Elmélet

Motivation

- Big Data:
    - How to store any dart as is
    - How to run any query (analytics)
- NoSQL:
    - How to store data so that some queries are fast
    - Limited ad-hoc queries
- Realtime queries:
    - Execute time: milliseconds, seconds
- Non-realtime queries:
    - Execute time: minutes, hours, days

Definition of Big Data:
There is no exact definition of Big Data.
It deals wit data sets that are too large or complex to be dealt with by traditional data-processing application software.

Big Data is not about size.

- Volume:
    - Size of data
- Velocity:
    - Speed of data creation, storage, processing, analysis and visualization
- Variety:
    - Data comes in different formats (90% of the data is unstructured)

Example: Self driving cars.
They have all kind of sensors.
They collect data about the environment.
This data needs to be processed in real time.
The data is unstructured.

It needs to be collected, stored, processed, analyzed, understanding and visualized.

Moving data to processing unit (Traditional approach) or Moving processing unit to the data (Map reduce approach)

e.g. National election:
Traditional approach:

- Collecting the data from the voting stations
- Moving the data to the algorithm
- National office counts the votes

Distribution approach:

- Collecting the data from the voting stations
- Move the algorithm to the data
- Local offices process their data (map)
- National office summarizes the data (reduce)

Hadoop

An open source framework for storing and processing big data in a distributed fashion on large clusters of commodity hardware.

- distributed file system (HDFS)
- distributed processing (MapReduce)
- very large data sets
- on computer clusters built from commodity hardware
- optimized for large files

Distributed computing:
YARN: Yet Another Resource Negotiator
A framework for job scheduling and cluster resource management.

- Resource manager:
  - Maintains a process overview of all the nodes in the cluster
  - Designates a resource manager for each application
  - Scheduling workloads
- Node manager:
  - Tracks resource usage on each node
  - Acts as a slave to the resource manager

Programming framework:
MapReduce:
A YARN-based system for parallel processing of large data sets.

- Map:
  - Breaks down a query into subqueries
  - Distributes the subqueries to the nodes
  - Each node processes its subquery

- Reduce:
    - Combines the results of the subqueries
    - Returns the result to the client

Hadoop components:

- HDFS:
    - Hadoop Distributed File System
    - Stores data in a distributed fashion
    - Replicates data across multiple nodes

HIVE:
SQL interface to Hadoop
Data warehouse software for:

- reading, writing, managing large data sets
- residing in distributed storage
- using SQL syntax

Architecture:

- Hive Metastore:
    - Stores metadata about Hive tables
    - Stores metadata about partitions
    - Stores metadata about columns
- Hive Server:
    - Receives queries from clients
    - Translates queries into MapReduce jobs
    - Sends jobs to Hadoop cluster

Features:

- Supports:
- A large subset of SQL
- Complex data types (arrays, structs, maps)
- Views
- Indexing
- Partitioning
- Bucketing
- User-defined functions
- Sampling
- Explain
- Optimization hints

- NOT a relational database

- NOT designed for OLTP
- (on-line transaction processing)
- NOT real-time

Spark

Lightning-fast unified analytics engine for large-scale data processing.

- Analytics engine for large-scale data processing
    - Distributed computing
    - Not a data store - but can integrate with them
    - Not for OLTP
- Lightning-fast:
    - Distributed
    - In-memory
    - On iterative processing
    - High-level optimizations

Easy to use

- High level operations
- Supports:
    - several programming languages (Java, Scala, Python, R, SQL)
    - Many data stores (HDFS, Hive, HBase, Cassandra, ...)
- Interactive shell