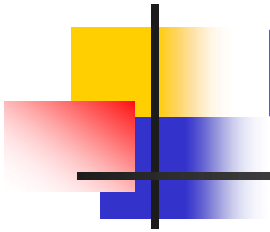
Hadoop architecture

Based on:

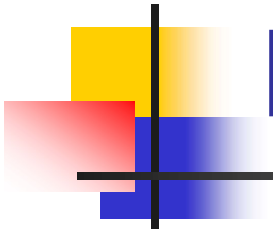
- Hadoop the Definitive Guide [1]
- https://hadoop.apache.org/ [2]



Big Data

- Gartner's definition, circa 2001:
  - "Big data is data that contains greater **variety** arriving in increasing **volumes** and with ever-higher **velocity**"

- Wikipedia
  - "Big data is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data-processing application software."



Big Data

- Some examples (From [1]):
  - Ancestry.com, the genealogy site, stores around 10 PB of data
  - The Internet Archive stores around 18.5 PB of data
  - The New York Stock Exchange **generates** about 4–5 TB of data per day (≈ 20 PB per year)
  - The Large Hadron Collider near Geneva, Switzerland, **produces** about 30 PB of data per year
  - Facebook hosts more than 240 billion photos, **growing** at 7 PB per month

Big Data

Please process my data
(I have a 1 Gbit/s link)

30 PB per year ≈ 2.5 PB per month

2.5 PB = 2.5 x 10^{15} byte

$$\frac{2.5 \times 10^{15} \times 8}{1 \times 10^{9}} \times bit \times \frac{second}{bit} = 20 \times 10^{(15-9)}\ second \approx 231\ days\ (0.6\ years)$$

(or 6.3 years if link speed is 100 Mbit/s)

Big Data and Hadoop

- "MapReduce provides a programming model that abstracts the problem from disk reads and writes, transforming it into a computation over sets of keys and values." [1]

- "Hadoop provides a reliable, scalable platform for storage and analysis … because it runs on commodity hardware and is open source, Hadoop is affordable." [1]

(Apache) Hadoop *vs.* MapReduce

"Hadoop is an Eco-system of open source projects such as Hadoop Common, Hadoop distributed file system (HDFS), Hadoop YARN, Hadoop MapReduce. Hadoop as such is an open source framework for storing and processing huge datasets. The storing is carried by HDFS and the processing is taken care by MapReduce. MapReduce, on the other hand, is a programming model which allows you to process huge data stored in Hadoop"

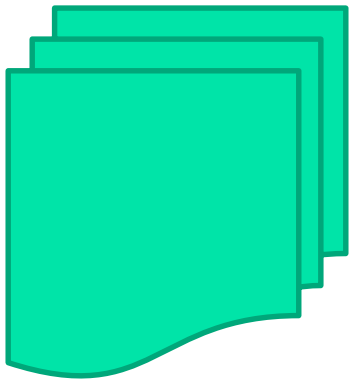
https://www.educba.com/hadoop-vs-mapreduce/

MapReduce fundamentals

- "MapReduce: Simplified Data Processing on Large Clusters"
  - https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf
    - Jeffrey Dean (jeff@google.com) and
    - Sanjay Ghemawat (sanjay@google.com)

MapReduce fundamentals

- Counting the number of occurrences of words in a large collection of files (sequential approach)



```
for each document as doc {
  for each word in doc as w {
    if ( tableOfWords.contains( w ) {
      tableOfWord[ w ] += 1;
    }
    else {
      tableOfWords.add( w, 1 );
    }
  }
}
```

MapReduce fundamentals

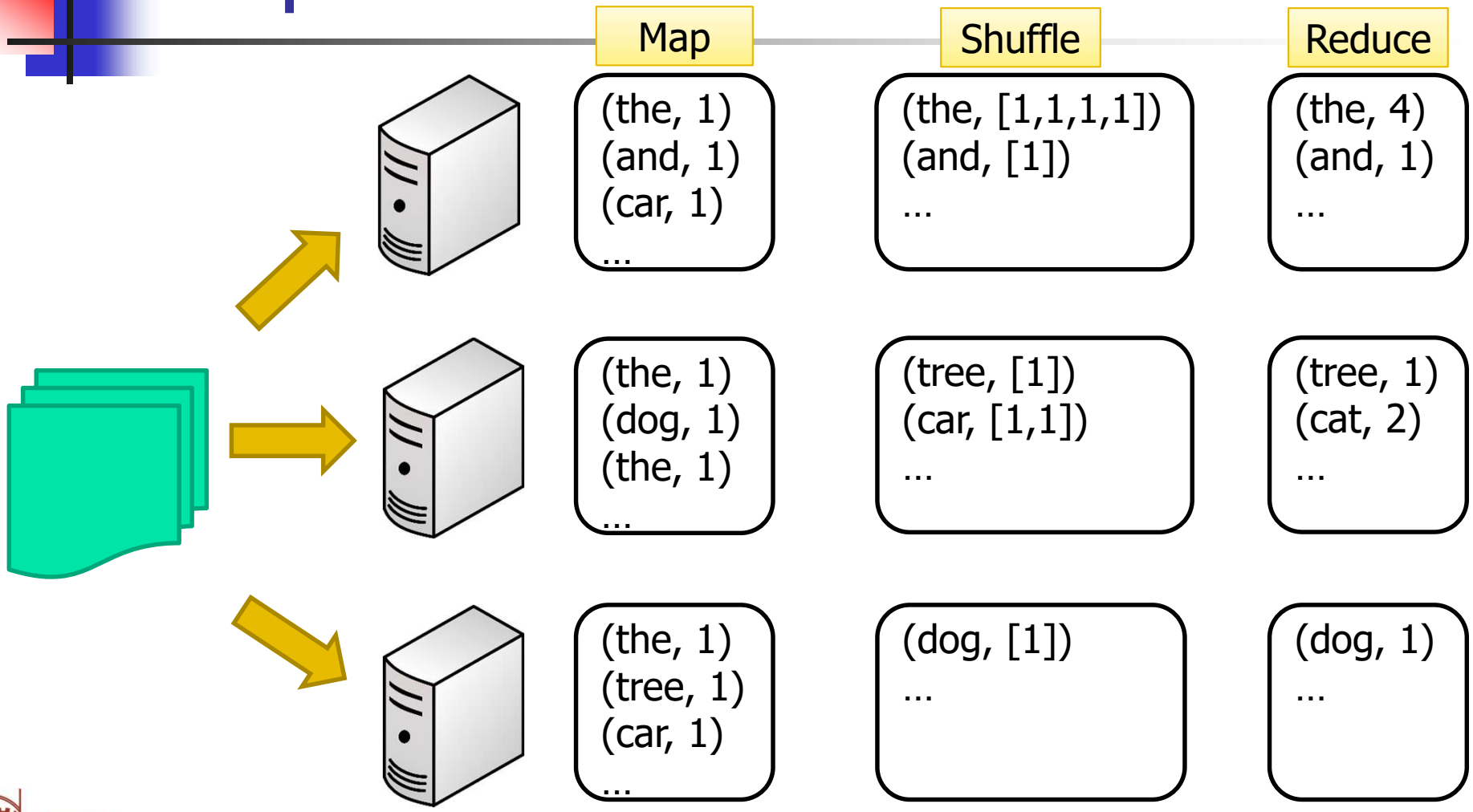
- Map phase

```
map(String key, String value):
// key:document name
// value:document contents
for each word w in value:
  EmitIntermediate( w, "1" );
```

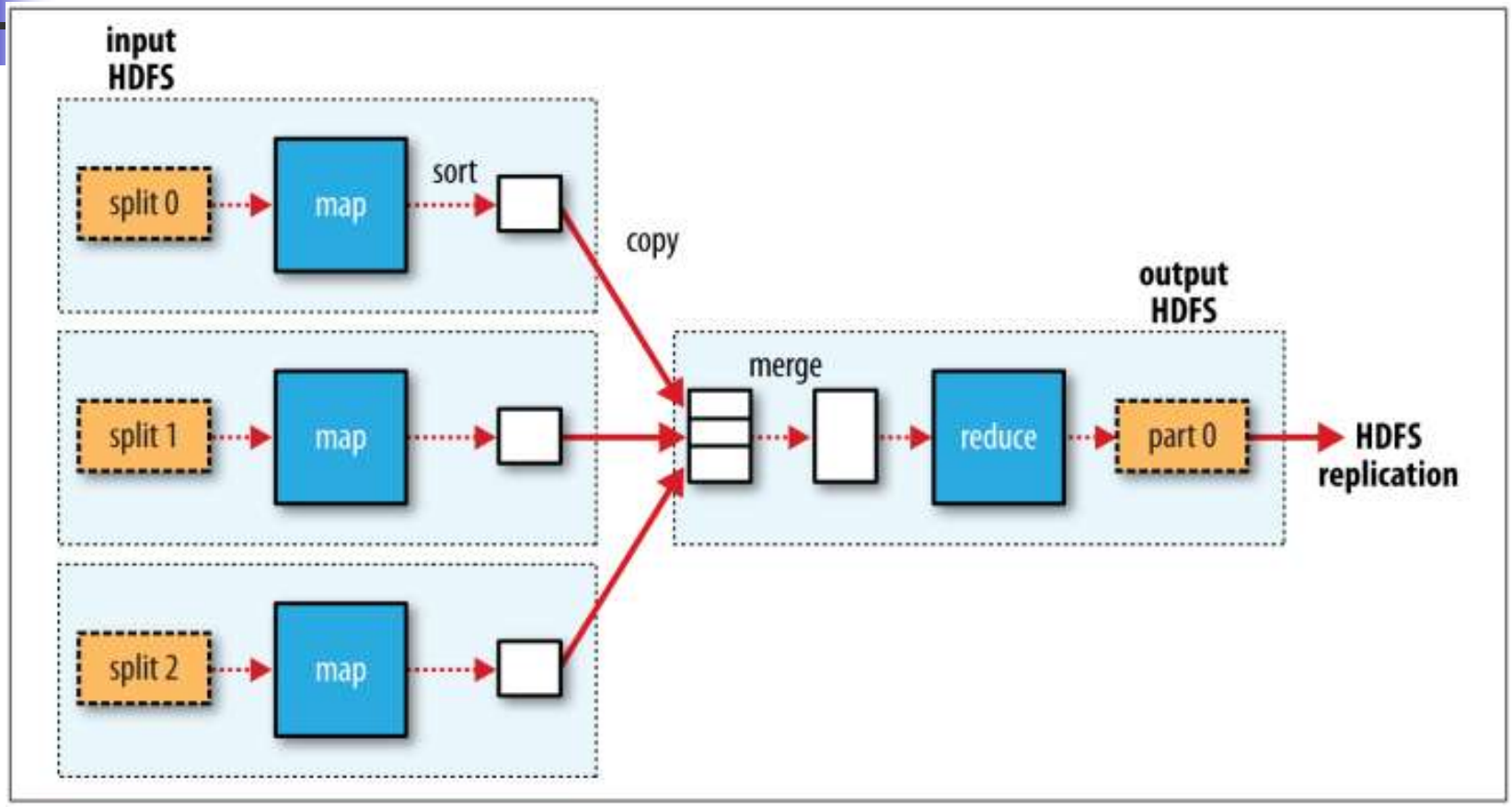
- Reduce phase

```
reduce(String key, Iterator values):
// key: a word
// values: a list of counts
int result = 0;

for each v in values:
  result += ParseInt( v );

Emit( key, AsString( result ) )
```

MapReduce fundamentals

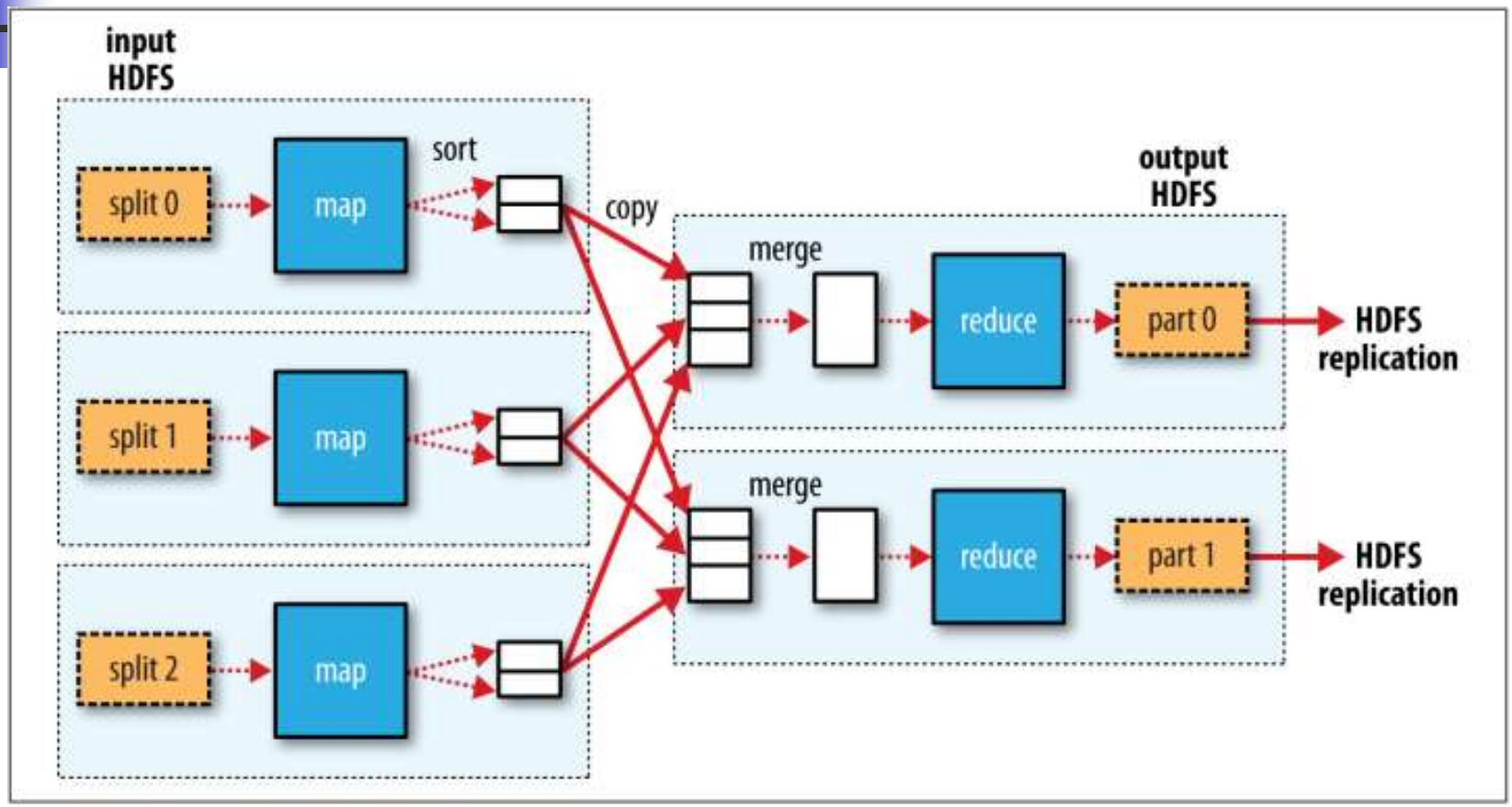


MapReduce fundamentals



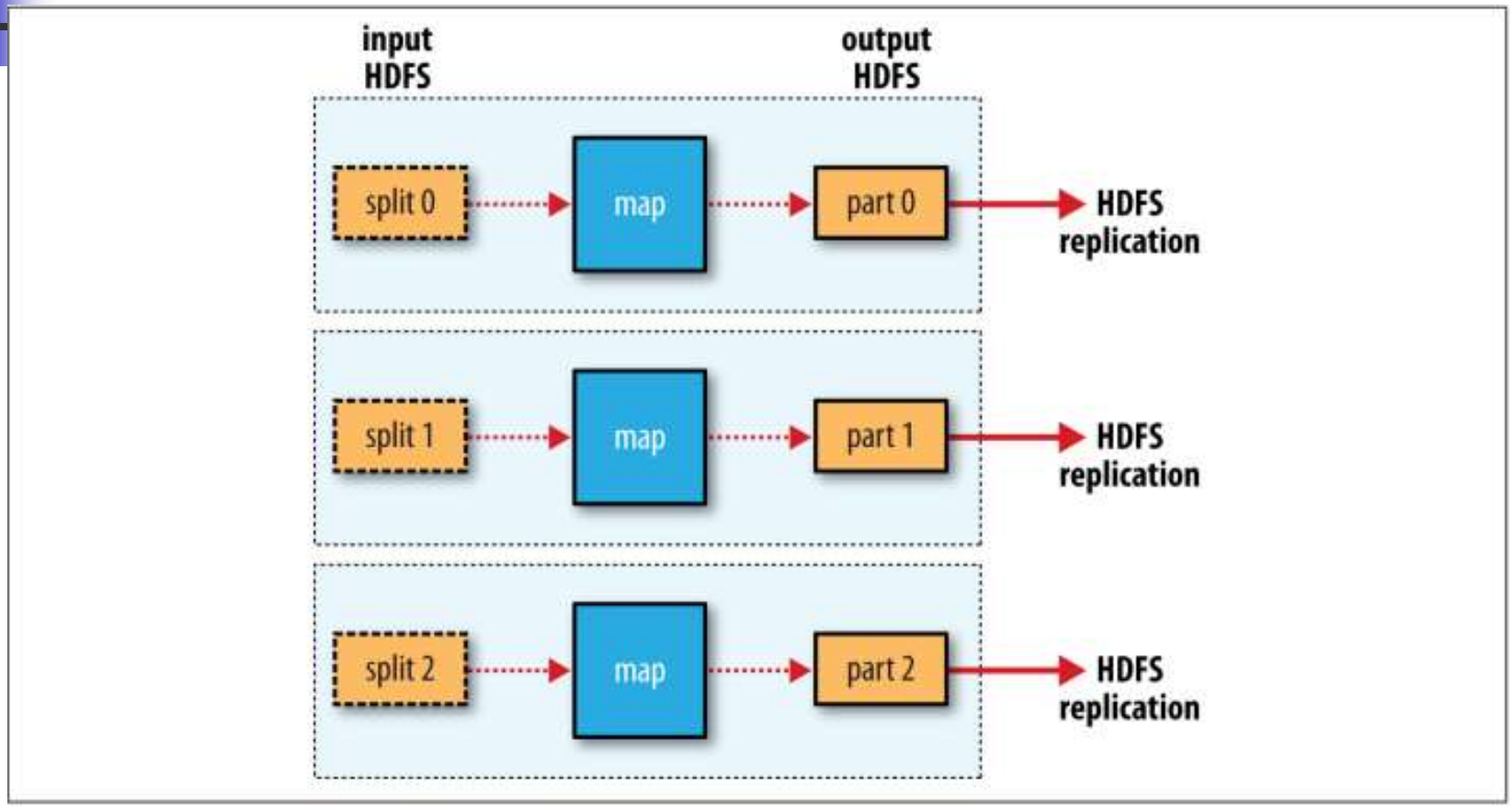
MapReduce data flow with a single reduce task (from [1])

MapReduce fundamentals



MapReduce data flow with multiple reduce tasks (from [1])

MapReduce fundamentals



MapReduce data flow with no reduce tasks (from [1])

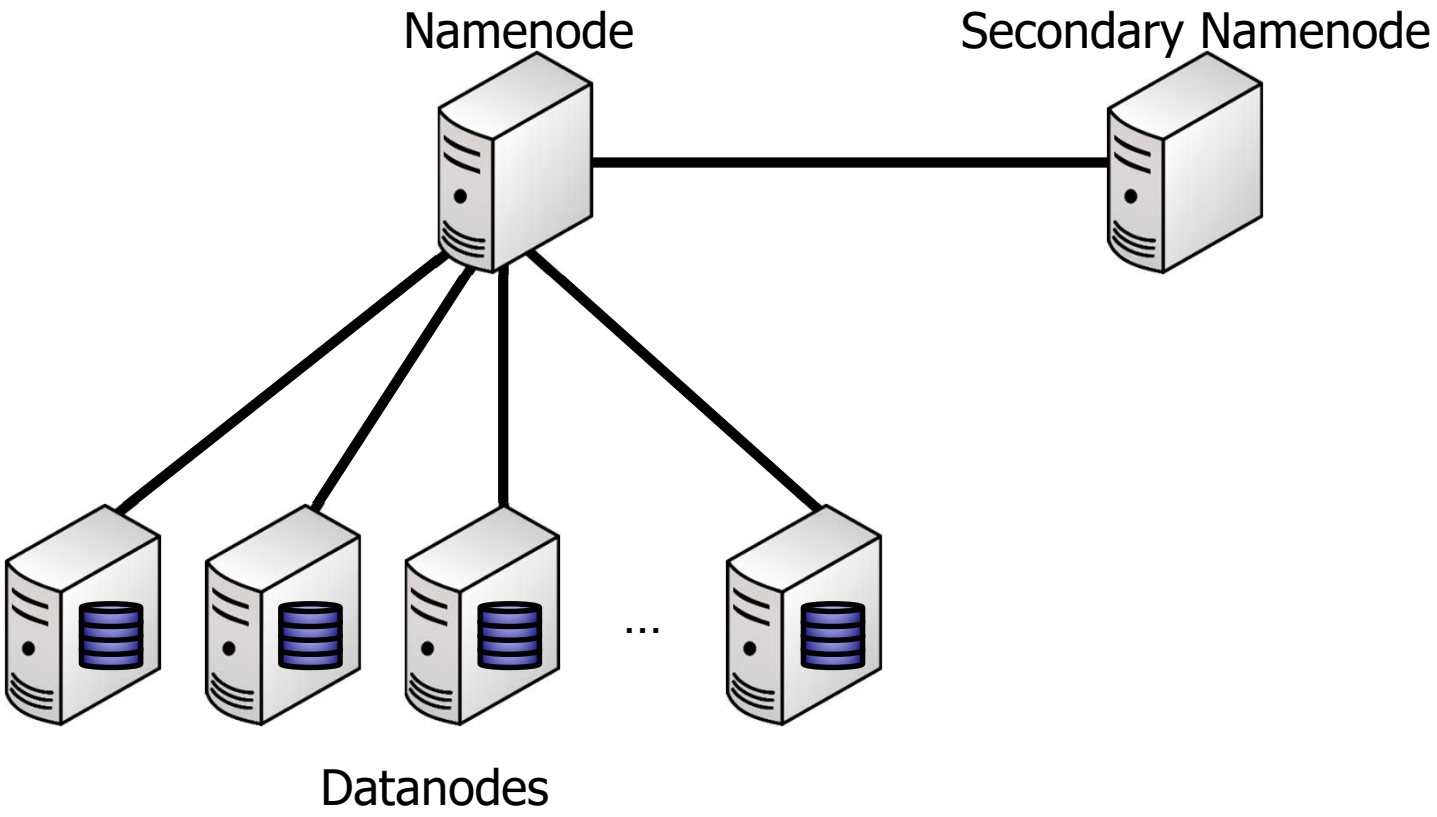
Hadoop Architecture

Hadoop core components

- HDFS – Hadoop Distributed File System
  - Master/slave architecture
  - Distributed file system designed to run on commodity hardware
- MapReduce
- YARN – Yet Another Resource Negotiator
  - Split up the functionalities of resource management and job scheduling/monitoring into separate daemons/processes

HDFS

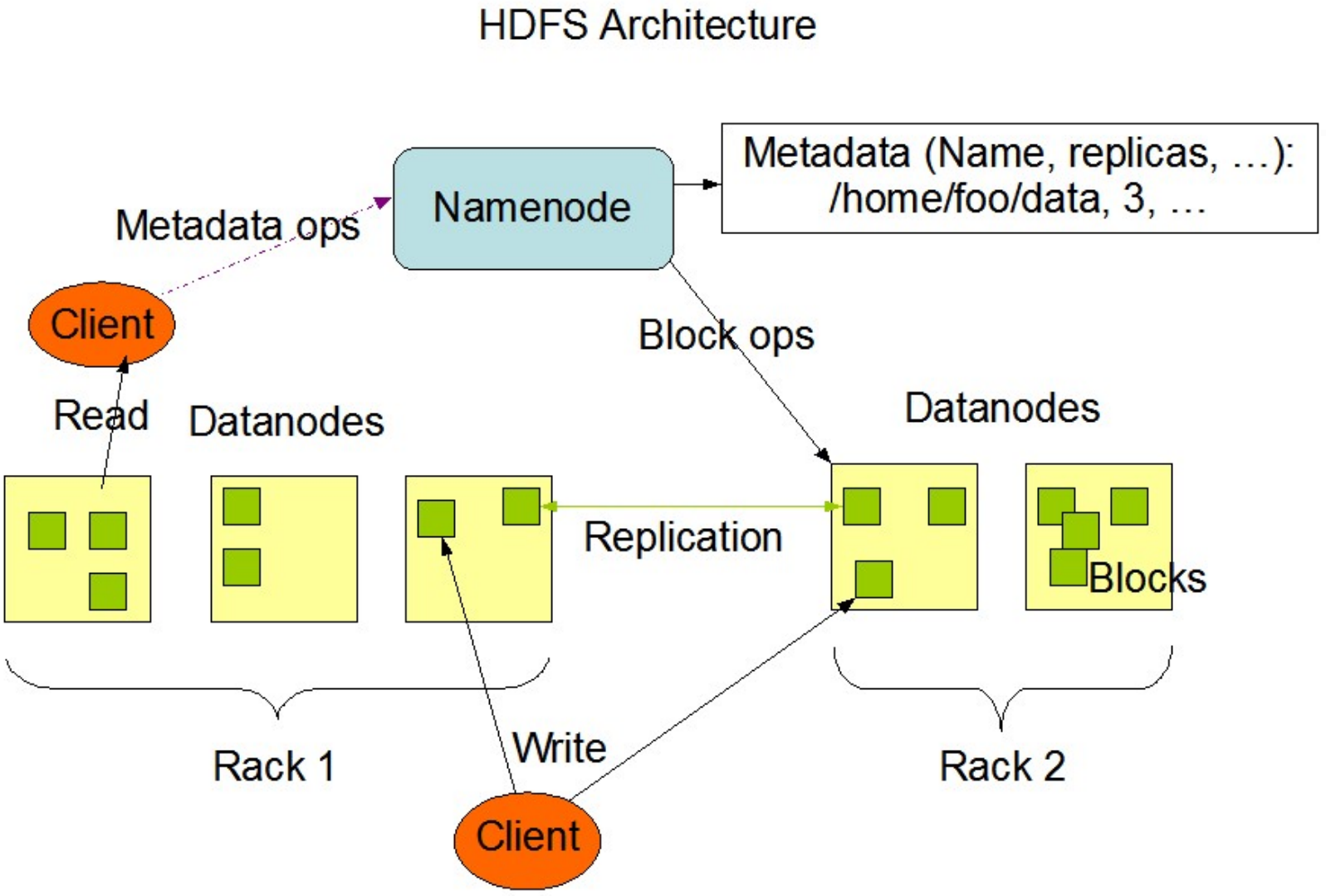
HDFS – Big picture



HDFS – Big picture

- Namenode – Master server that manages the file system namespace and regulates access to files by clients
- Secondary Namenode – Responsibility of merging `editlogs` with `fsimage` from the Namenode
- Datanodes – Clients, usually one per node in the cluster, manage storage attached to the nodes where they run

HDFS – Big picture



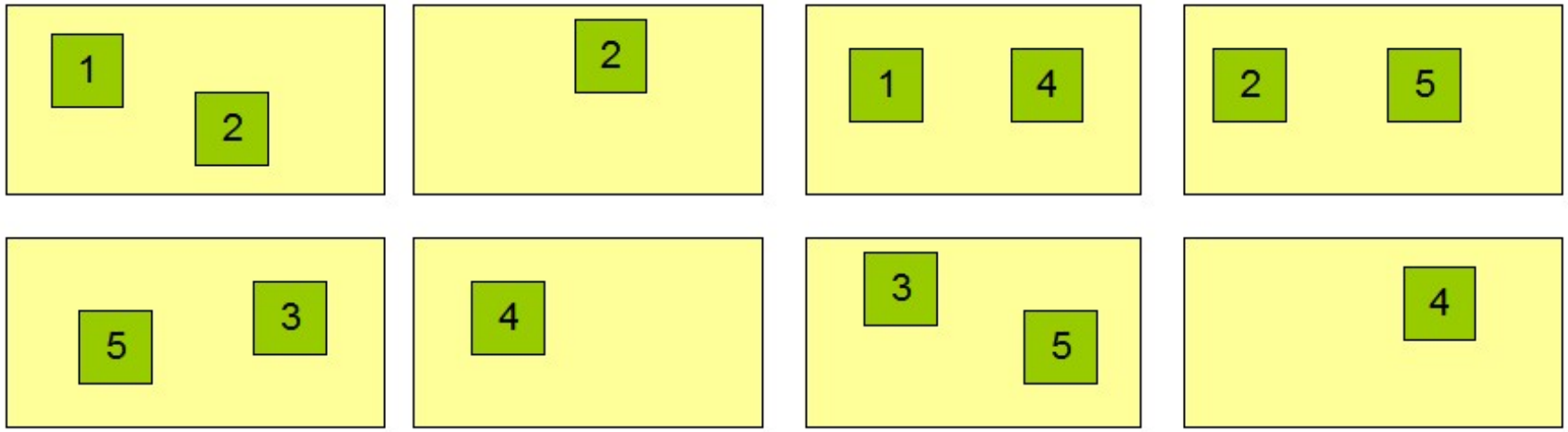
From [2]

HDFS – Big picture

Block Replication

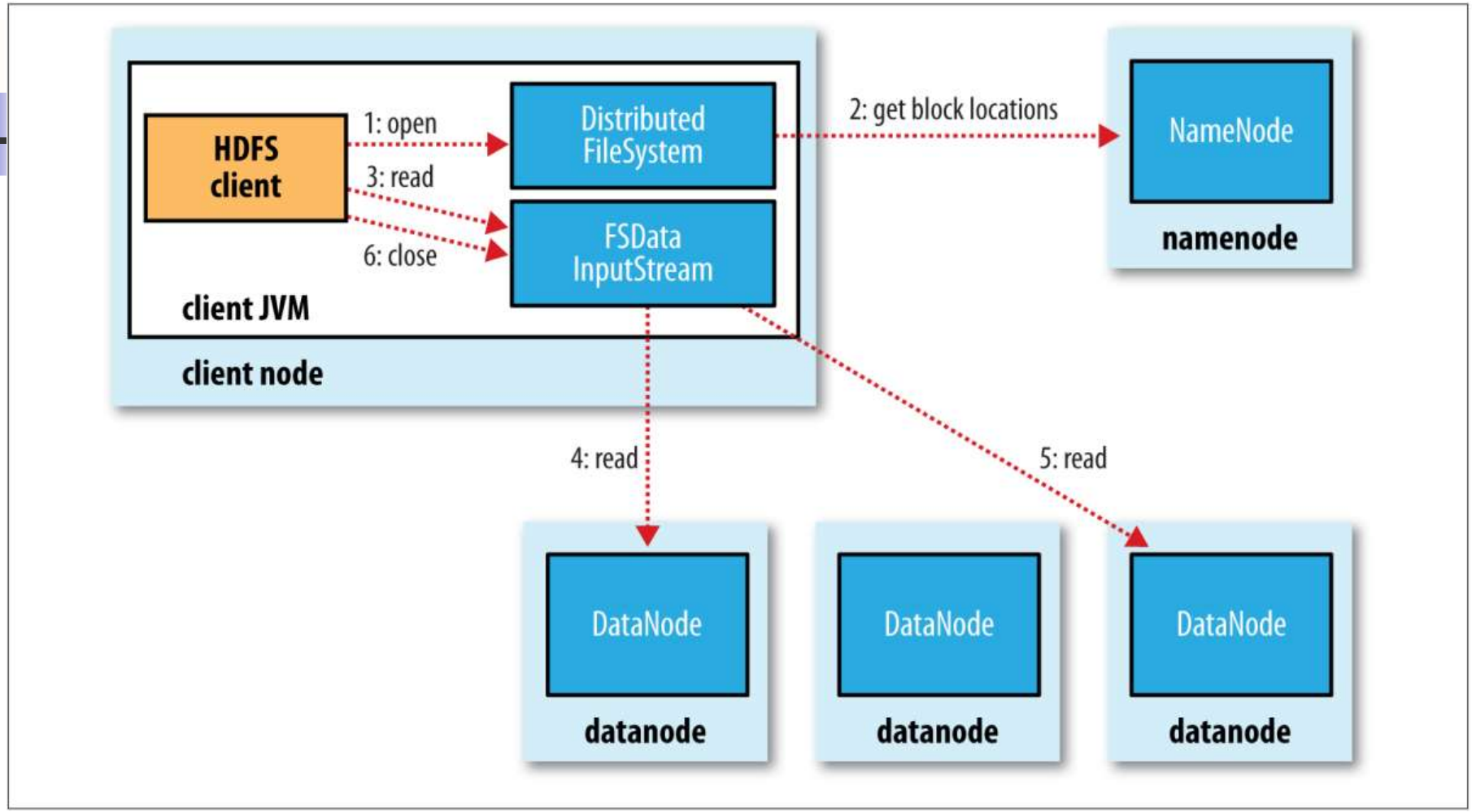
Namenode (Filename, numReplicas, block-ids, …)
/users/sameerp/data/part-0, r:2, {1,3}, …
/users/sameerp/data/part-1, r:3, {2,4,5}, …

Datanodes



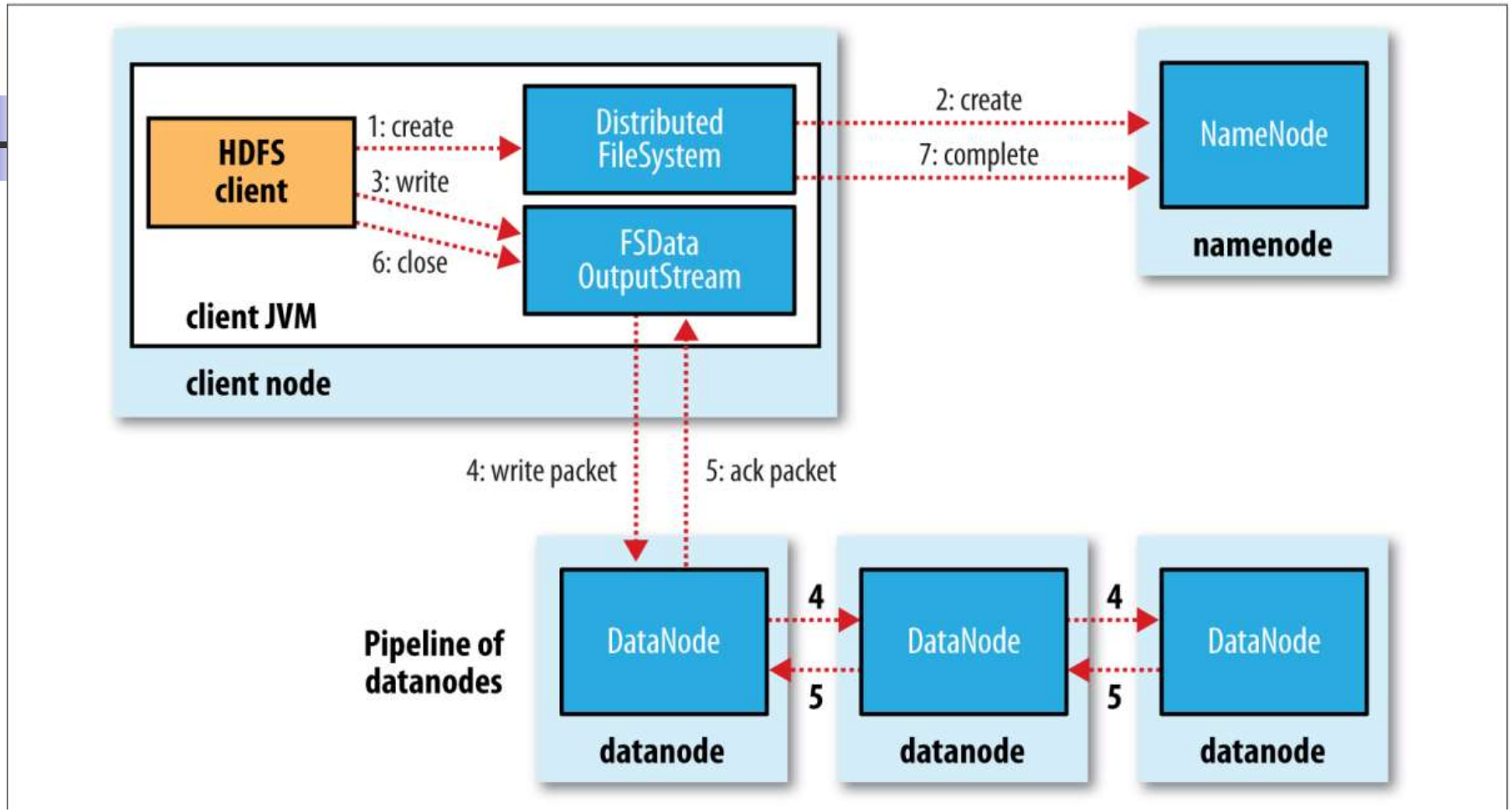
From [2]

HDFS – Big picture



Anatomy of a File Read (from [1])

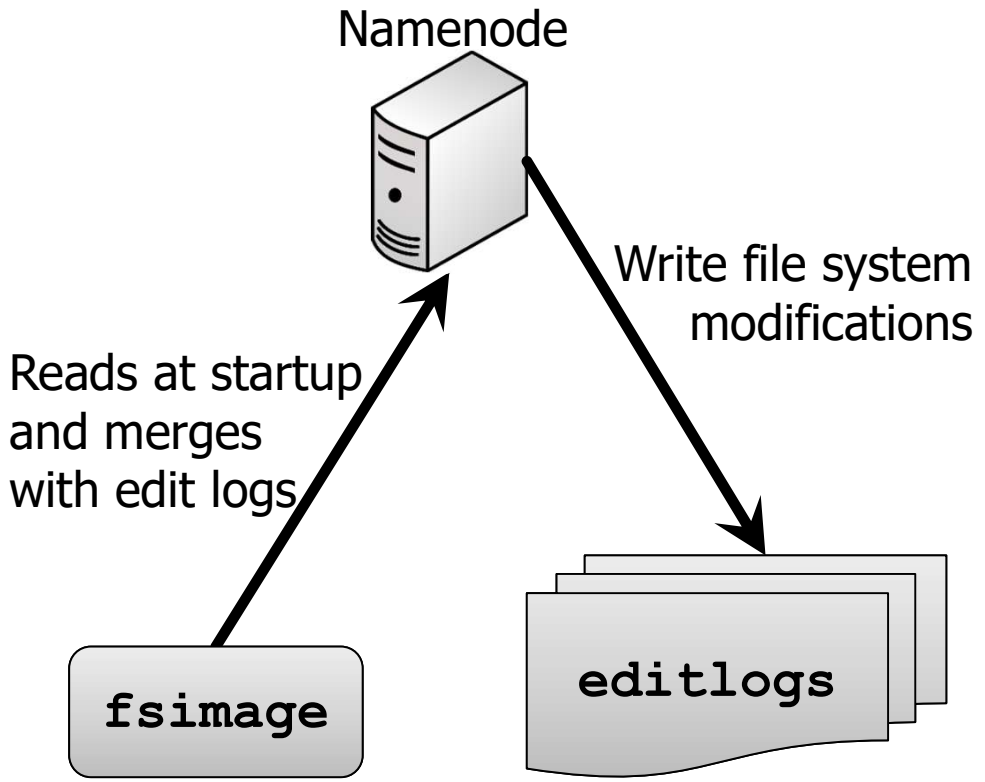
HDFS – Big picture



Anatomy of a File Write (from [1])

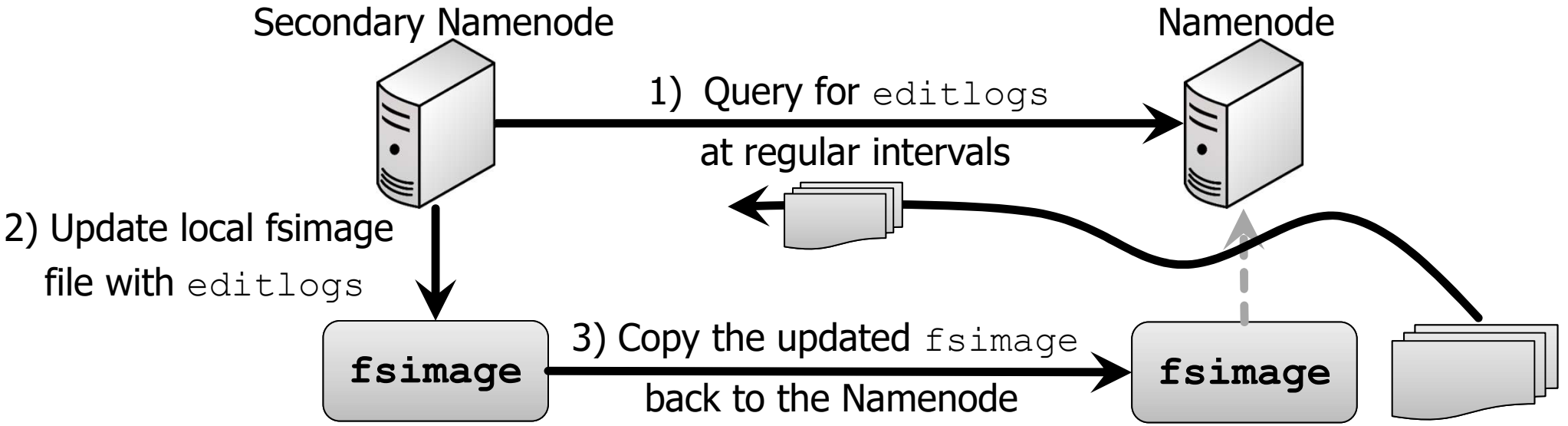
HDFS – Big picture

- Namenode – Holds the meta data for the HDFS like Namespace information, block information etc. When in use, all this information is stored in main memory. This information is also stored in disk for persistence storage



HDFS – Big picture

- Secondary Namenode – The purpose of this component is to have a checkpoint of HDFS. It is just a helper node for Namenode



YARN

YARN – Big picture

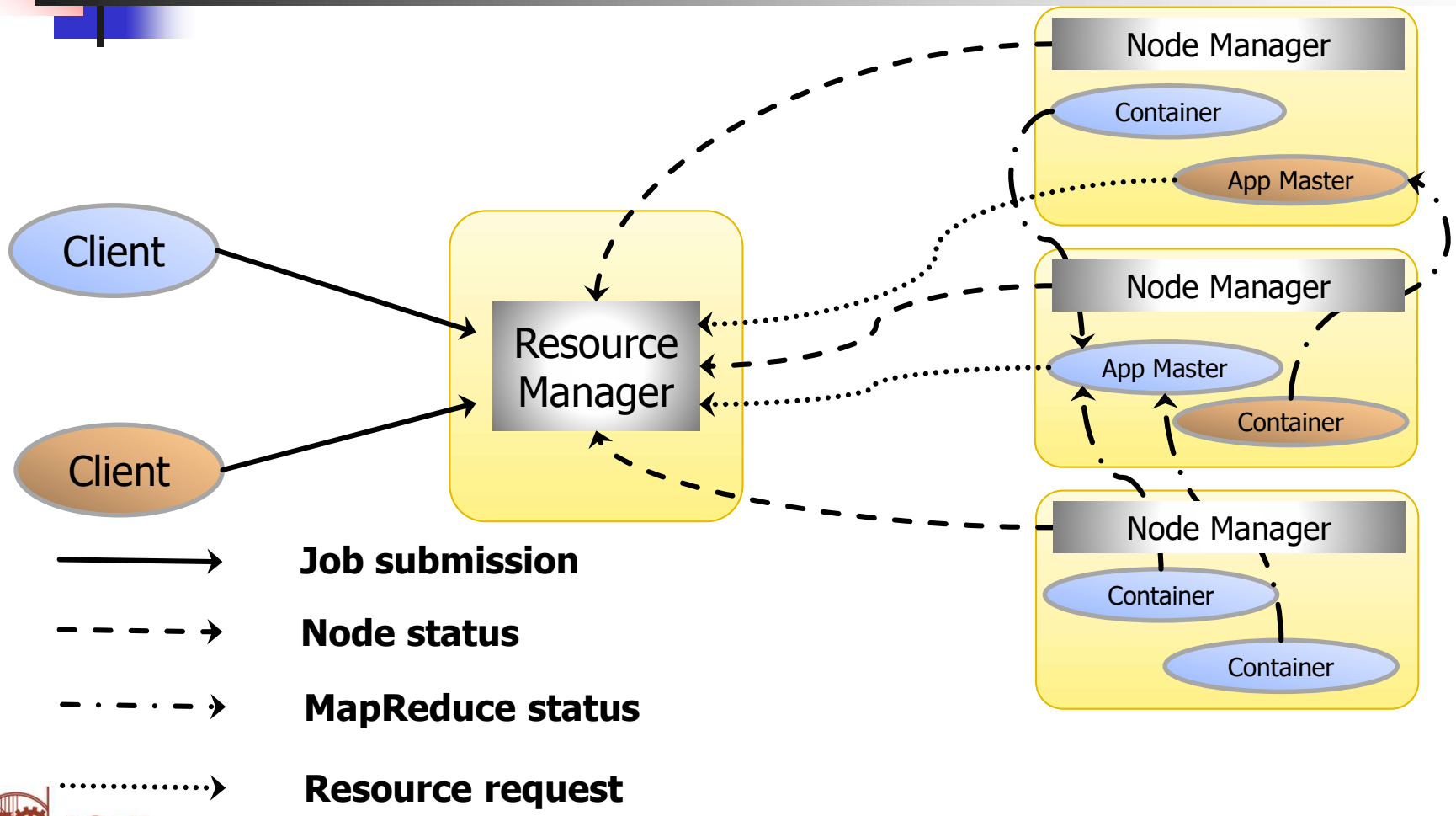
- The `ResourceManager` and the `NodeManager`(s) form the data-computation framework.

- The `ResourceManager` is the ultimate authority that arbitrates resources among all the applications in the system.

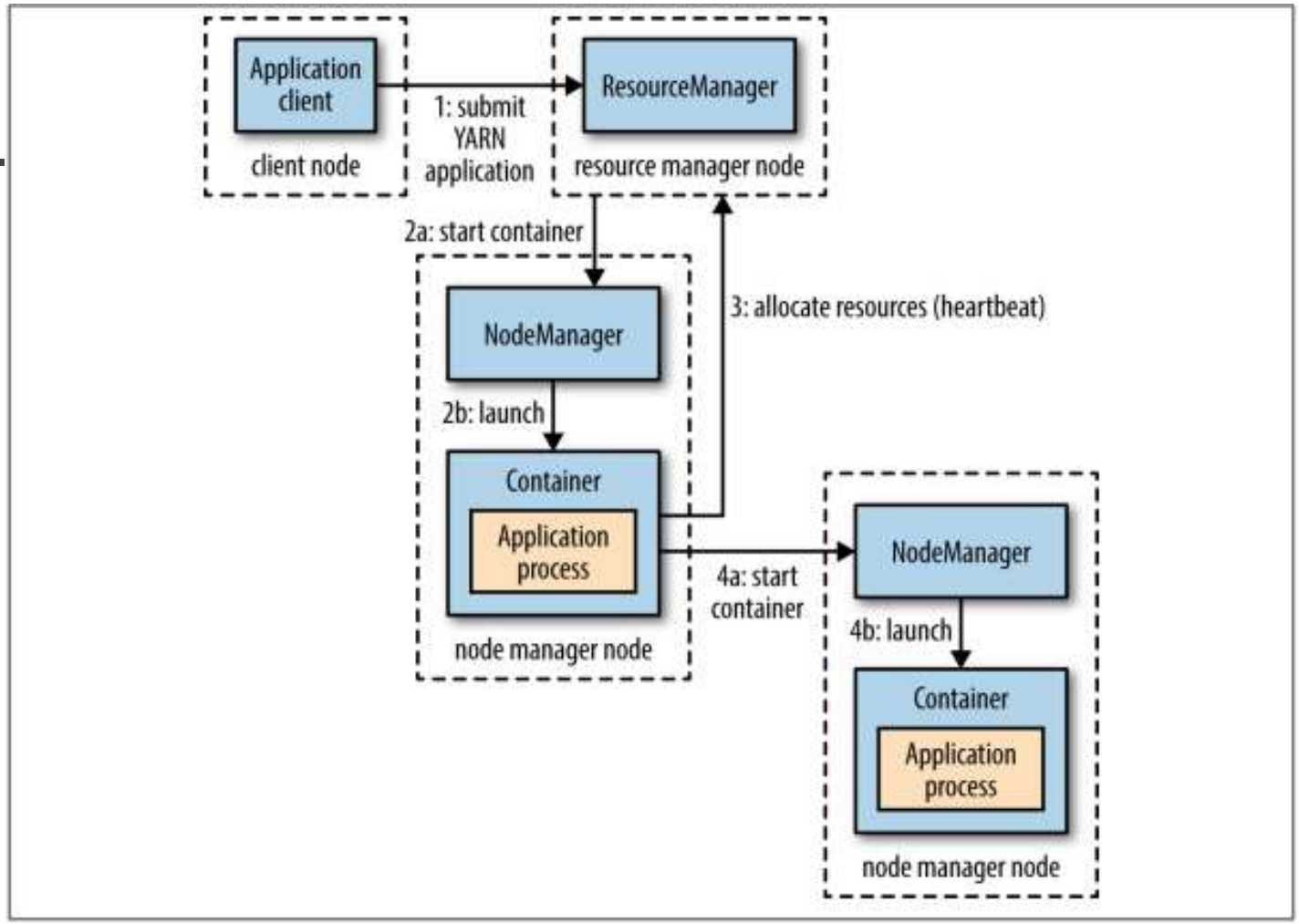
YARN – Big picture

- The `NodeManager` is the per-machine framework agent who is responsible for containers, monitoring their resource usage (CPU, memory, disk, network) and report the same to the `ResourceManager`/`Scheduler`.
- The per-application `ApplicationMaster` (`App Master`) is, in effect, a framework specific library and is tasked with negotiating resources from the ResourceManager and working with the `NodeManager`(s) to execute and monitor the tasks.

YARN – Big picture

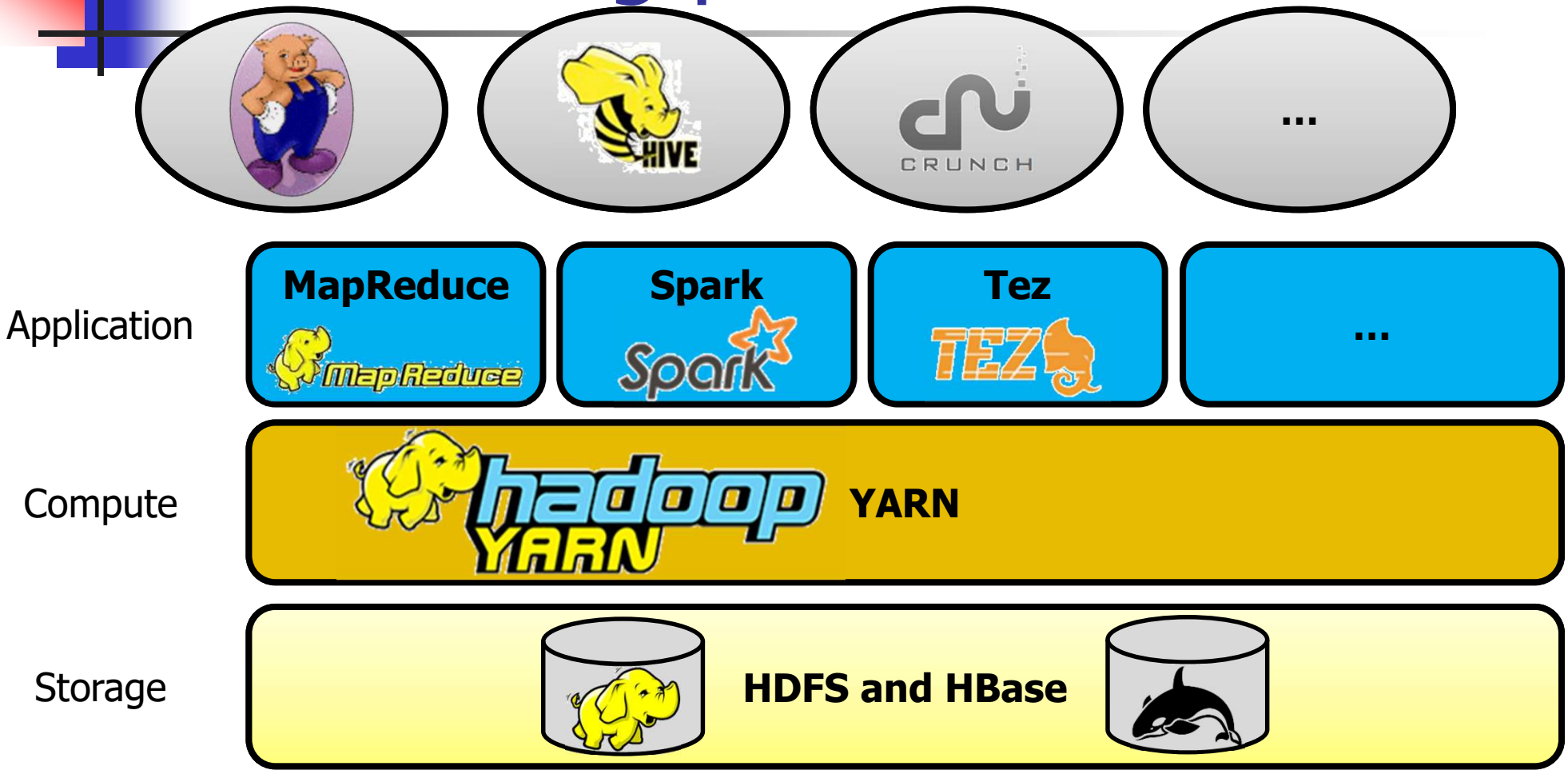


YARN – Big picture



Running an application using YARN (from [1])

YARN – Big picture



What is the one that I choose?

HDFS
Hadoop Distributed File System

- Apache Software foundation
  - https://www.apache.org
  - https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html
- HDFS is a distributed file system designed to run on commodity hardware

Apache HBase™

Distributed, scalable, big data store

- Apache Software foundation
  - https://www.apache.org
  - https://hbase.apache.org/
- Is an open-source, distributed, versioned, non-relational database (NoSQL) modeled after Google's Bigtable
  - https://ai.google/research/pubs/pub27898

Apache Hadoop YARN™

Job scheduling and cluster resource management

- Apache Software foundation
  - https://www.apache.org
  - https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html
- Hadoop's cluster resource management system

MapReduce
Programming model

- Apache Software foundation
  - https://www.apache.org
  - https://hadoop.apache.org/
- The Apache implementation of MapReduce is "Hadoop MapReduce", consisting in a YARN-based system for parallel processing of large data sets
- MapReduce paper
  - https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf

Apache Spark™

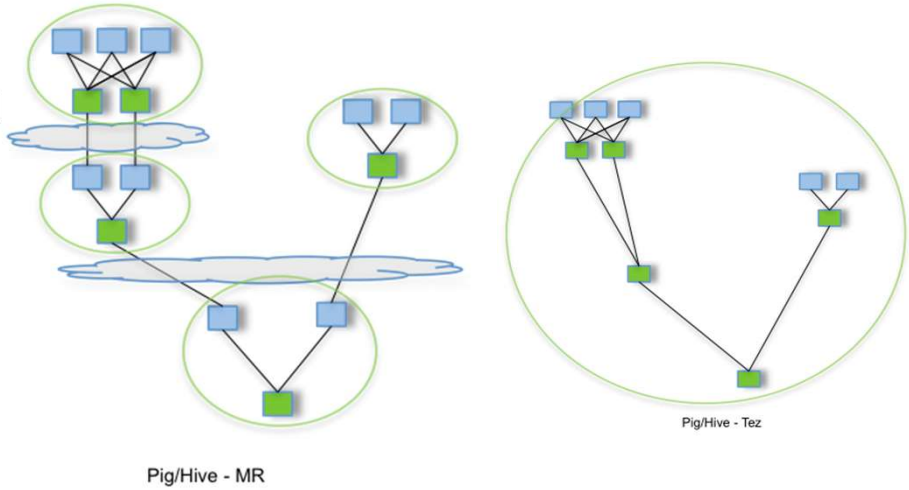
Fast and general compute engine for Hadoop data

- Apache Software foundation
  - https://www.apache.org
  - https://spark.apache.org/
- A unified analytics engine for large-scale data processing. Designed to perform both batch processing (similar to MapReduce) and new workloads like streaming, interactive queries, and machine learning

Apache Tez™

Generalized data-flow programming framework

- Apache Software foundation
  - https://www.apache.org
  - https://tez.apache.org/



- Aimed at building an application framework which allows for a complex Directed Acyclic Graph (DAG) of tasks for processing data. Currently built on top of Apache Hadoop YARN

Apache Pig™
High-level data-flow language

- Apache Software foundation
  - https://www.apache.org
  - https://pig.apache.org/
- Platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs

Apache Hive™
Data warehouse infrastructure

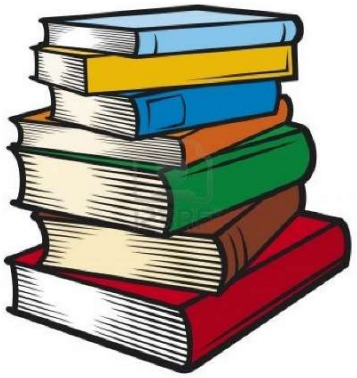
- Apache Software foundation
  - https://www.apache.org
  - https://hive.apache.org/
- A data warehouse software that facilitates reading, writing, and managing large datasets residing in distributed storage using SQL

Apache Crunch™

Simple and efficient MapReduce pipelines

- Apache Software foundation
  - https://www.apache.org
  - https://crunch.apache.org/
- Running on top of Hadoop MapReduce and Apache Spark™ , the Apache Crunch™ library is a simple Java API for tasks like joining and data aggregation that are tedious to implement on plain MapReduce

References



[1] T. White, "Hadoop - The Definitive Guide" 4th Edition", ISBN-13: 9781491901632, ISBN-10: 1491901632

[2] Apache Hadoop, http://hadoop.apache.org/