISTING dataset in order to have it discover relationships (so as to create a model/pattern/trend), and USING the result to analyze NEW data.

* Data mining consists of four general phases :

1. Data Preparation - In this phase the main data sets to be used by the data-mining operation are identified and cleansed of any data impurities. Because the data in the data warehouse is already integrated and filtered, the data warehouse usually is the targ et set for data-mining operations

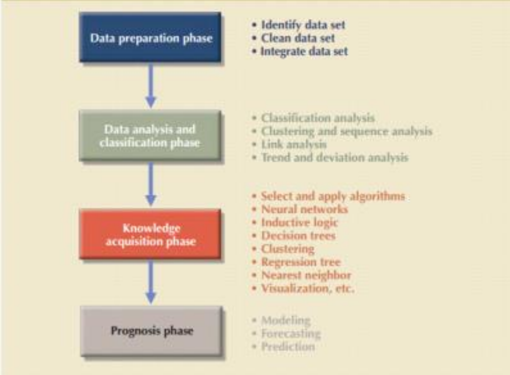
2. Data Analysis & Classification - The data analysis and classification phase studies the data to identify common data characteristics or patterns.

   During this phase, the data-mining tool applies specific algorithms to find :
   - Data groupings, classifications, clusters, or sequences
   - Data dependencies, links, or relationships
   - Data patterns, trends, and deviations

3. Knowledge Acquisition - this phase uses the results of the data analysis and classification phase. During the knowledge acquisition phase, the data-mining tool (with possible intervention by the end user) selects the appropriate modeling or knowledge acquisition algorithms

4. Prognosis - Although many data-mining tools focus on the knowledge–discovery phase, others continue to the prognosis phase. In that phase the data-mining findings are used to predict future behavior and forecast business outcomes.

FIGURE 14.13 DATA-MINING PHASES



** Data mining can be run in two modes:*

1. Guided. The end user guides the data-mining tool step by step to explore and explain known patterns or relationships. In this mode, the end user decides what techniques to apply to the data.
2. Automated. In this mode, the end user sets up the data-mining tool to run automatically and uncover hidden patterns, trends, and relationships. The data-mining tool applies multiple techniques to find significant relationships