**Introduction**

Data too big (big data) to be processed on a single machine needs distributed processing. Big data has three Vs, **Volume**, **Variety**, and **Velocity**.

Hadoop is a kernel for big data. You can easily do MapReduce, or do SQL queries using pig and hive, or interactive SQL using impala.

**Hive 🡪** translates hive code to MapReduce

**Pig 🡪** Same as Hive, they both need to translate to MapReduce

**Impala 🡪** Doing queries directly on HDFS data without MapReduce, much faster than Hive or Pig.

**HDFS**

**HDFS 🡪** Hadoop distributed file system

When data is loaded to HDFS it will be divided into **blocks** of 64MB data and get a name.

Each **block** will be loaded on one node (machine), and a daemon called **data node** exist on those machines.

There is one **name node** on a single machine that has the meta data information on how blocks of data are connected. Data **redundancy** assures data is available at least on three nodes. Also two name nodes exist to keep the metadata.

**hadoop fs -put data 🡪** loads data into HDFS

**hadoop fs -any\_command 🡪** performs shell commands inside the Hadoop environment on the available data and datasets

**hadoop fs -put data folder 🡪** loads data into HDFS into a folder

**hadoop fs -get data folder 🡪** to get data from the cloud

**MapReduce**

**MapReduce 🡪** Instead of serial, data is being processed in parallel.

**Mapper 🡪** gets a part of data and performs key-value pair generation. Each mapper generates a pile of key value pairs.

**Shuffle and Sort 🡪** before data is being transferred from mapper to reducer, it is shuffled and sorted

**Reducer 🡪** reducers get their own pile from mapper and generate required results based on sorted keys.

**Combiner 🡪** A reducer right after the mapper. Sometimes, the number of data being transferred to reducer is quite large. Some reduction is therefore performed inside the mapper.

**Clusters**

**Cluster 🡪** A combination of several machines used to store and process large amount of data

**Daemon 🡪** Set of codes on each machine

**Job Tracker 🡪** It receives mapper and reducer code from the user and divide the task into mappers and reducers.

**MapReduce Patterns**

**Filtering 🡪** Simplifying the data, generating top-n list, random sampling,

**Summarization 🡪**Provides high level understanding of the data, counting records, min, max, mean, median, index, inverted index.

**Structural 🡪** Combining two data sets