Hadoop Learning Notes

A Preliminary Introduction

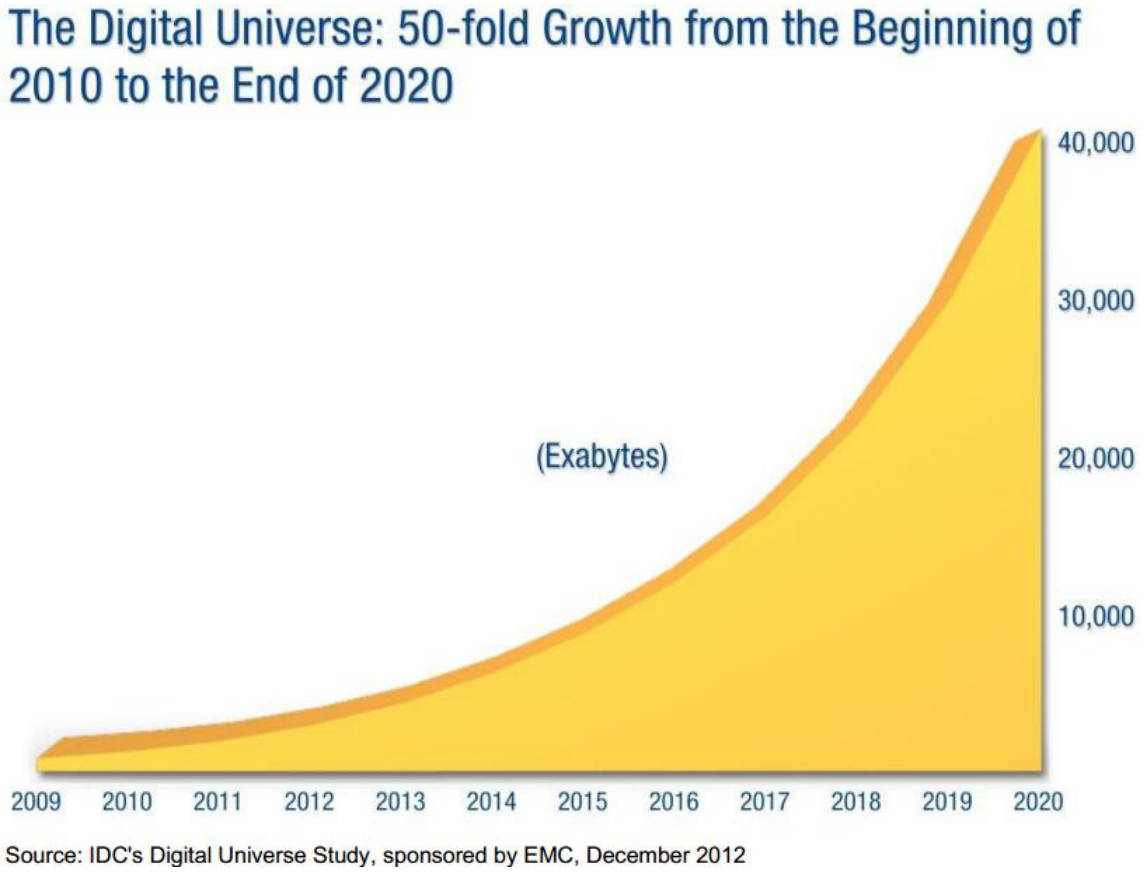
Song, Changyue
Peking University
Oct 1st, 2014
www.songcy.net

Outline

- Background
- Overview of Hadoop
- Hadoop subprojects
  - HDFS
  - MapReduce
  - Pig
  - HBase
  - Hive
- Summary

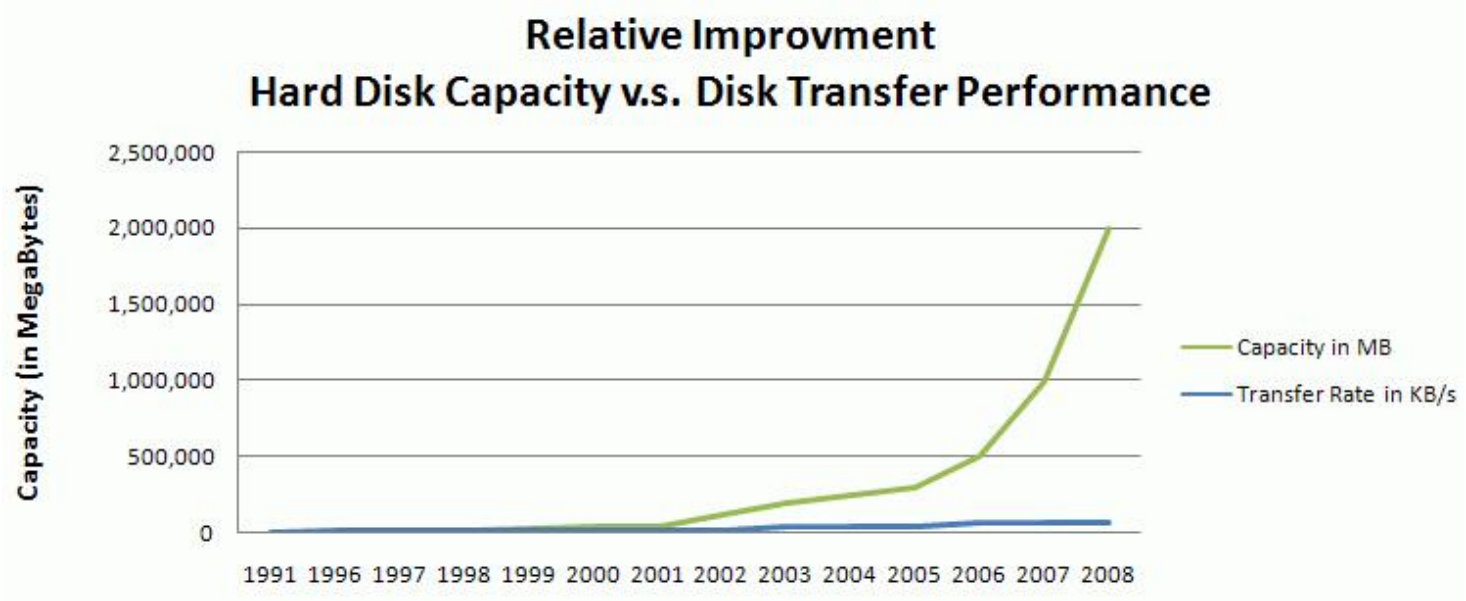
Background

- Big data!
  - Data grows explosively



Background

- Data storage and analysis
  - Storage capacity of hard drives increased massively
  - However, access speed (data reading & writing) have not kept up



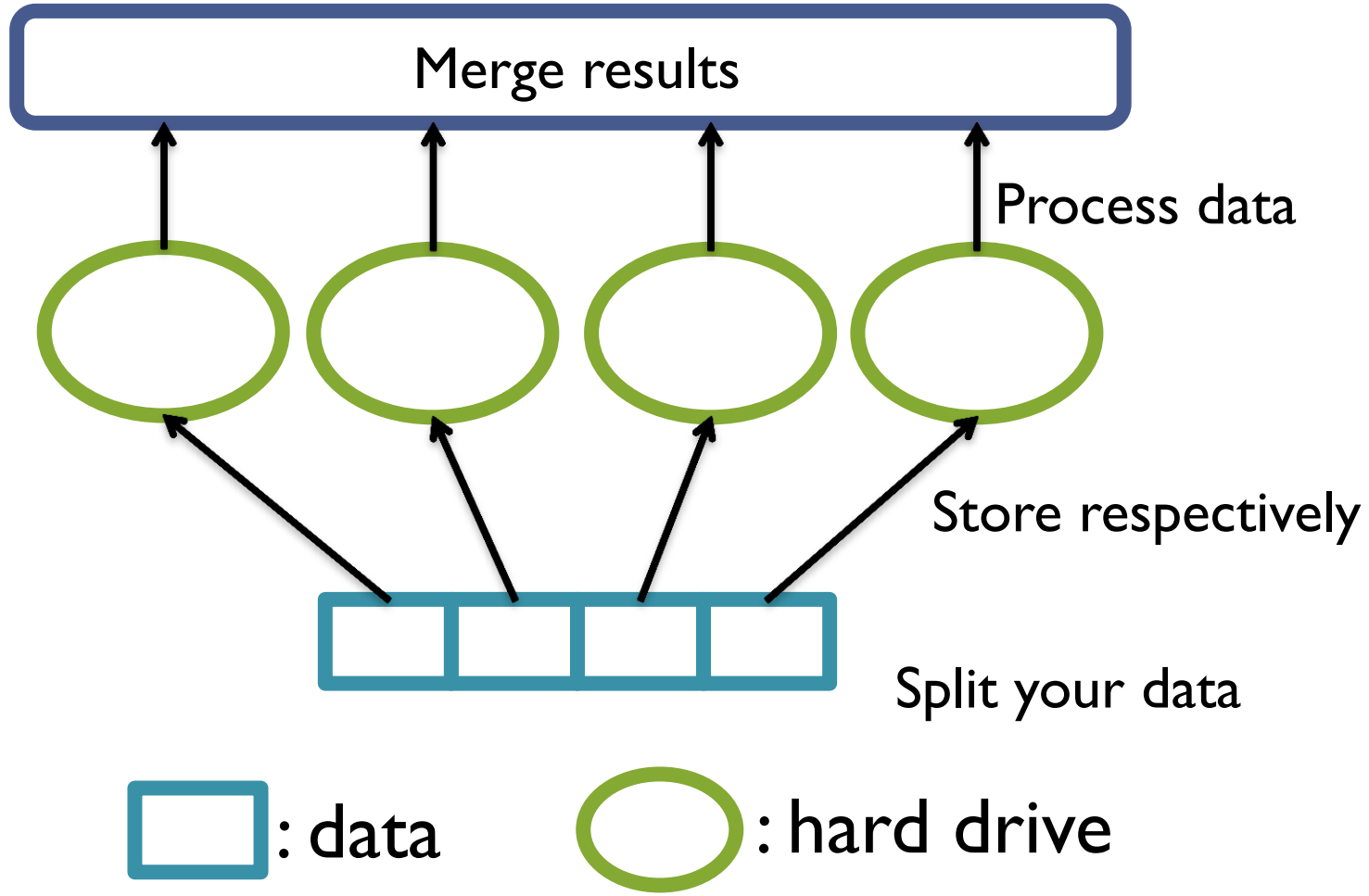
Source: http://www.cs.ucla.edu/classes/winter13/cs111/scribe/10c/

Background

- Data storage and analysis
  - Data access rate is the bottleneck!
  - If you have 1 TB of data on a hard disk, you need more than 2.5h to read all the data
- How to deal with the bottleneck?
  - Multiple hard drives & parallel working!

Background

- Multiple hard drives: a coarse model



Background

- Multiple hard drives
  - Question: If a hard drive is 1TB and your data is 1TB, isn't splitting your data and storing to 4 drives wasteful of hard drive space?
  - Answer: Imagine multiple datasets (or multiple users of the system). Each of them are split and stored in multiple drives. Then the capacity of hard drives is fully utilized, and the data access time is shortened for each dataset – as long as datasets are not analyzed at the same time!

Background

- Multiple hard drives: problems should be addressed
  - Hardware failure possibility is high: the system fails as long as one of the drives fails
  - The data from multiple drives must be combined in some way for analysis
- Hadoop is for distributed computing!

Background

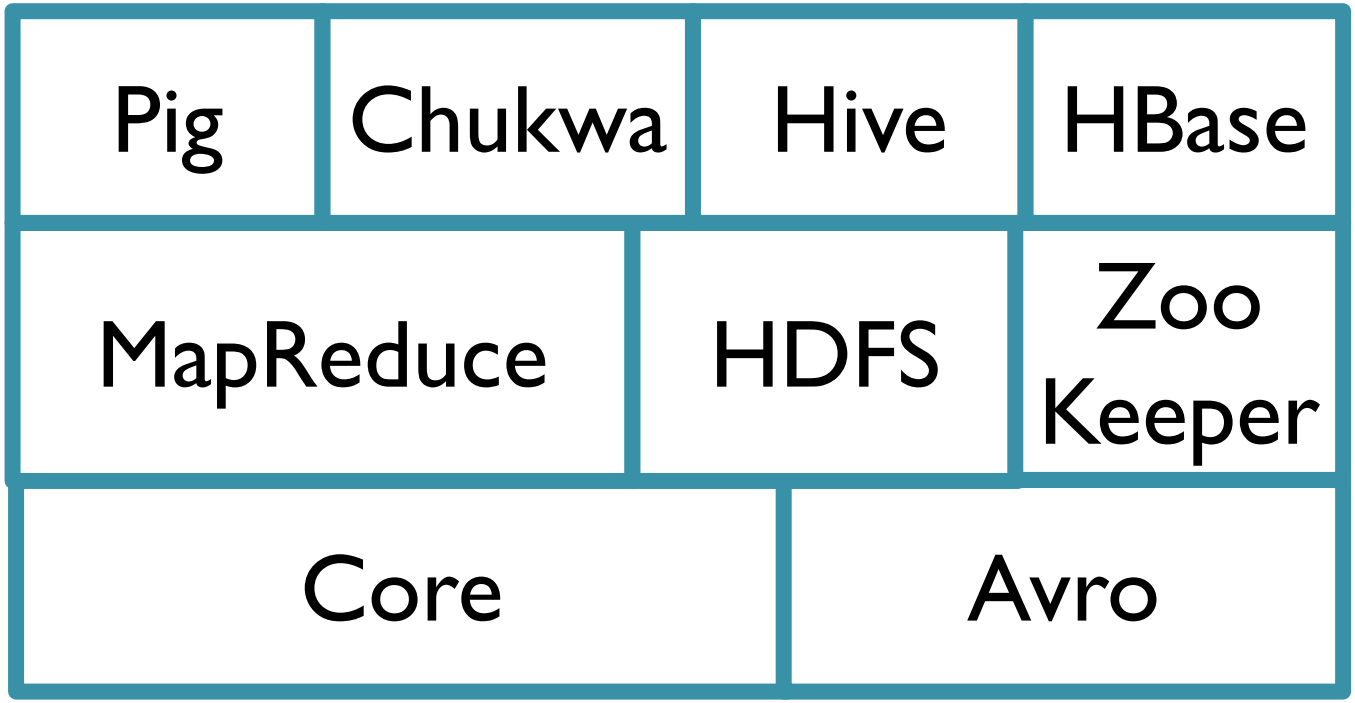
- Hadoop project provides you
  - Concept, ideas & framework
  - System to install (it is like the operating system, but on a cluster of machines)
  - Libraries (so that you can write your own distributed computing applications)

Overview of Hadoop

- Hadoop has multiple subprojects
  - HDFS: a distributed filesystem
  - MapReduce: a distributed data processing model and execution environment
  - Pig: a data flow language and execution environment for exploring very large datasets
  - HBase: a distributed, column-oriented database
  - ZooKeeper: a distributed, highly available coordination service
  - Hive: a distributed data warehouse

Overview of Hadoop

- Hadoop subprojects



Source: White, Tom. "*Hadoop: The definitive guide.*"

Overview of Hadoop

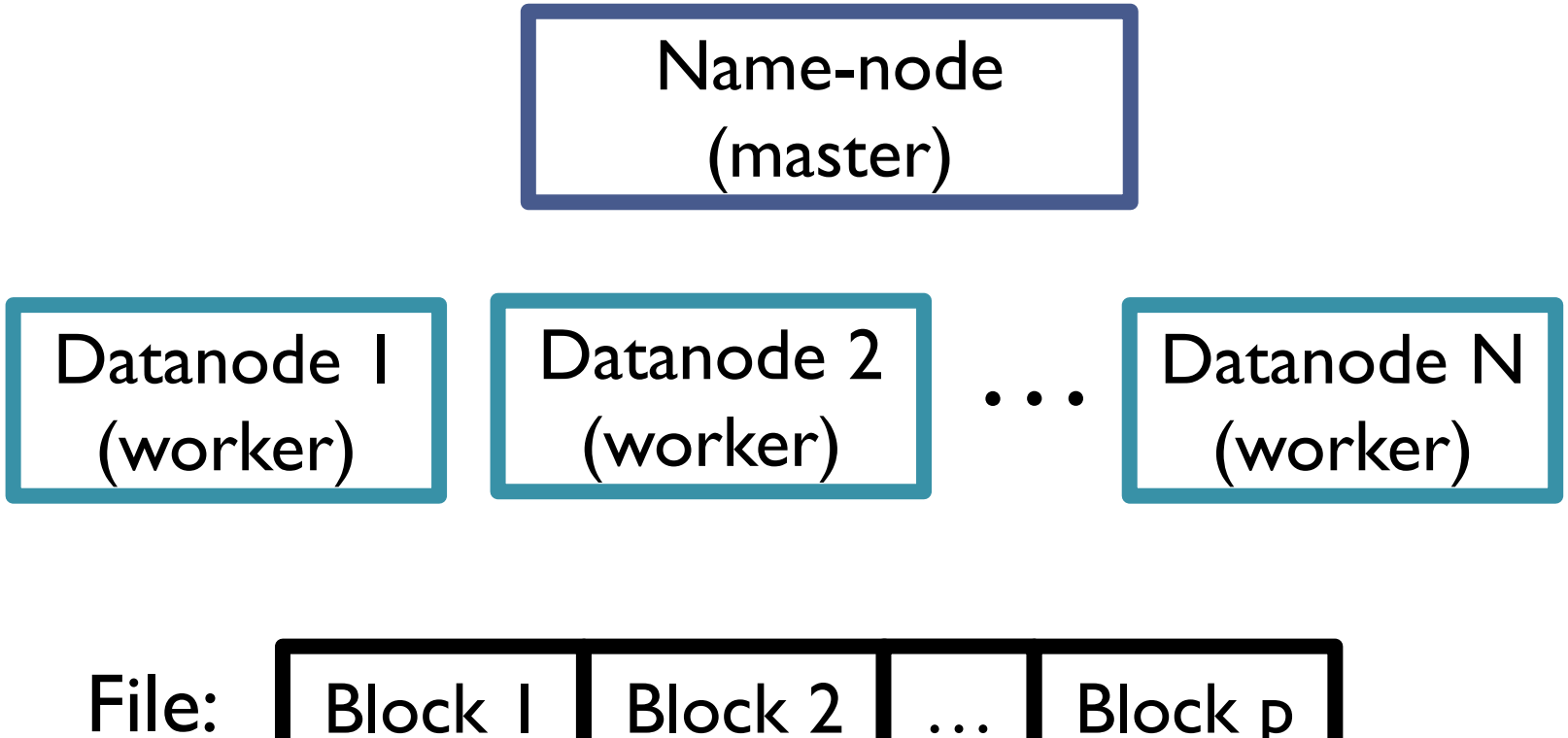
- Kernel subprojects of Hadoop
  - HDFS: data storage
  - MapReduce: data analysis
  - The two parts provide systematical solution to the problems in Slide #8

HDFS

- HDFS: Hadoop Distributed Filesystem
  - Manage storage across a network of machines
- HDFS is suitable for
  - Very large files (GB or TB sized)
  - Streaming data access (write once, read many times)
  - Commodity hardware
- HDFS is not suitable for
  - Low-latency data access
  - Lots of small files
  - Multiple writers, arbitrary file modifications

HDFS

- Structure of HDFS



HDFS

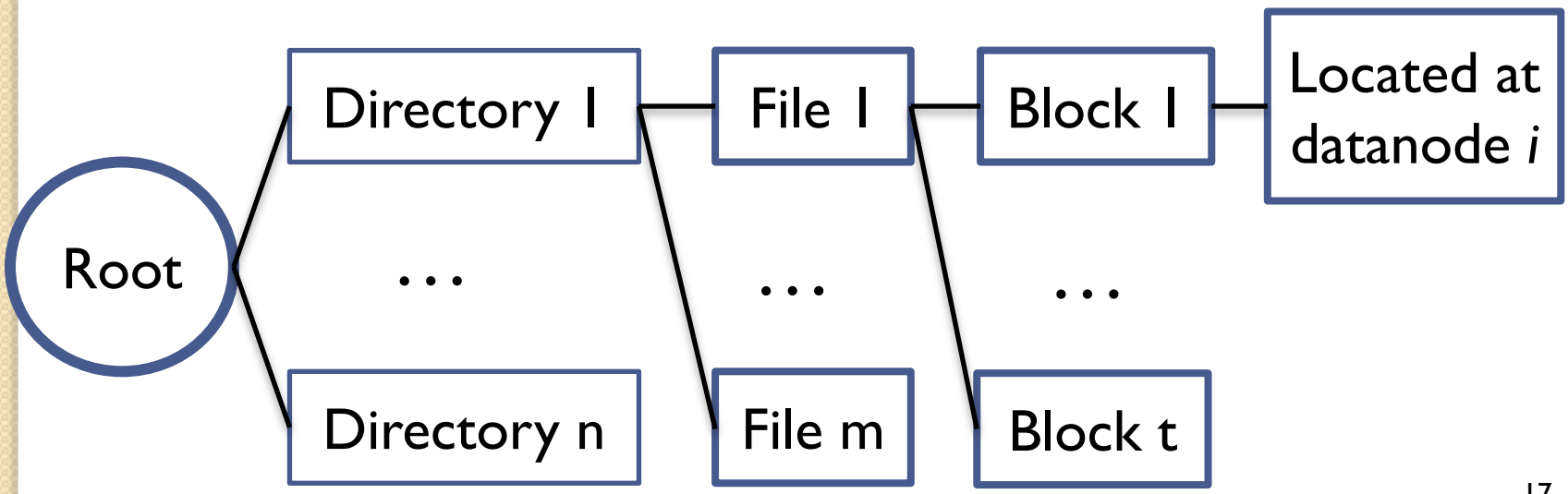
- Structure of HDFS
  - Block
    - A block size is the minimum amount of data that the system can read or write.
    - Files in HDFS are broken into block-sized chunks (e.g. 64MB), which are stored independently.
    - Each block is replicated (e.g. 3 copies) in case of hardware corruption.

HDFS

- Structure of HDFS
  - Datanodes
    - Store and retrieve blocks
    - Report to namenode periodically with lists of blocks that they are storing

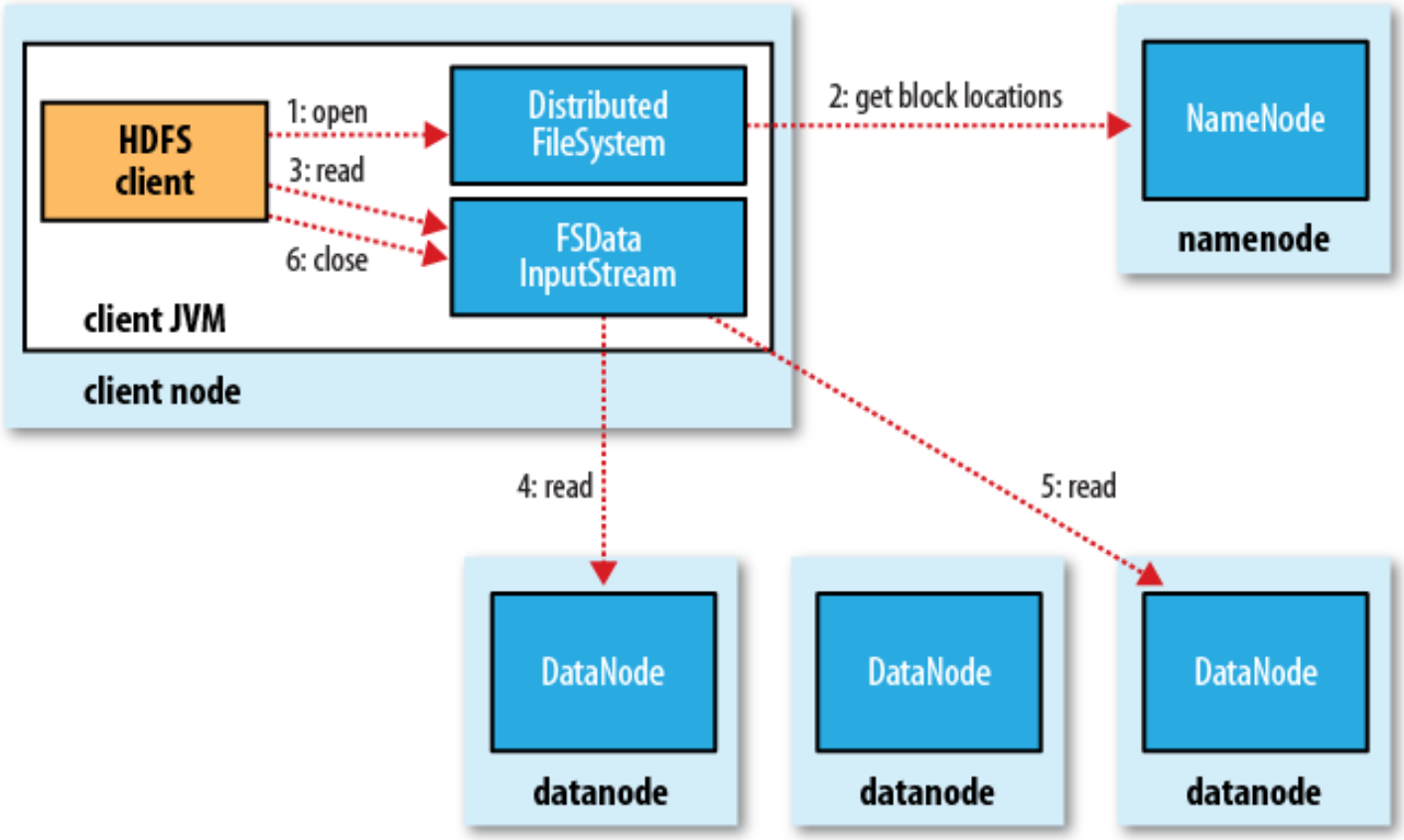
HDFS

- Structure of HDFS
  - Name-node
    - Maintains the filesystem tree and metadata for all the files and directories in the tree
    - Knows the datanodes on which all blocks of a given file are located



HDFS

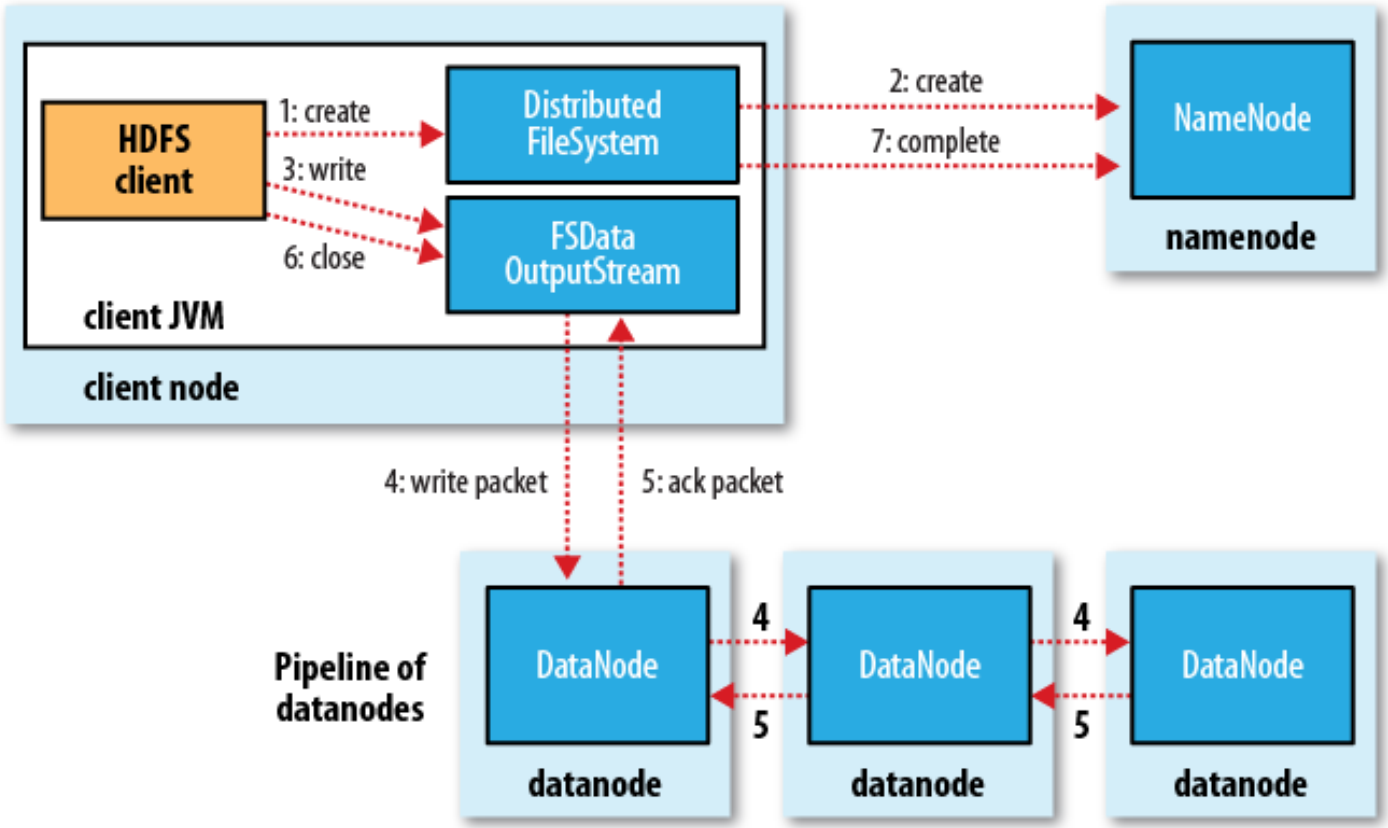
- Data reading in HDFS



Source: White, Tom. "*Hadoop: The definitive guide.*" 3rd Edition, Page 68, O'Reilly Media, Inc., 2012.

HDFS

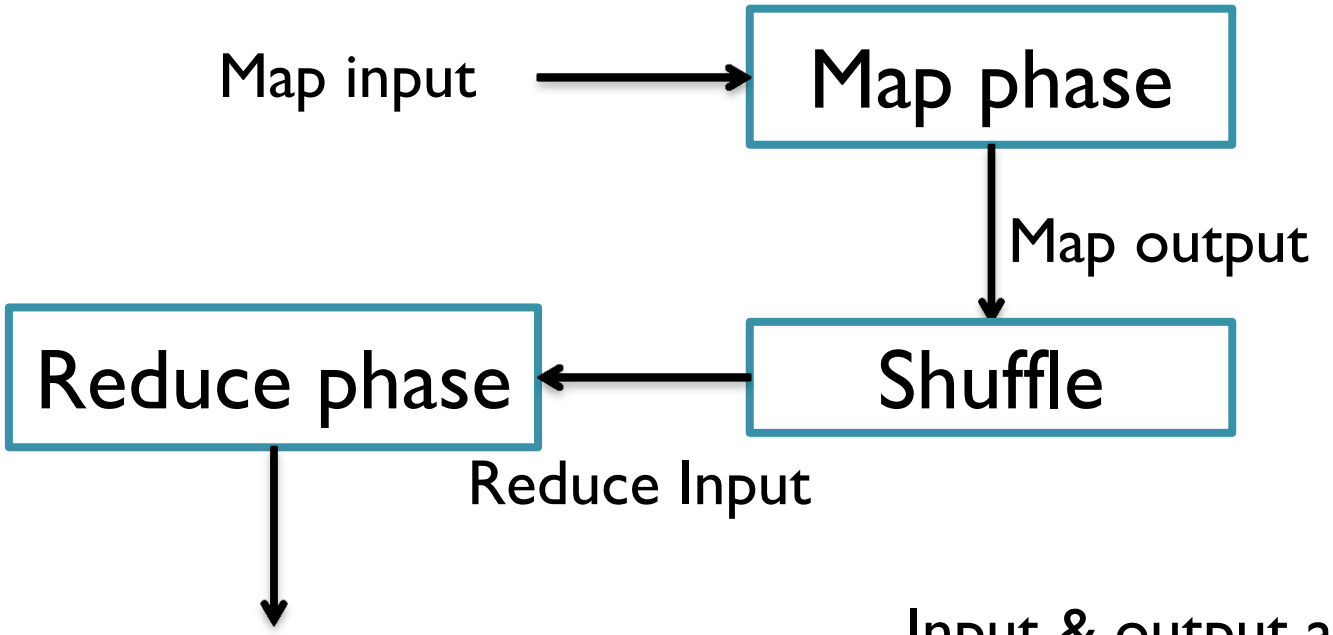
- Data writing in HDFS



Source: White, Tom. "*Hadoop: The definitive guide*. " 3rd edition, page 71, O'Reilly Media, Inc., 2012.

MapReduce

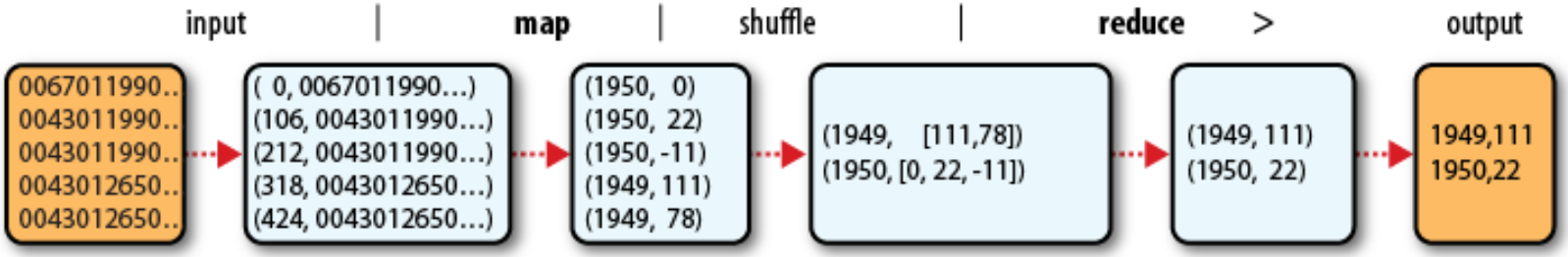
- MapReduce: a parallel data-processing model for distributed computing
- Structure of MapReduce



Input & output are in (key, value) format

MapReduce

- An example
  - Find the highest recorded temperature for each year in the dataset



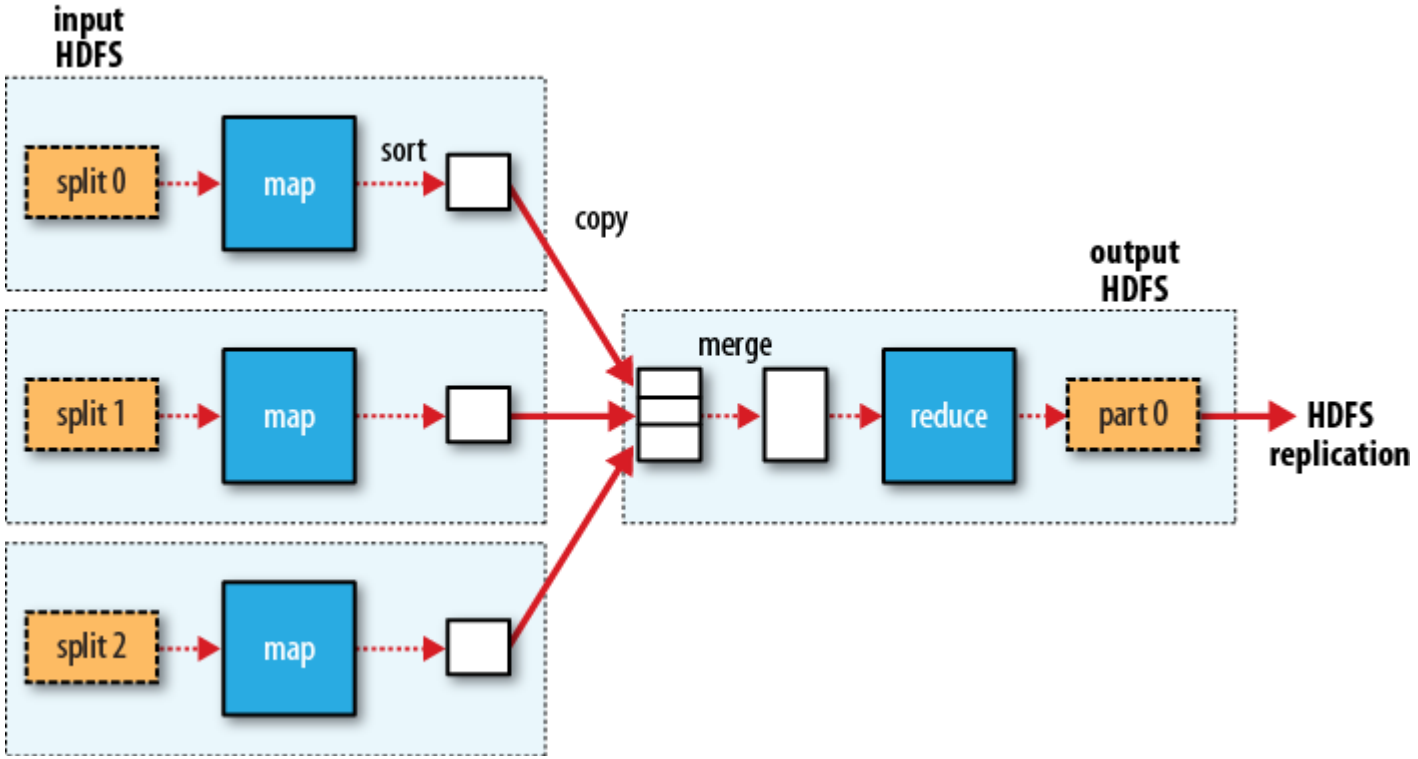
  - Input: each line contains various information for a specific day
  - Map: extract year and temperature from each line
  - Shuffle: merge the map output for each year
  - Reduce: calculate the maximum temperature for each year

MapReduce

- For parallel computing
  - Divide a job (the unit of work we want to do) into multiple tasks (including map tasks and reduce tasks)
  - Accordingly, input dataset is split into equal-sized pieces (usually the block size of HDFS)
  - Use jobtracker and tasktrackers for coordination

MapReduce

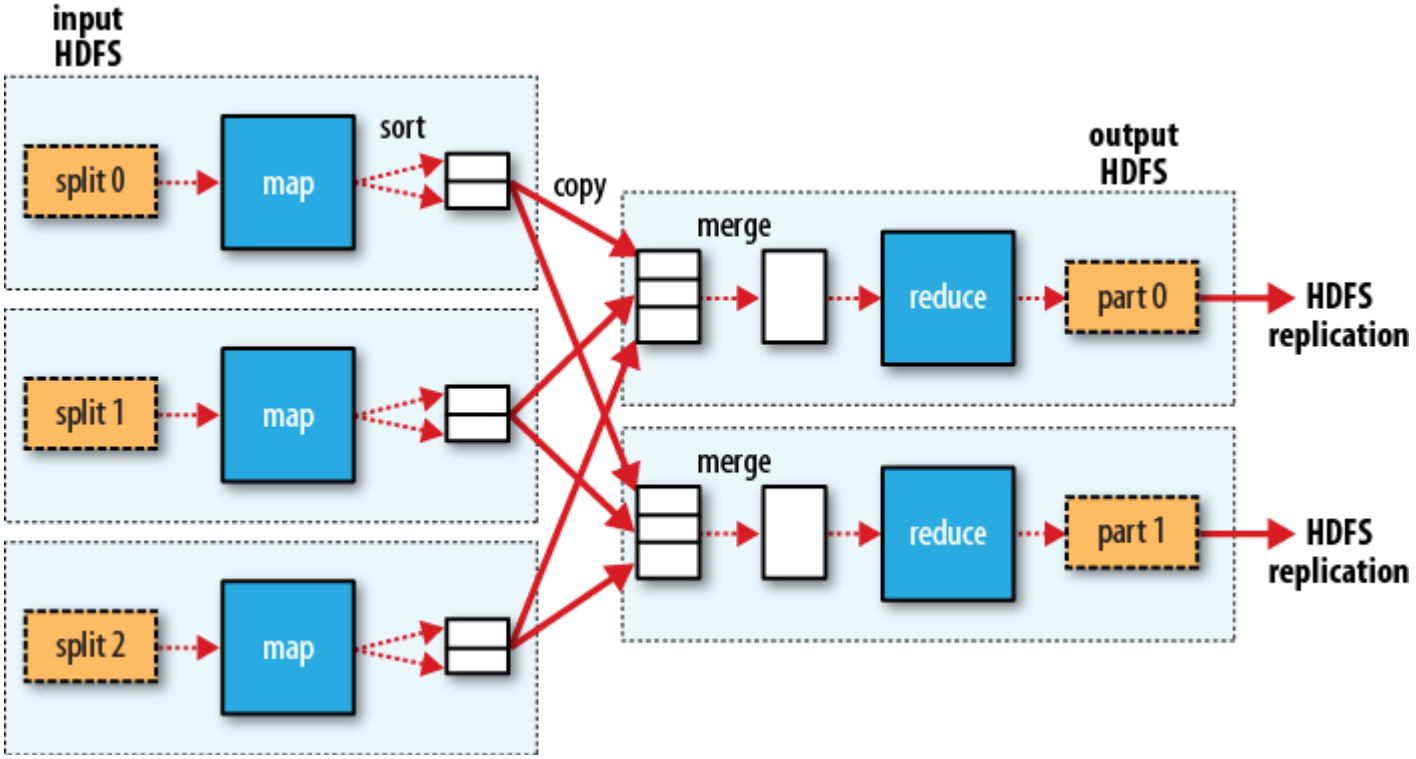
- For parallel computing
  - Single reduce task



Source: White, Tom. "*Hadoop: The definitive guide*. " 3rd edition, page 32, O'Reilly Media, Inc., 2012.

MapReduce

- For parallel computing
  - Multiple reduce tasks



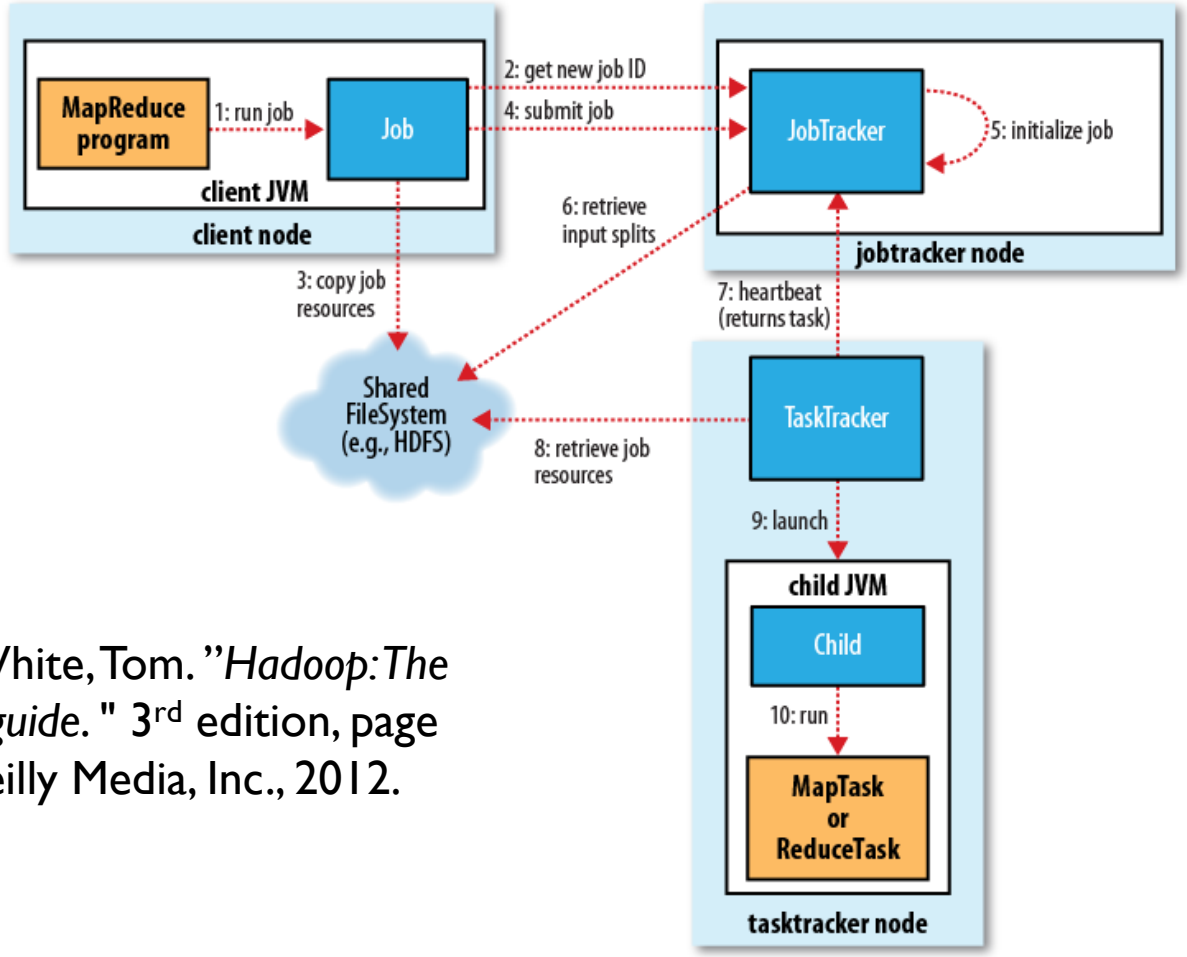
Source: White, Tom. "*Hadoop: The definitive guide*." 3rd edition, page 33, O'Reilly Media, Inc., 2012.

MapReduce

- Data locality optimization
  - Hadoop does its best to run map task on the computer where input split resides in HDFS
  - Data locality decreases the data transfer through the network
  - However, reduce tasks don't have the advantage of data locality
  - Combiner functions can be used between map and reduce to minimize data transfer through the network

MapReduce

- How Hadoop runs a MapReduce job



Source: White, Tom. "*Hadoop: The definitive guide.*" 3rd edition, page 191, O'Reilly Media, Inc., 2012.

Pig

- Pig is the language and execution environment for processing large datasets
  - MapReduce deals with low-level operations and Pig constructs high-level operations based on MapReduce
- Pig consists of two pieces:
  - Pig Latin: the language to express data flow
  - Execution environment to run Pig Latin programs

Pig

- Pig Latin program
  - Consists of a series of operations or transformations applied to input data
  - Pig execution environment translates the language into a series of MapReduce jobs

Pig

- An example
  - Find the highest recorded temperature for each year (same with Slide #21)

```
-- max_temp.pig: Finds the maximum temperature by year
records = LOAD 'input/ncdc/micro-tab/sample.txt'
  AS (year:chararray, temperature:int, quality:int);
filtered_records = FILTER records BY temperature != 9999 AND
  (quality == 0 OR quality == 1 OR quality == 4 OR quality == 5 OR quality == 9);
grouped_records = GROUP filtered_records BY year;
max_temp = FOREACH grouped_records GENERATE group,
  MAX(filtered_records.temperature);
DUMP max_temp;
```

Source: White, Tom. "*Hadoop: The definitive guide.* "

HBase

- HBase: a distributed column-oriented database built on HDFS
  - Doesn't support SQL
  - Automatically partitions a table into regions horizontally if the table grows too large
  - Uses a master node to manage the whole space and regionserver slaves to manage one or more regions

Hive

- Hive: a data warehouse infrastructure built on top of Hadoop
  - Supports the Hive Query language – very similar to SQL
  - Hive compiler converts SQL commands to MapReduce jobs

Summary

- Hadoop is a collection of subprojects used for parallel data storage, retrieval & analysis in a group of computers/hardwares, in order to mitigate the data access bottleneck in hard drives
- For more information
  - White, Tom. "*Hadoop: The definitive guide*." O'Reilly Media, Inc., 2012.
  - Apache Hadoop homepage *http://hadoop.apache.org/*
- Reference
  - White, Tom. "*Hadoop: The definitive guide*." O'Reilly Media, Inc., 2012.