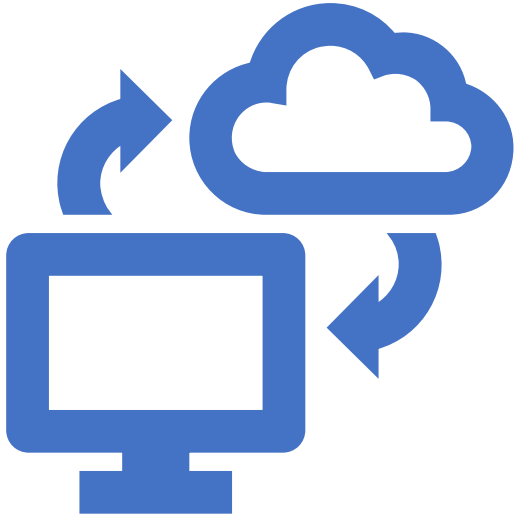
MapReduce Cloud Computing Models & Hadoop

- **ECE 4150, CS 4893**

- **Vijay Madisetti**
- **Fall 2024**
- **Georgia Tech**

MR – A Model of Computation Developed for Cloud

- We did not specifically look at new models of computation in earlier models discussed in class.
- Today, we will look at a new model of computing of that is specially adapted to distributed cloud computing –>> MR



Types of Logs Collected by Enterprises

- **Hardware & Networking Devices** – servers, routers, switches, bridges, hubs, firewalls, etc. – often distributed across multiple datacenters
- **Operating Systems** – Windows & Unix based operating systems event logs in various levels of information detail, containing registry, event, file system, Syslog, SNMP, NetFlow and other types of logs.
- **Database Systems** – transaction, query, recovery, configuration and audit log
- **Applications** – HTTP weblogs, FTP logs, Log4J, JMS, JMX, Net events, Portal logs, but mainly the core business applications that often have their own homegrown log formats
- **Customer Facing Data** – logs reporting on product usage, click stream data, shopping card logs, online transaction logs
- **Virtualization & Cloud** – hypervisor logs, VM logs, cloud app logs
- **Outside Datacentre** – Manufacturing, logistics logs, CDRs & IPDRs, power consumption, RFID or GPS data logs

https://www.youtube.com/watch?v=BPC_mClNSXk&feature=emb_logo

Big Data Sets are Very Large

Example: 20+ billion web pages x 20KB = 400+ terabytes

- One computer can read 30-35 MB/sec from disk

  ~four months to read the web

- ~1,000 hard drives just to store the web
- Even more to *do* something with the data

Parallel Computing

Good news: same problem with 1000 machines, < 3 hours

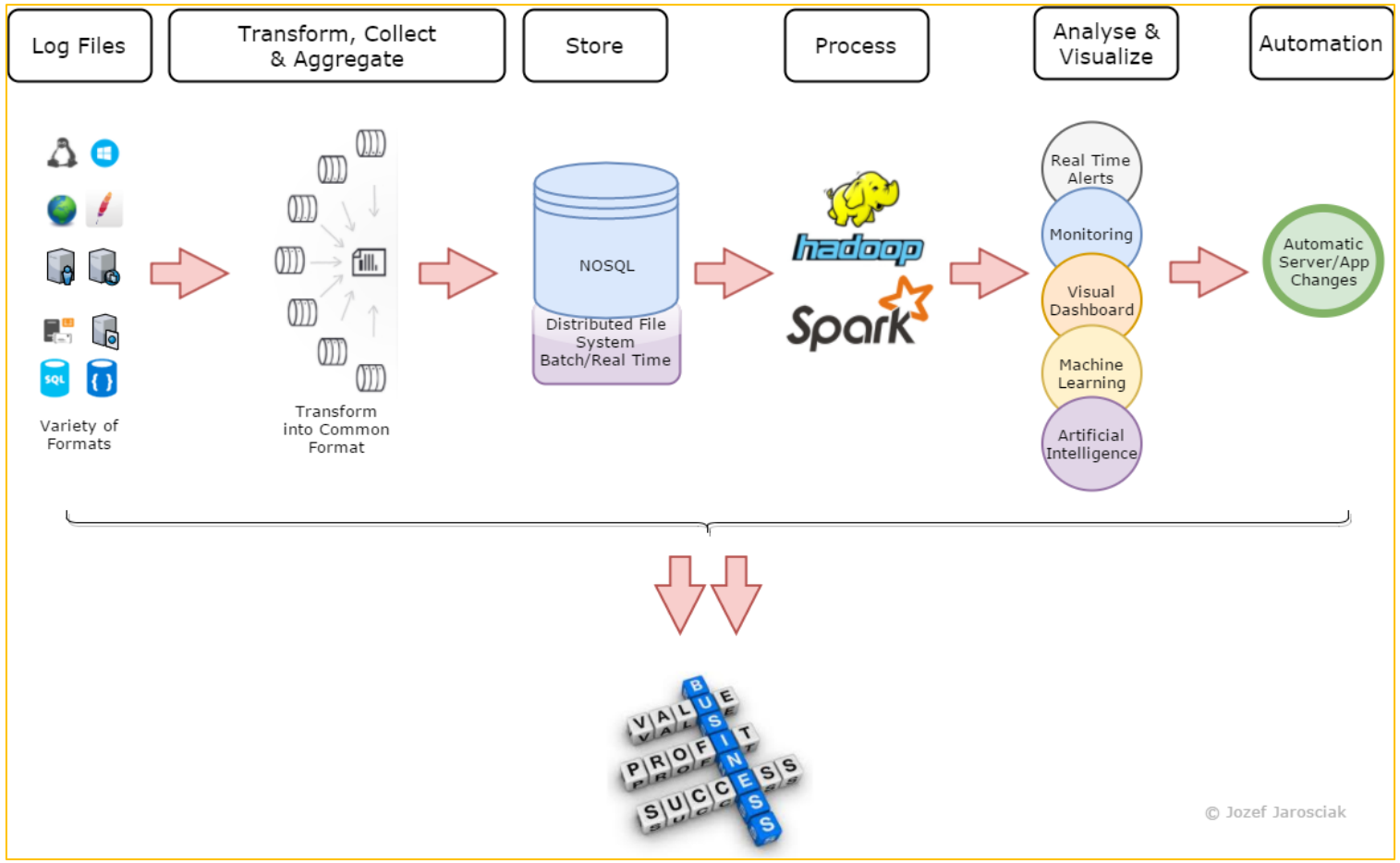
Bad news: programming work

- communication and coordination
- recovering from machine failure (all the time!)
- status reporting
- debugging
- optimization
- locality

Bad news II: repeat for every problem you want to solve

How can we make this easier?

Enterprise Data Analytics Flow



What is MapReduce MR ?

A simple programming model that applies to many large-scale computing problems

Hide messy details in MapReduce runtime library:

- automatic parallelization
- load balancing
- network and disk transfer optimization
- handling of machine failures
- robustness
- **improvements to core library benefit all users of library!**

MR approach

Read a lot of data

Map: extract something you care about from each record

Shuffle and Sort

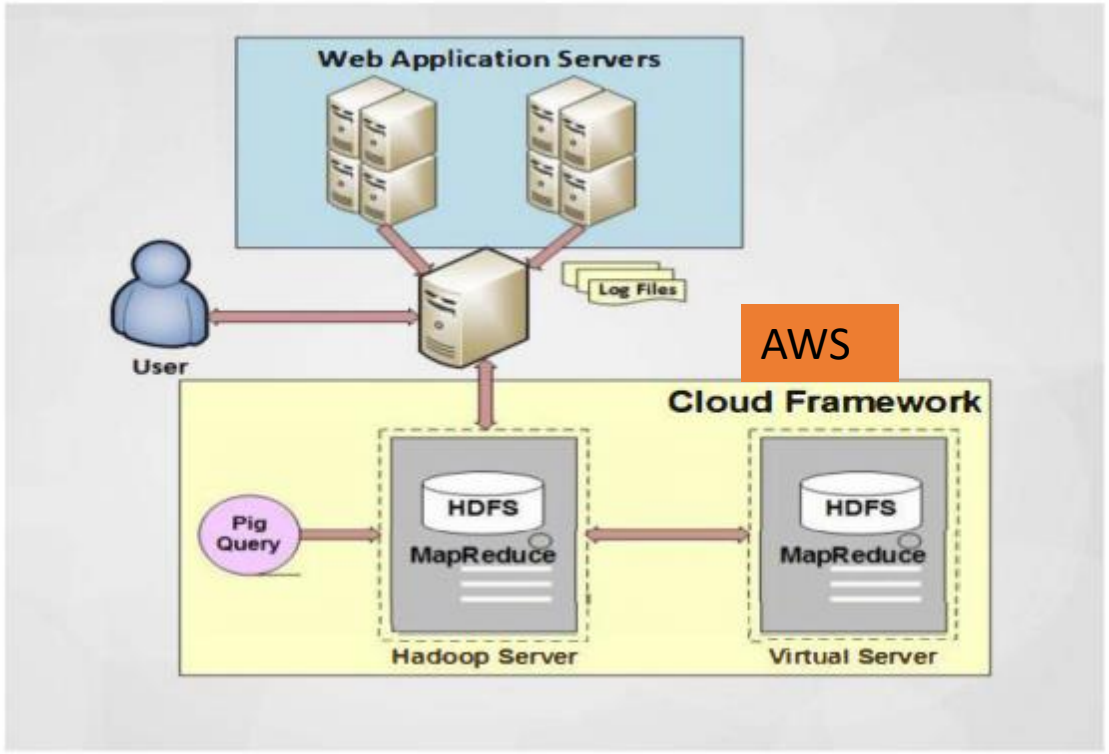
Reduce: aggregate, summarize, filter, or transform

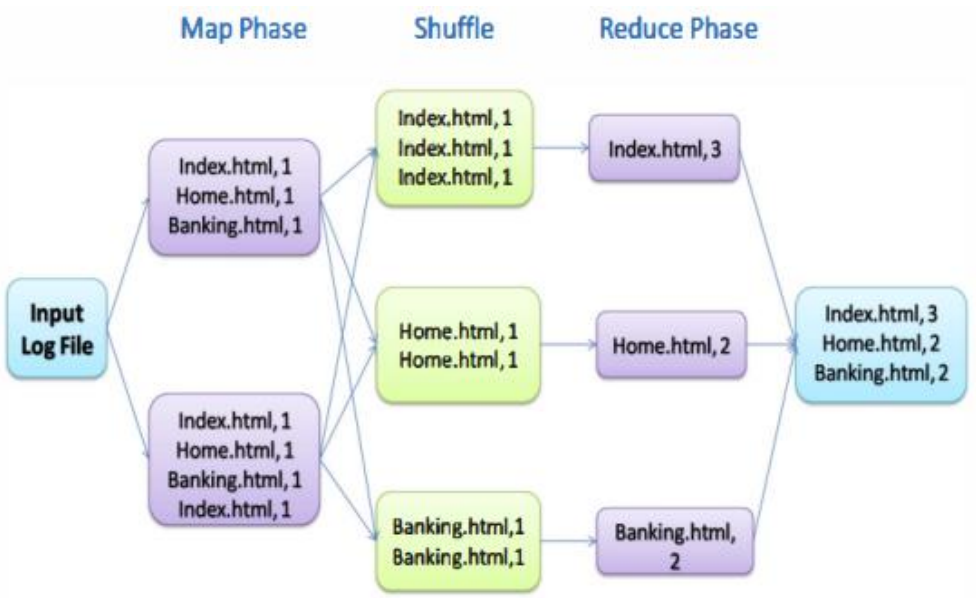
Write the results

Outline stays the same,

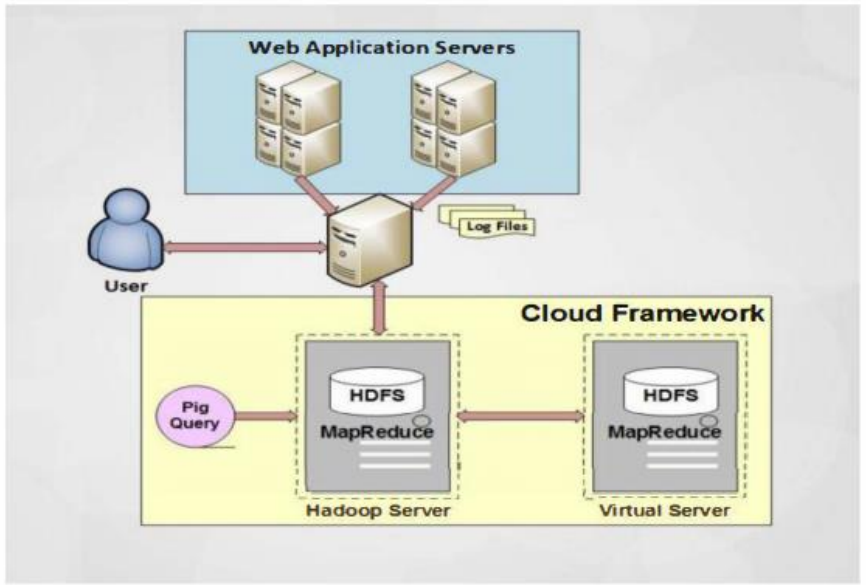
Map and Reduce change to fit the problem

How Does MR work?





Hadoop/MR in Website Logs Analytics



INPUT
Web Log

Cnn.com
Cnn.com
Nytimes.com
Wsj.com
Cnn.com
Wsj.com
Nbc.com
Ft.com
Wsj.com
Cnn.com

MAP

<cnn.com, 1>
<cnn.com, 1>
<nytimes.com, 1>
<wsj.com, 1>
<cnn.com ,1>
<wsj.com, 1>
<nbc.com, 1>
<ft.com, 1>
<wsj.com, 1>
<cnn.com, 1>

Shuffle

REDUCE

<cnn.com,1>
<cnn.com,1>
<cnn.com,1>
<cnn.com,1>

<wsj.com, 1>
<wsj.com, 1>
<wsj.com, 1>

<ft.com, 1>

<nbc.com, 1>

OUTPUT

<cnn.com, 4>

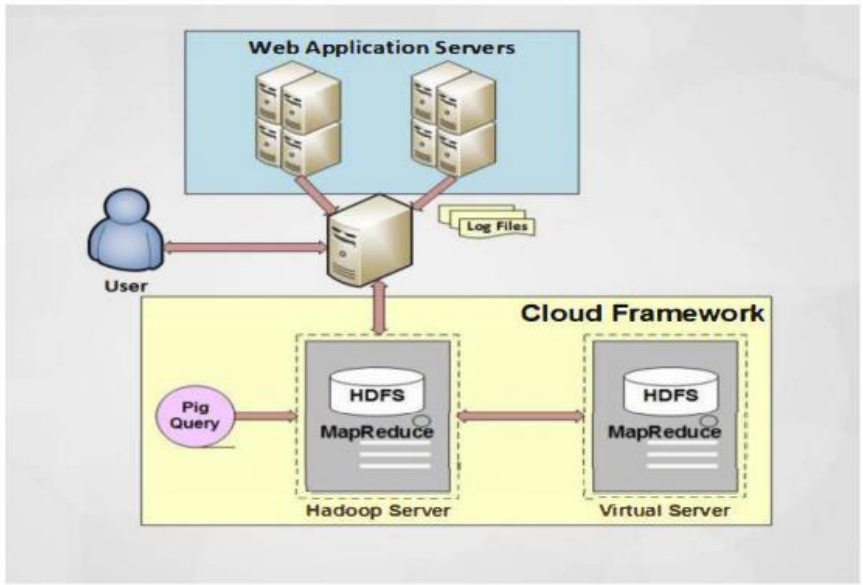
<wsj.com, 3>

<ft.com, 1>

<nbc.com, 1>

Speedup by 3

Hadoop/MR in Website Logs Analytics



INPUT
Web Log

Cnn.com
Cnn.com
Nytimes.com
Wsj.com
Cnn.com
Wsj.com
Nbc.com
Ft.com
Wsj.com
Cnn.com

MAP

REDUCE

OUTPUT

Node 1

Node 3

Node 2

Shuffle

Speedup by 3

MR Template

Basic data type: the key-value pair (k,v).

For example, key = URL, value = HTML of the web page.

Programmer specifies two primary methods:

- Map: $(k, v) \mapsto \langle (k_1,v_1), (k_2,v_2), (k_3,v_3),\ldots,(k_n,v_n) \rangle$
- Reduce: $(k', \langle v'_1, v'_2,\ldots,v'_{n'} \rangle) \mapsto \langle (k', v''_1), (k', v''_2),\ldots,(k', v''_{n''}) \rangle$

All v' with same k' are reduced together.

(Remember the invisible “Shuffle and Sort” step.)

Typical Training Exercise at Google

A typical exercise for a new Google engineer in his or her first week

Input: files with one document per record

Specify a *map* function that takes a key/value pair
key = document URL
value = document contents

Output of map function is (potentially many) key/value pairs.
In our case, output (word, "1") once per word in the document

"document1", "to be or not to be"

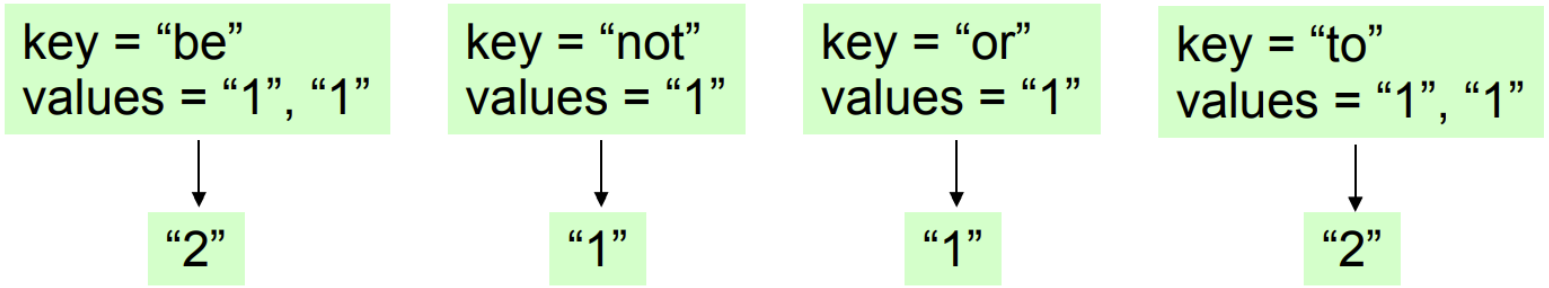
↓

"to", "1"
"be", "1"
"or", "1"

Word Frequencies in a Web Page

MapReduce library gathers together all pairs with the same key (shuffle/sort)

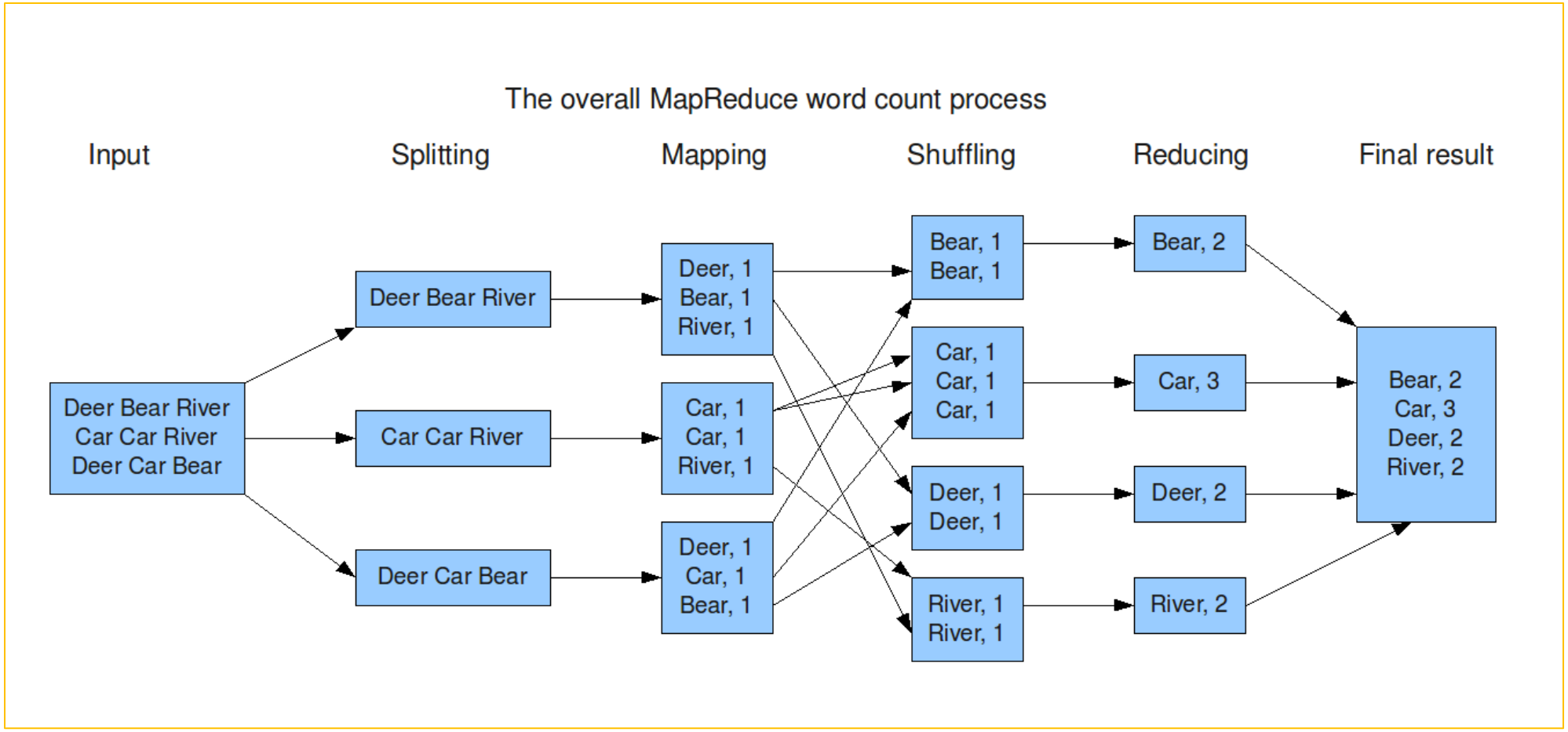
Specify a *reduce* function that combines the values for a key
In our case, compute the sum



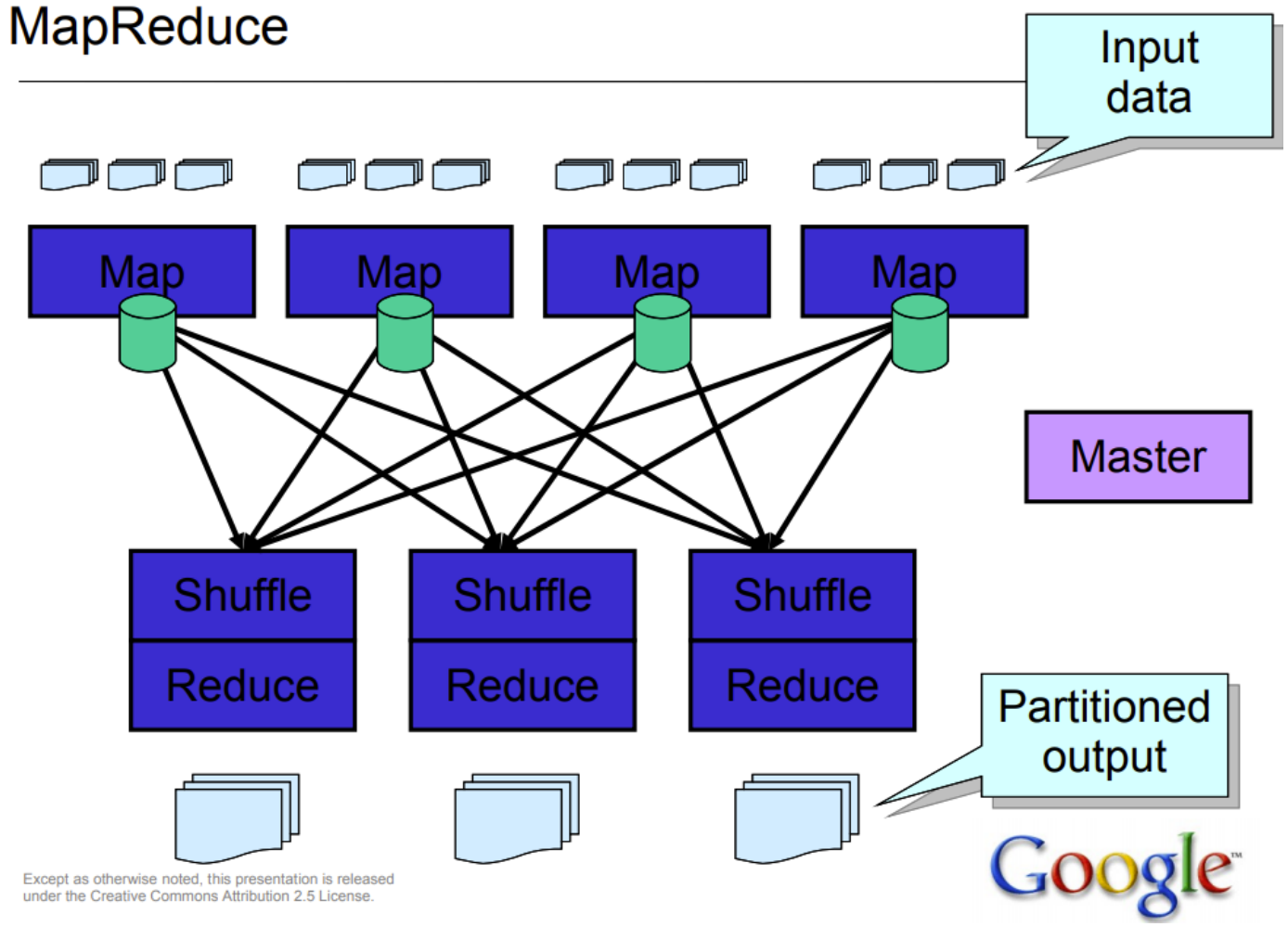
Output of reduce (usually 0 or 1 value) paired with key and saved

“be”, “2”
“not”, “1”
“or”, “1”

MapReduce: Word Count



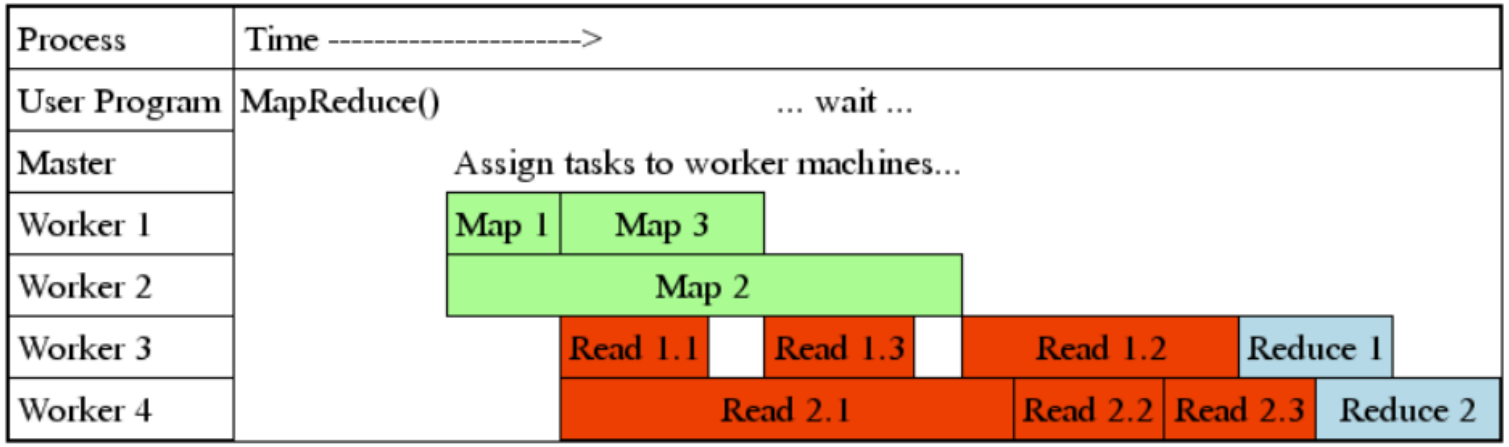
MapReduce model at Google



MapReduce Model at Google

Fine granularity tasks: many more map tasks than machines

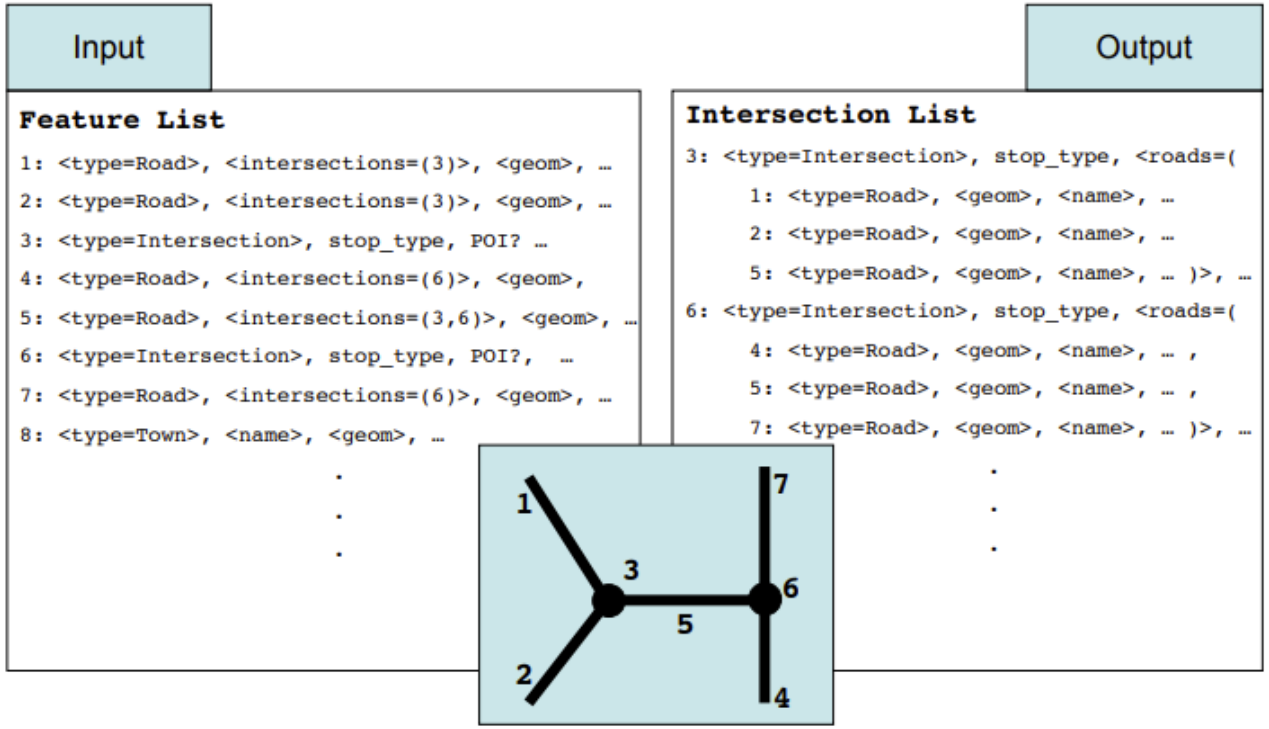
- Minimizes time for fault recovery
- Can pipeline shuffling with map execution
- Better dynamic load balancing



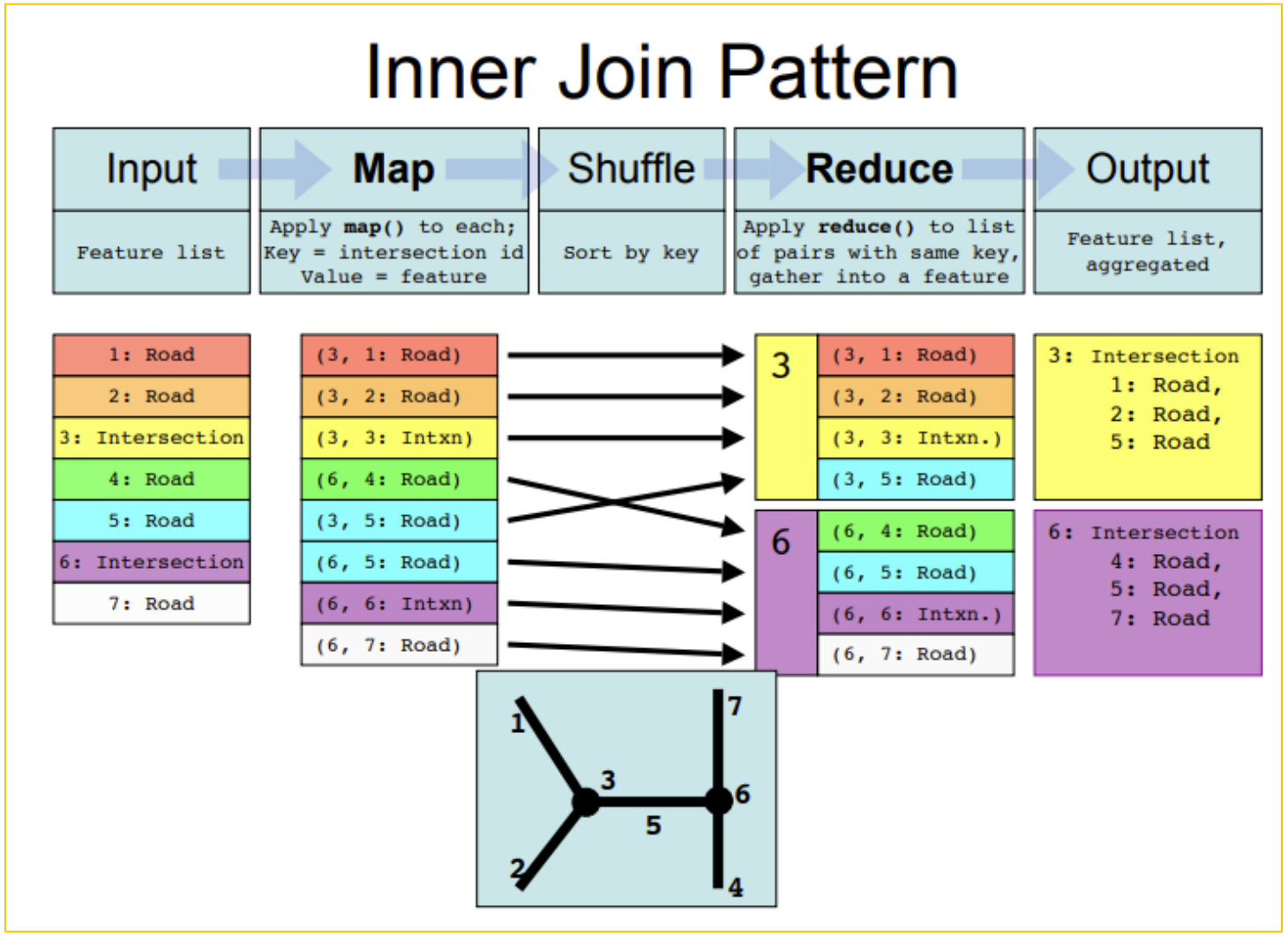
Google™

Google Maps using MR

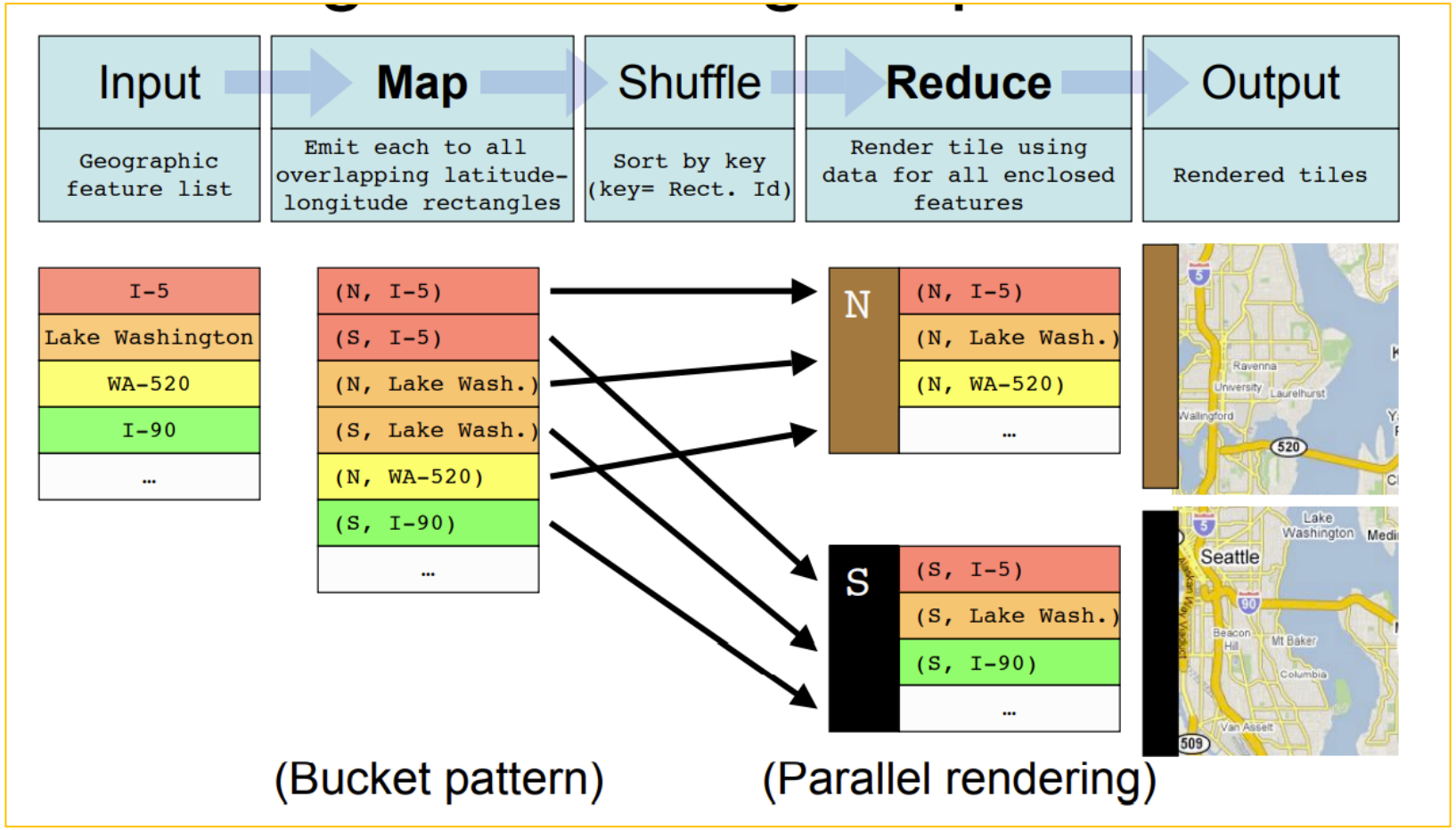
Pointer Following (or) Joining



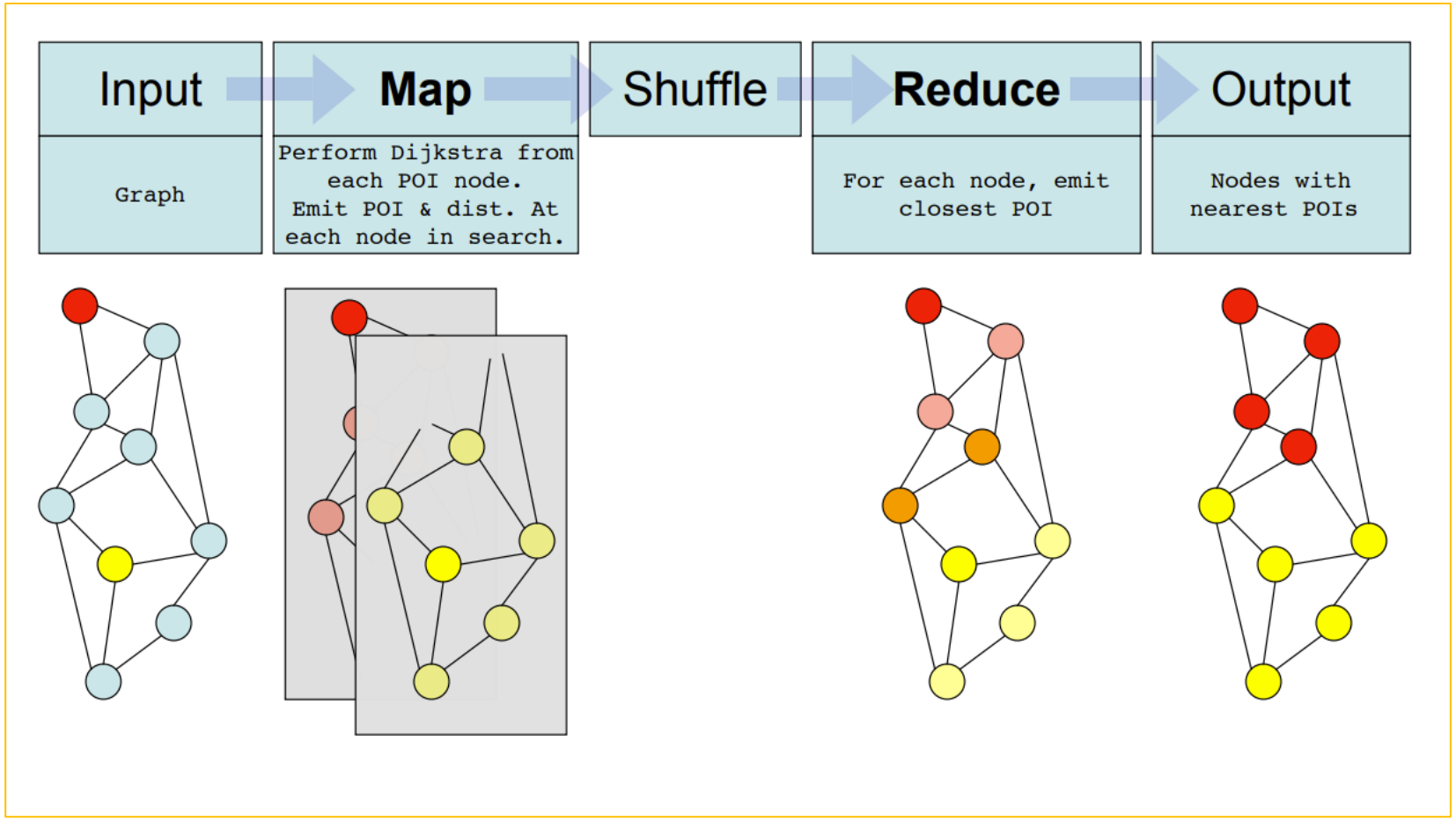
MR in Google Maps

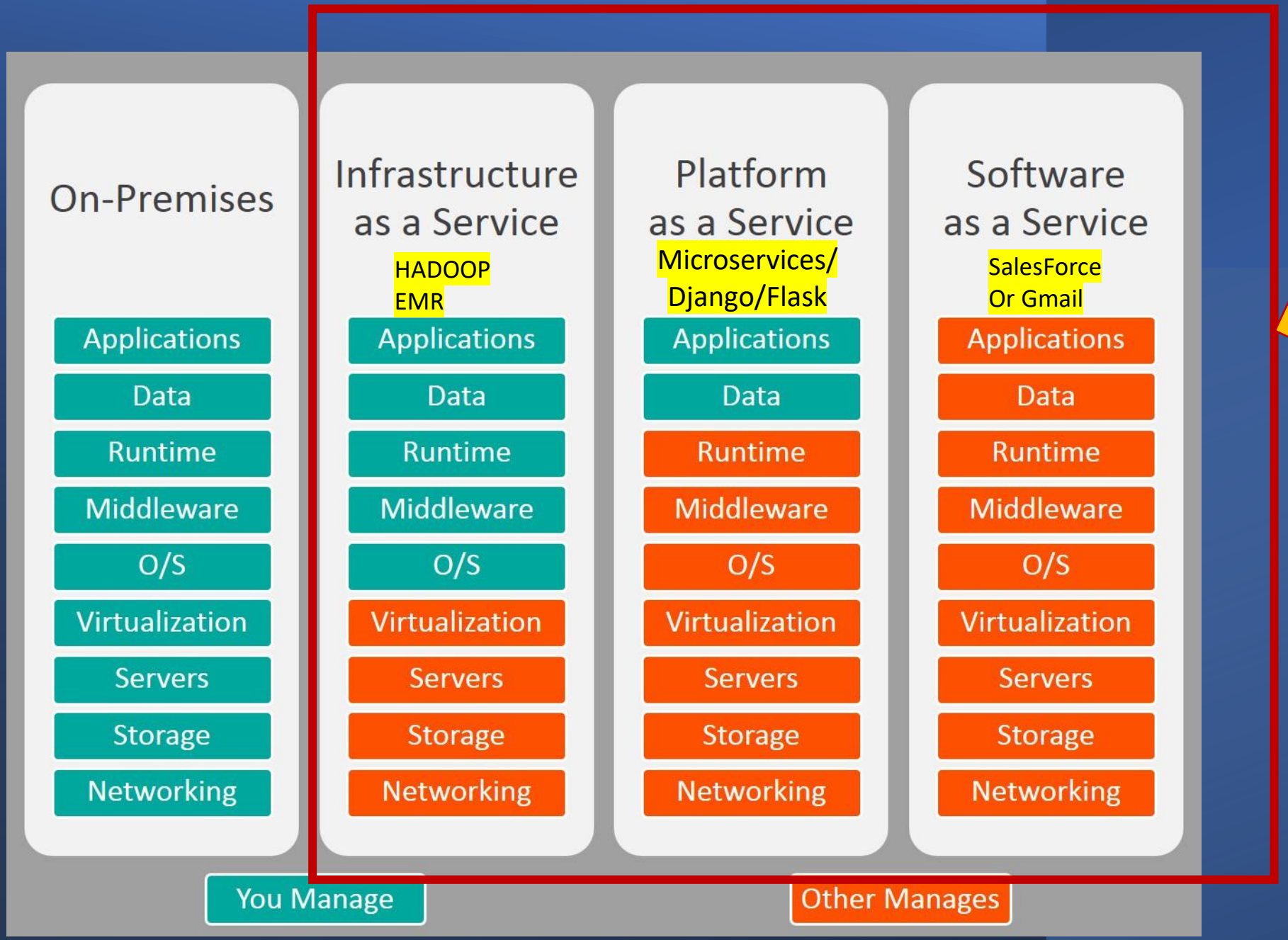


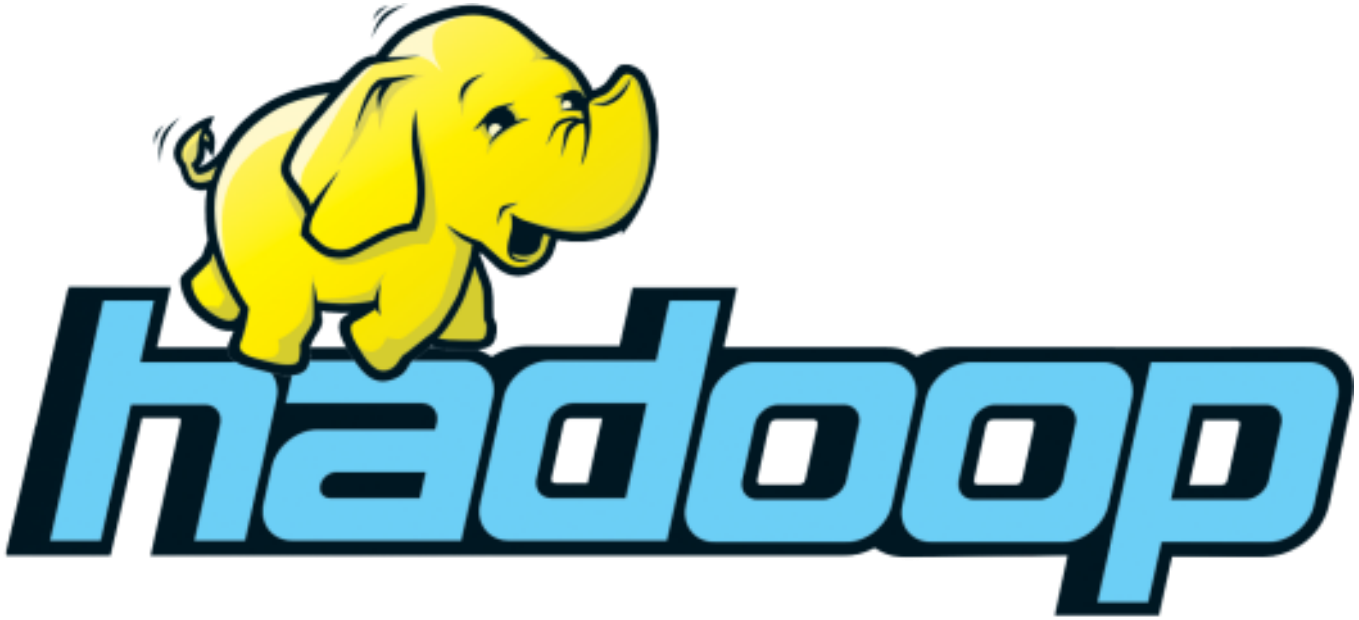
Distributing MR across map tiles



Finding Point of Interest (POI) in Google Maps




On-Premises
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
Infrastructure as a Service
HADOOP EMR
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
Platform as a Service
Microservices/ Django/Flask
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
Software as a Service
SalesForce Or Gmail
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
You Manage
Other Manages
Cloud Models


hadoop

Hadoop

- Based on work done by Google in the early 2000s
  - “The Google File System” in 2003
  - “MapReduce: Simplified Data Processing on Large Clusters” in 2004
- The core idea was to distribute the data as it is initially stored
  - Each node can then perform computation on the data it stores without moving the data for the initial processing

Core Hadoop Concepts

- Applications are written in a high-level programming language
  - No network programming or temporal dependency
- Nodes should communicate as little as possible
  - A “shared nothing” architecture
- Data is spread among the machines in advance
  - Perform computation where the data is already stored as often as possible

Hadoop High-Level Overview

- When data is loaded onto the system it is divided into blocks
  - Typically , 64MB or 128MB
- Tasks are divided into two phases
  - Map tasks which are done on small portions of data where the data is stored
  - Reduce tasks which combine data to produce the final output
- A master program allocates work to individual nodes

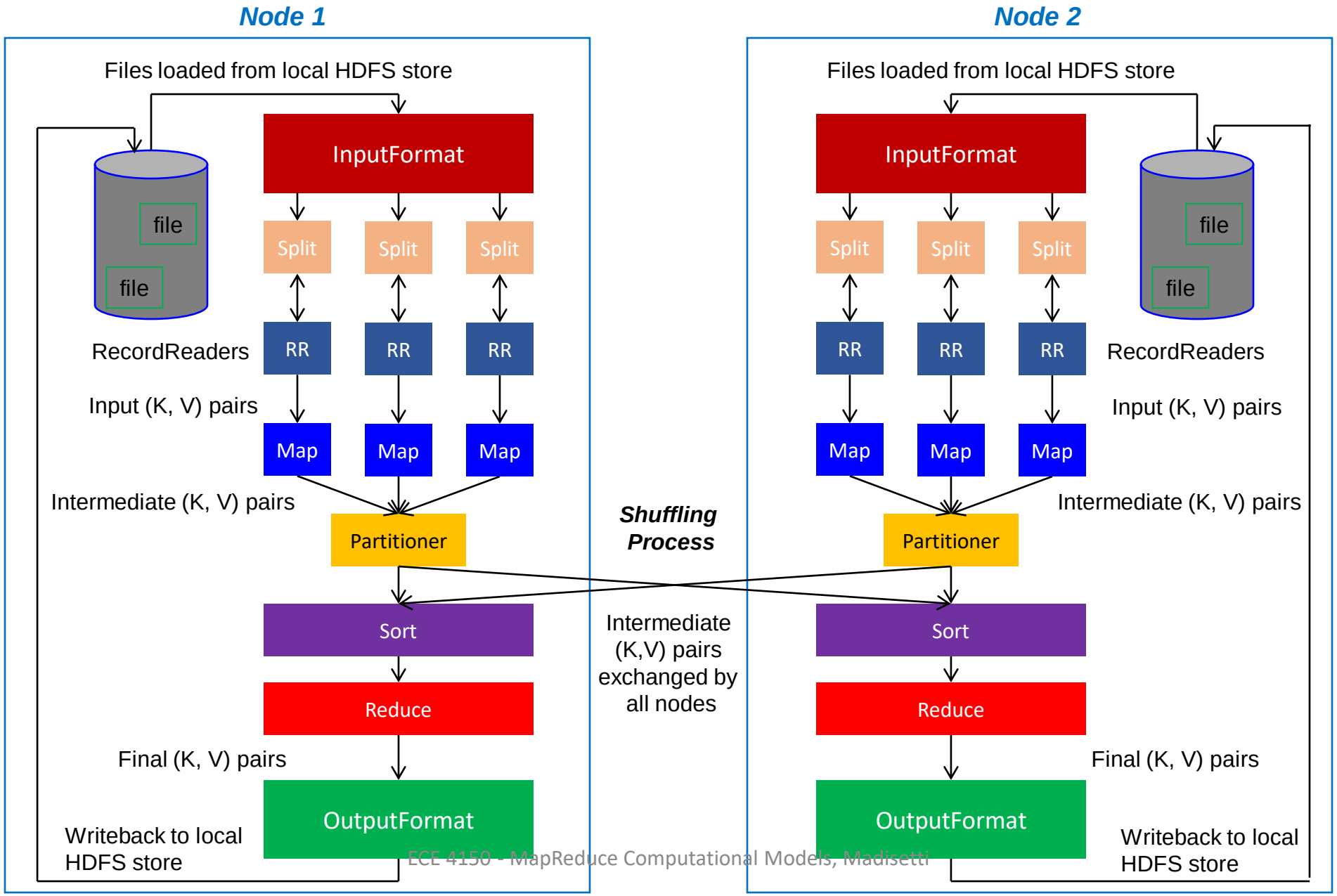
Hadoop Overview

Responsible for storing data on the cluster

Data files are split into blocks and distributed across the nodes in the cluster

Each block is replicated multiple times

Hadoop MapReduce: A Closer Look



HDFS Basic Concepts

- HDFS is a file system written in Java based on the Google's GFS
- Provides redundant storage for massive amounts of data

MRJob Example for Log Analysis

Input

```
root@Acer-VKM: ~/vkm
0,Will,33,385
1,Jean-Luc,26,2
2,Hugh,55,221
3,Deanna,40,465
4,Quark,68,21
5,Weyoun,59,318
6,Gowron,37,220
7,Will,54,307
8,Jadzia,38,380
9,Hugh,27,181
10,Odo,53,191
11,Ben,57,372
12,Keiko,54,253
13,Jean-Luc,56,444
14,Hugh,43,49
15,Rom,36,49
16,Weyoun,22,323
17,Odo,35,13
18,Jean-Luc,45,455
19,Geordi,60,246
20,Odo,67,220
--More--(3%)
```

<Index, Name, Age, Number of Friends>

MR Job

```
root@Acer-VKM:~/vkm# more Friends-By-Age-2.py
from mrjob.job import MRJob

class MRFriendsByAge(MRJob):

    def mapper(self, _, line):
        (ID, name, age, numFriends) = line.split(',')
        yield age, float(numFriends)

    def reducer(self, age, numFriends):
        total = 0
        numElements = 0
        for x in numFriends:
            total += x
            numElements += 1

        yield age, total / numElements


if __name__ == '__main__':
    MRFriendsByAge.run()
```

Output

```
root@Acer-VKM: ~/vkm
Streaming final output from /tmp/Friends-By-Ag
...
"18"    343.375
"19"    213.27272727272728
"20"    165.0
"21"    350.875
"22"    206.42857142857142
"23"    246.3
"24"    233.8
"25"    197.45454545454547
"26"    242.05882352941177
"27"    228.125
"28"    209.1
"29"    215.91666666666666
"30"    235.8181818181818
"31"    267.25
"32"    207.9090909090909
"33"    325.3333333333333
"34"    245.5
"35"    211.625
"36"    246.6
"37"    249.33333333333334
```

<Age, Number of Friends>

AWS MapReduce Support from 2009

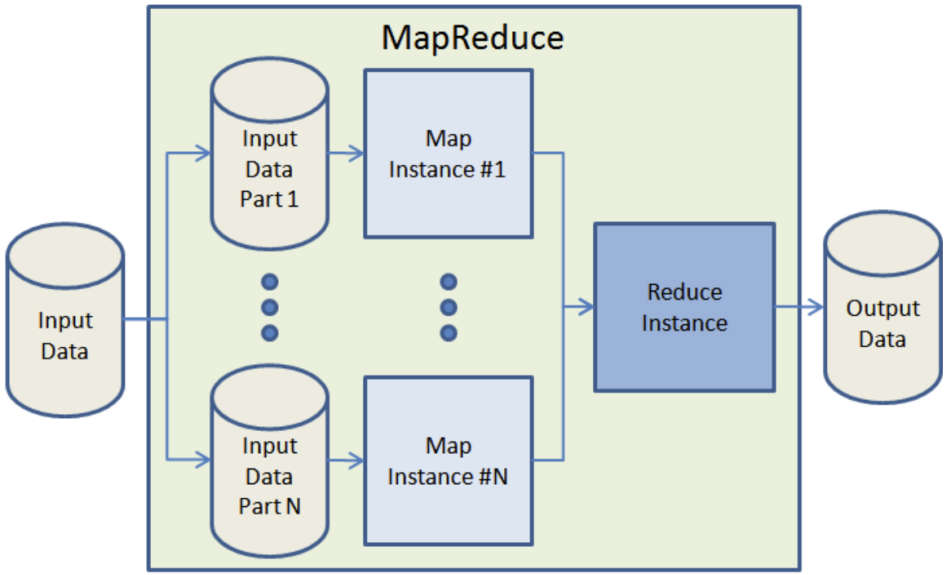
aws

Contact Sales Support My Account Create an AWS Acc

Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Customer Enablement Events Explore More

Blog Home Category Edition Follow Search Blogs

Over the past 3 or 4 years, scientists, researchers, and commercial developers have recognized and embraced the MapReduce programming model. Originally described in a landmark paper, the MapReduce model is ideal for processing large data sets on a cluster of processors. It is easy to scale up a MapReduce application to jobs of arbitrary size by simply adding more compute power. Here's a very simple overview of the data flow in a typical MapReduce job:



Follow

Twitter
Facebook
LinkedIn
Twitch
RSS Feed
Email Updates

AWS Pi Week 2021
Join us during the week of March 14th for live events celebrating 15 years of the AWS Cloud
Register now »

[Announcing Amazon Elastic MapReduce | AWS News Blog](#) in 2009

Hadoop Architecture for Cloud

ECE 4150

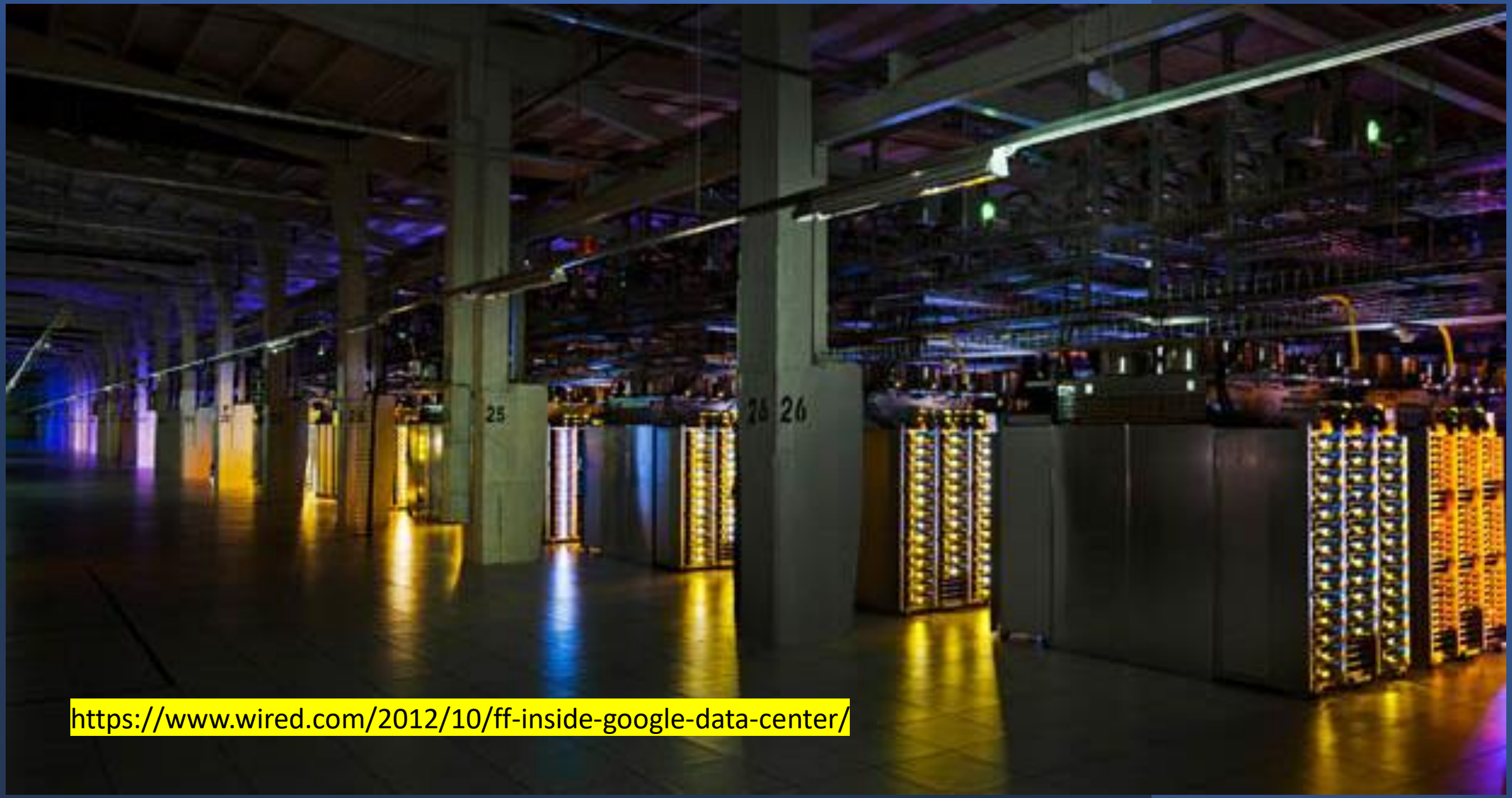
Vijay Madisetti

How does the cloud computer look like ?
Central Ideas of Hadoop Ecosystem

DON'T TELL ME DETAILS!!

I DON'T CARE!!

https://www.wired.com/2012/10/ff-inside-google-data-center/



https://www.wired.com/2012/10/ff-inside-google-data-center/


Microsoft
Azure


Air

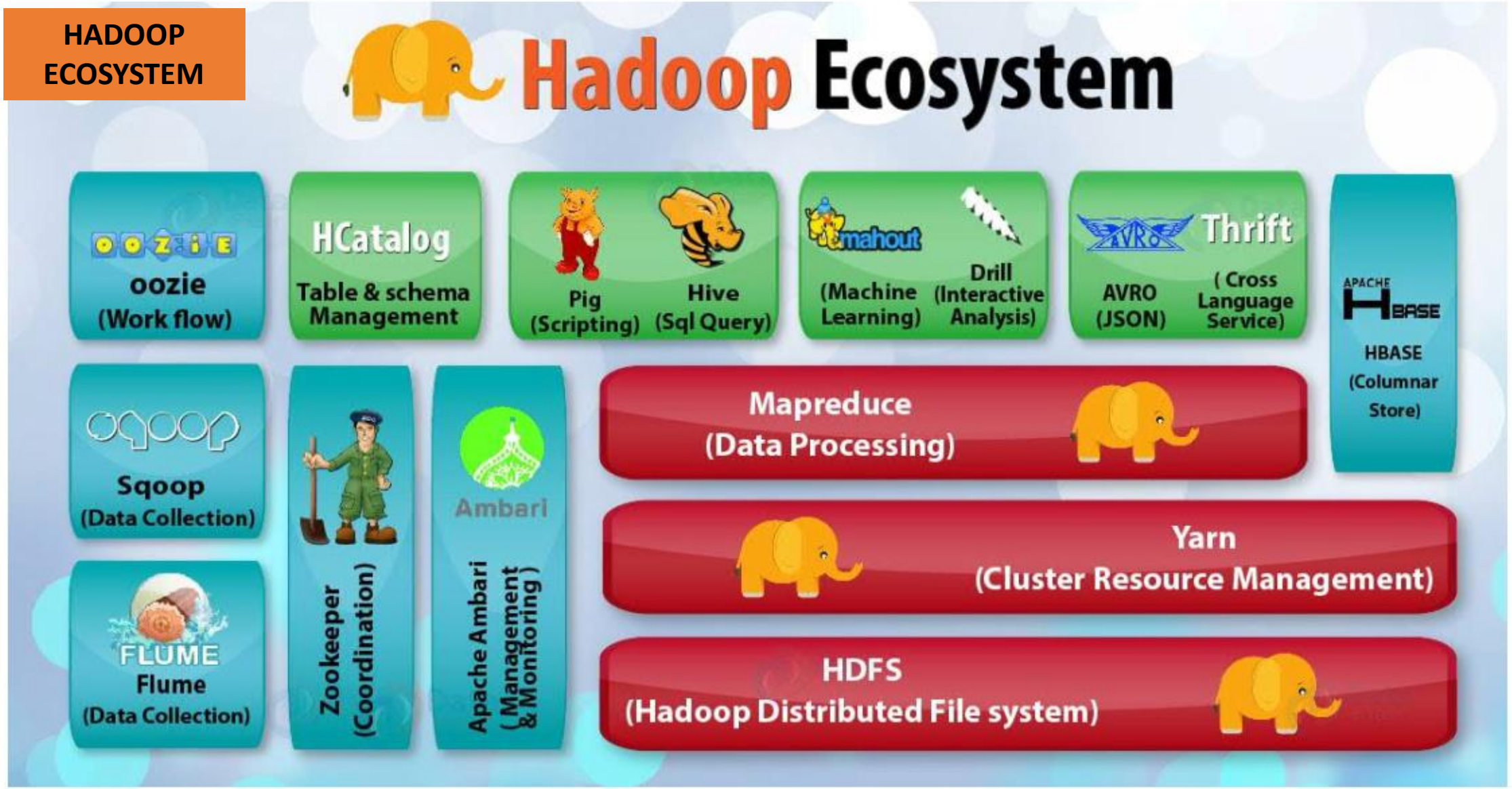

SIEMENS
Water

Interesting videos on data centers

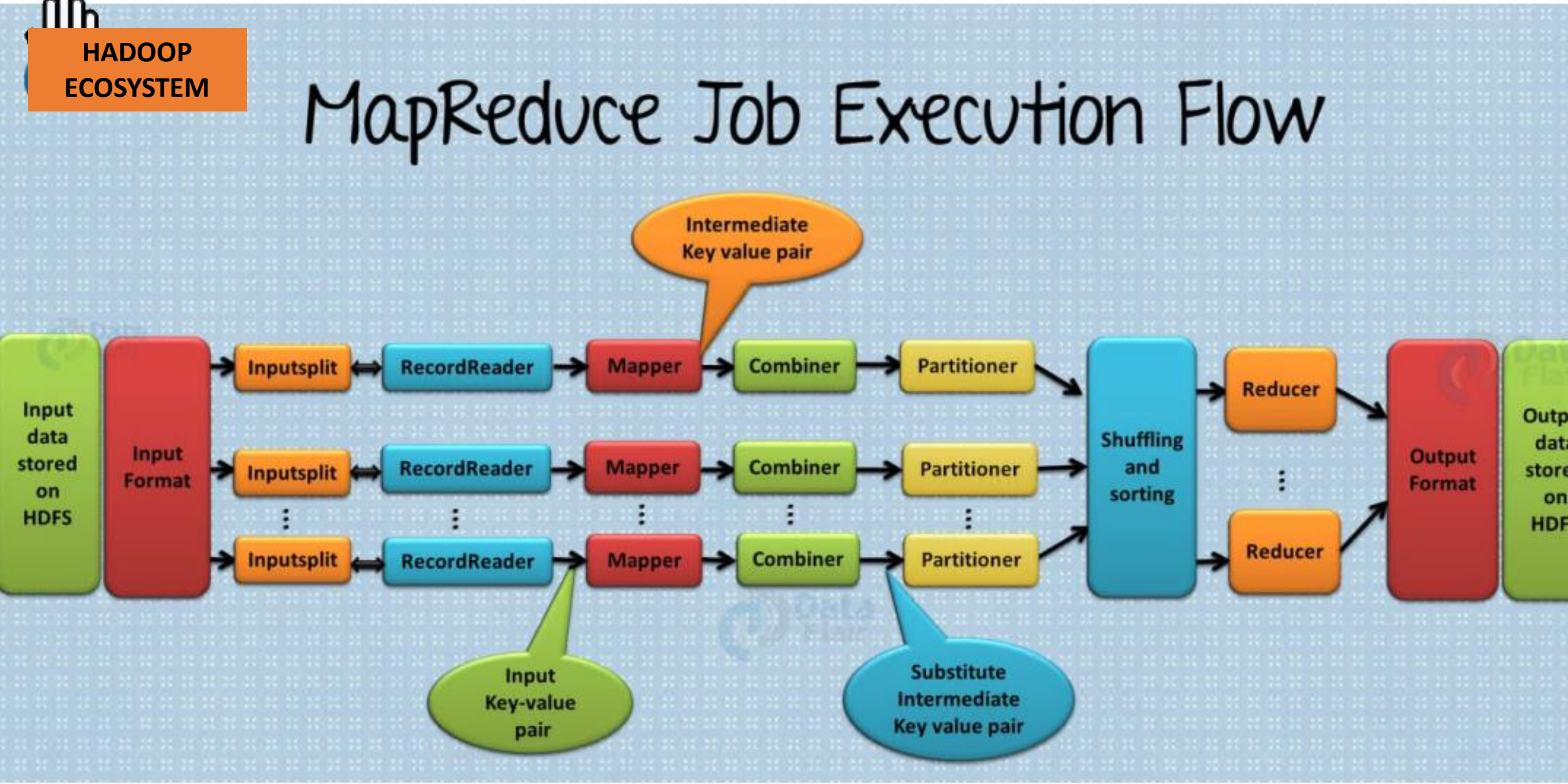
- Microsoft GFS Datacenter Tour (Youtube)
  - http://www.youtube.com/watch?v=hOxA1l1pQIw

- Timelapse of a Datacenter Construction on the Inside (Fortune 500 company)
  - http://www.youtube.com/watch?v=ujO-xNvXj3g

HADOOP ECOSYSTEM

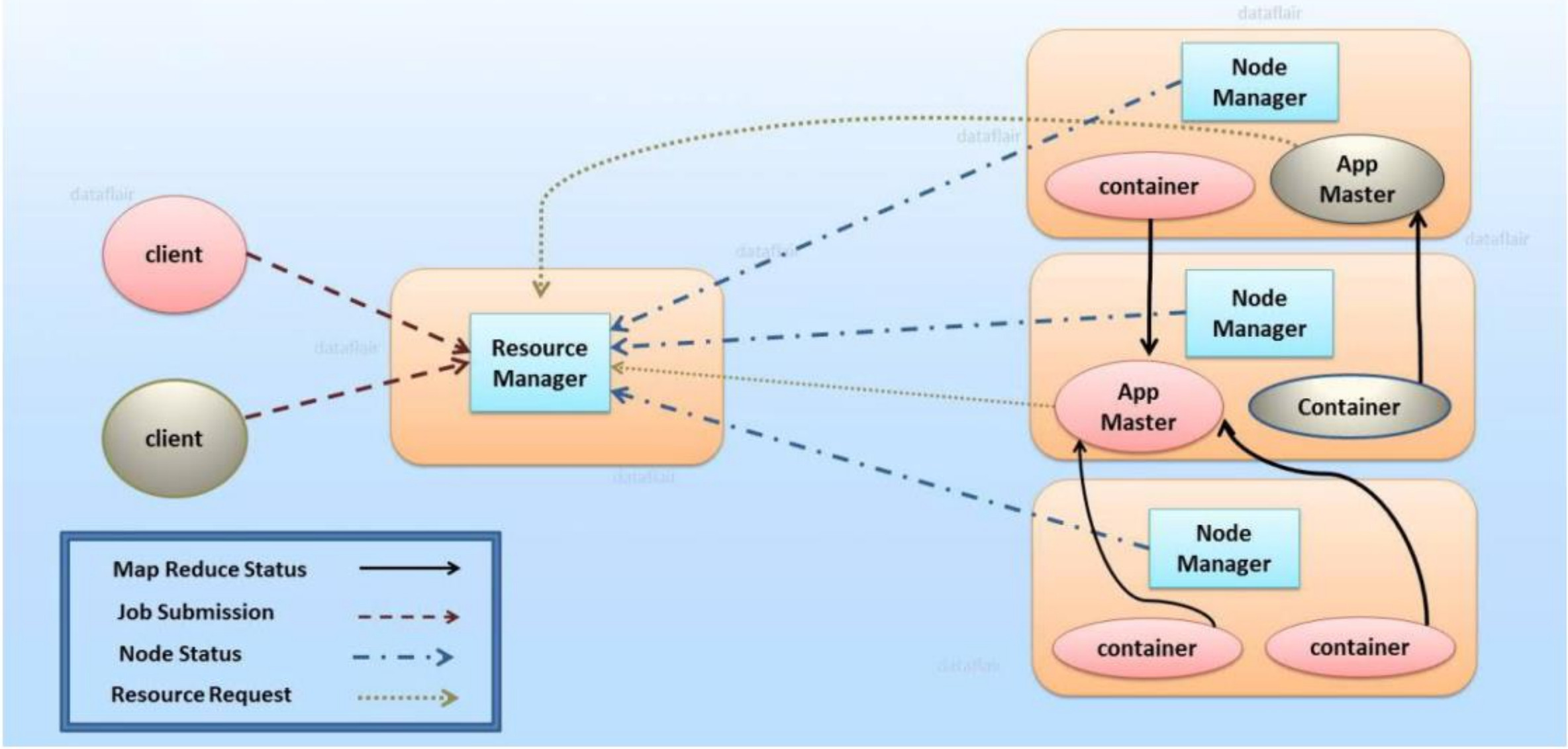


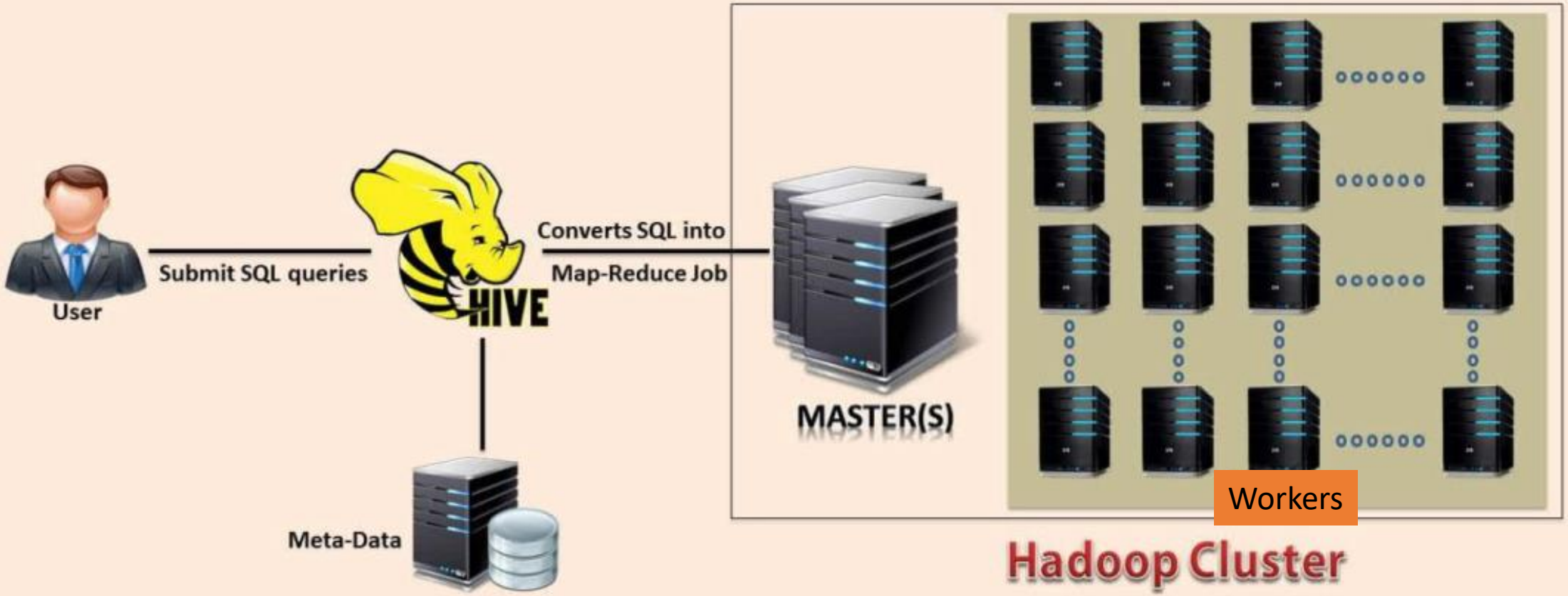
Hadoop Ecosystem and Their Components



Hadoop MapReduce

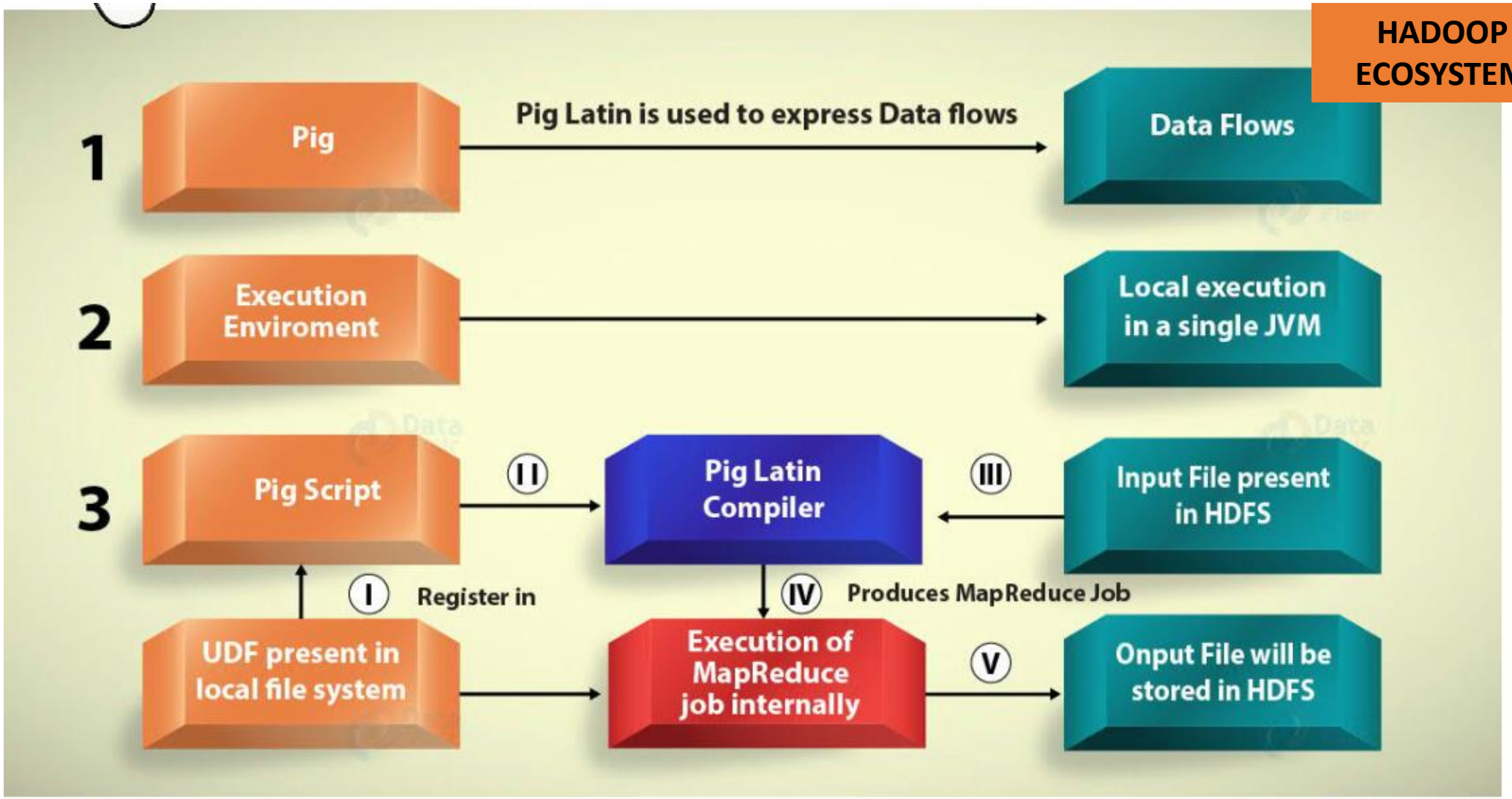
Apache Hadoop YARN-Architecture





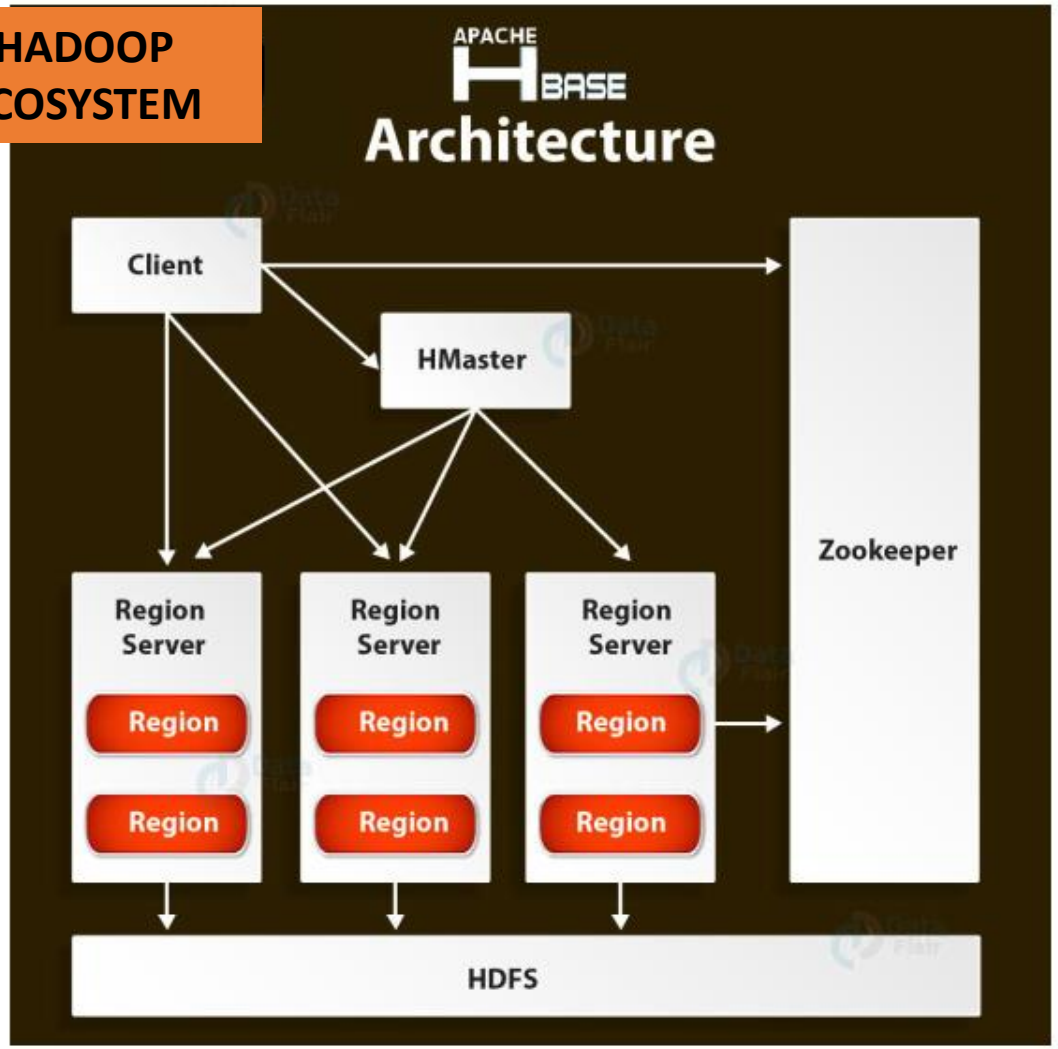
HADOOP ECOSYSTEM

Hive Diagram


HADOOP ECOSYSTEM
1
Pig
Pig Latin is used to express Data flows
Data Flows
2
Execution Enviroment
Local execution in a single JVM
3
Pig Script
II
Pig Latin Compiler
III
Input File present in HDFS
I
Register in
IV
Produces MapReduce Job
UDF present in local file system
Execution of MapReduce job internally
V
Onput File will be stored in HDFS

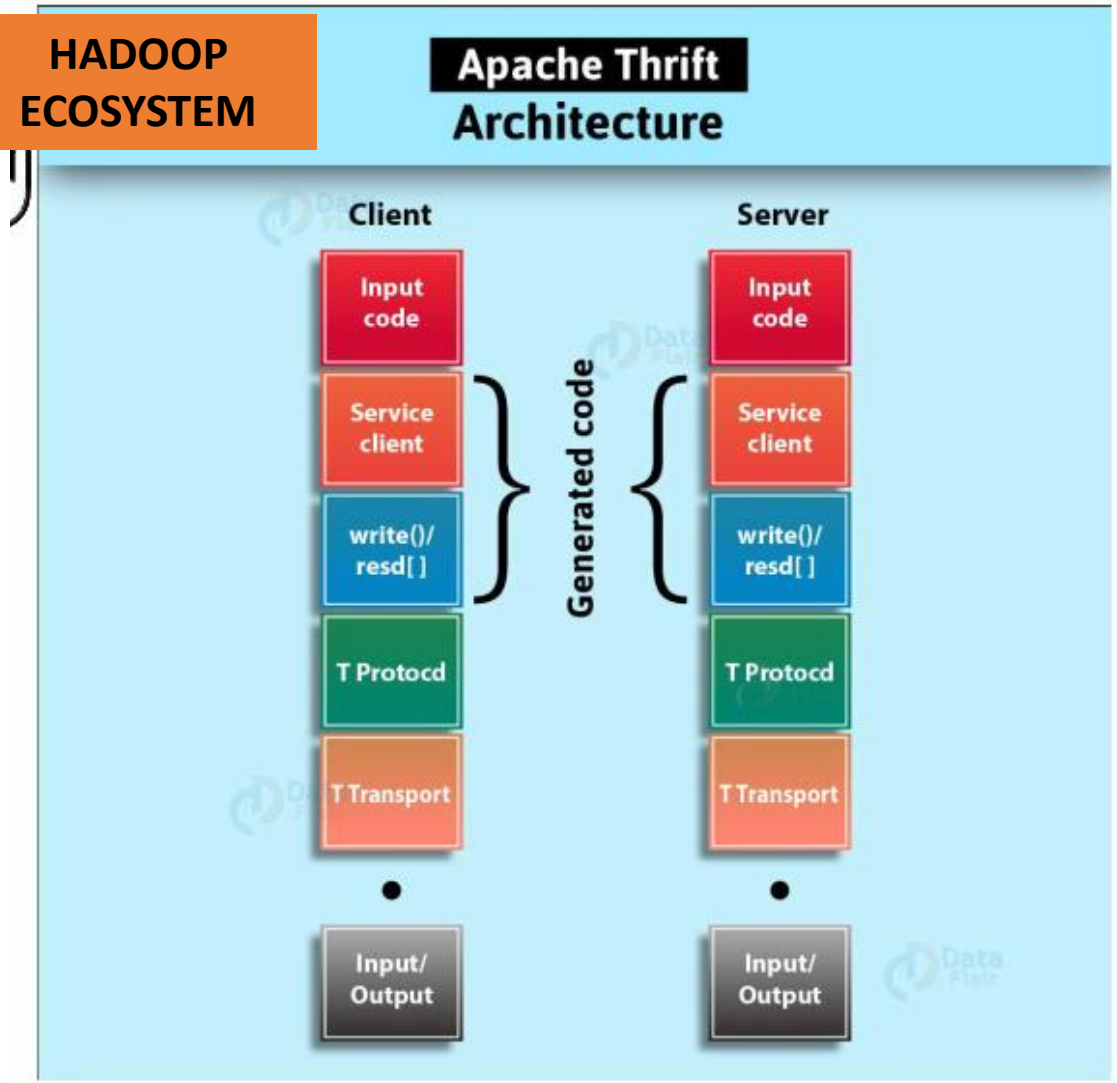
Pig Diagram

HADOOP ECOSYSTEM



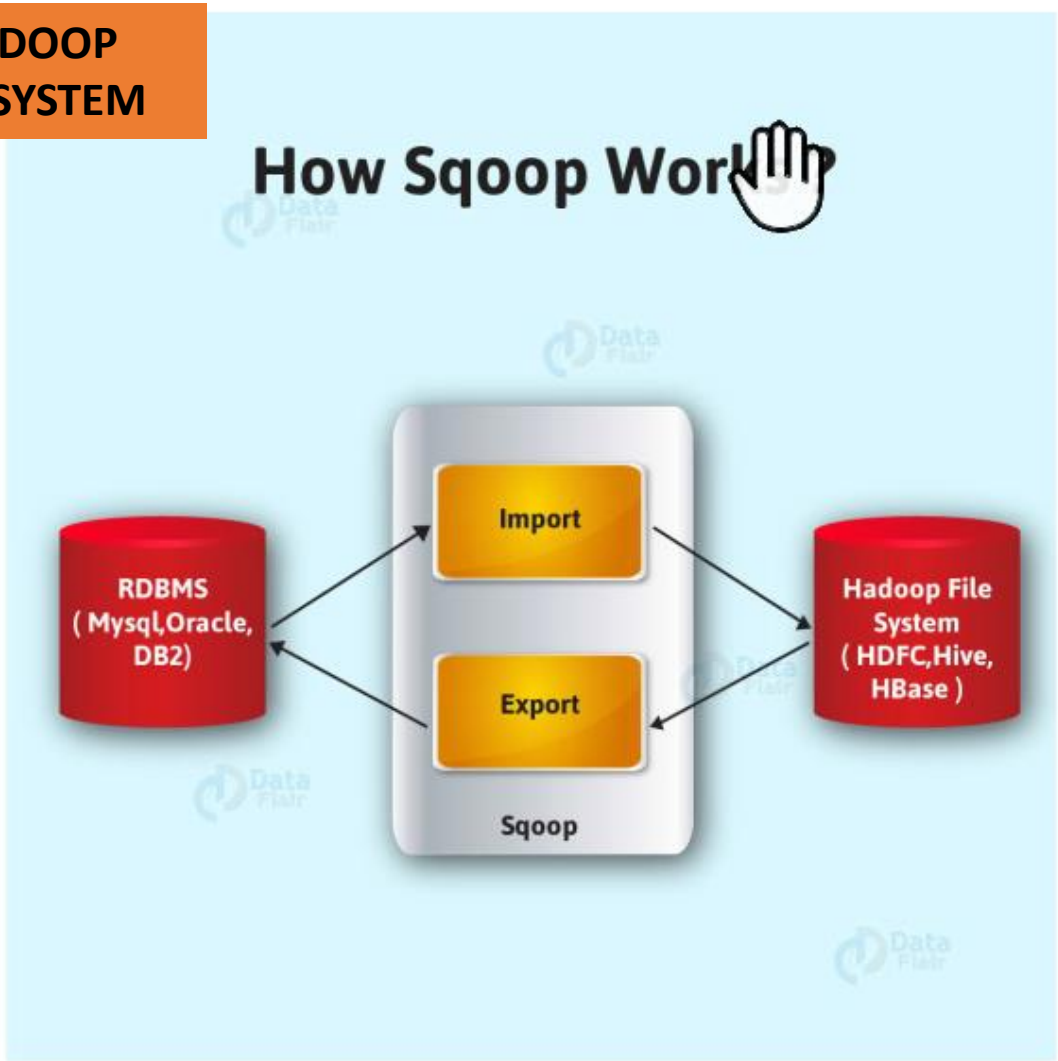
HBase Diagram

HADOOP ECOSYSTEM

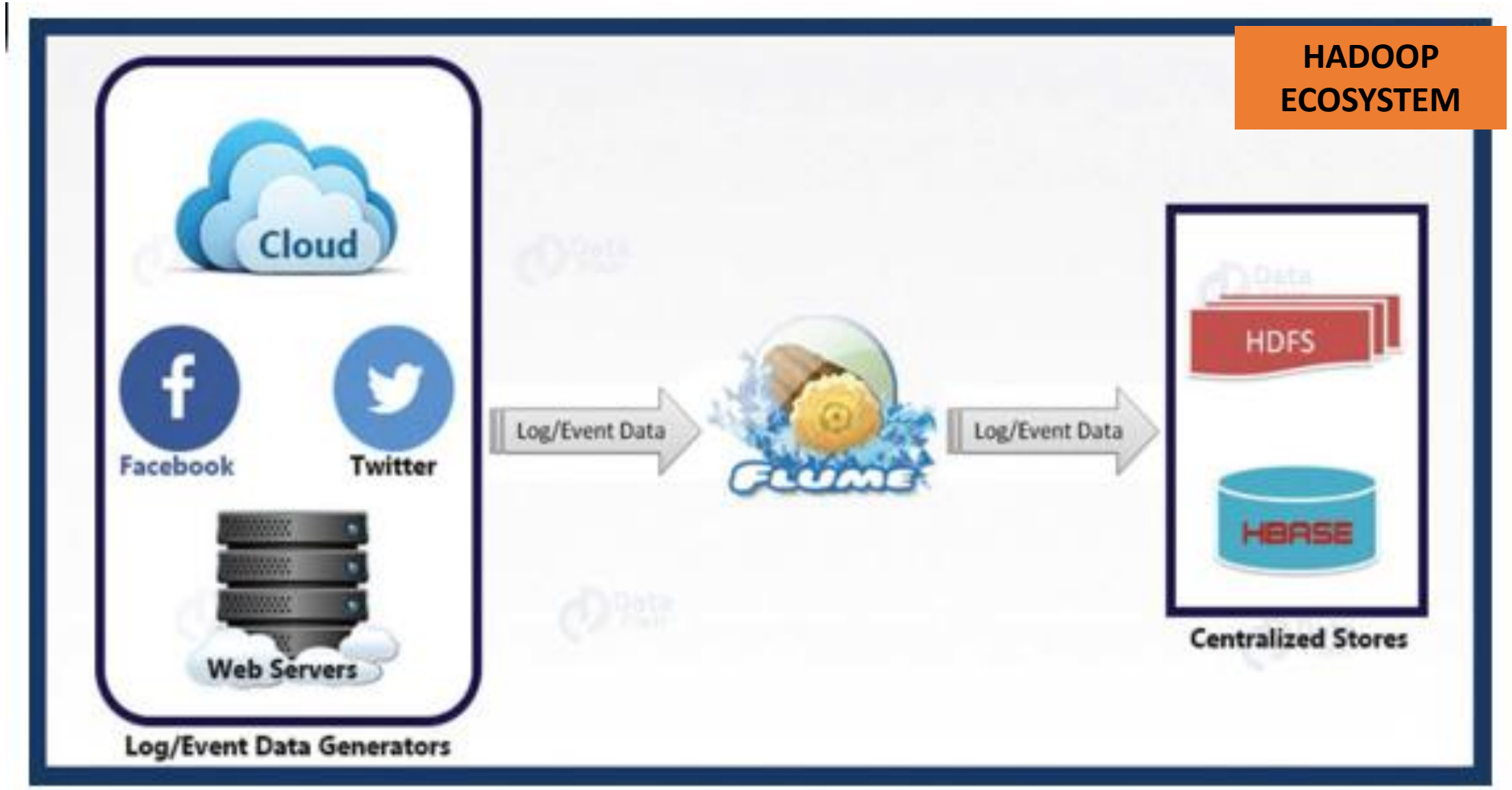


Thrift Diagram

HADOOP ECOSYSTEM

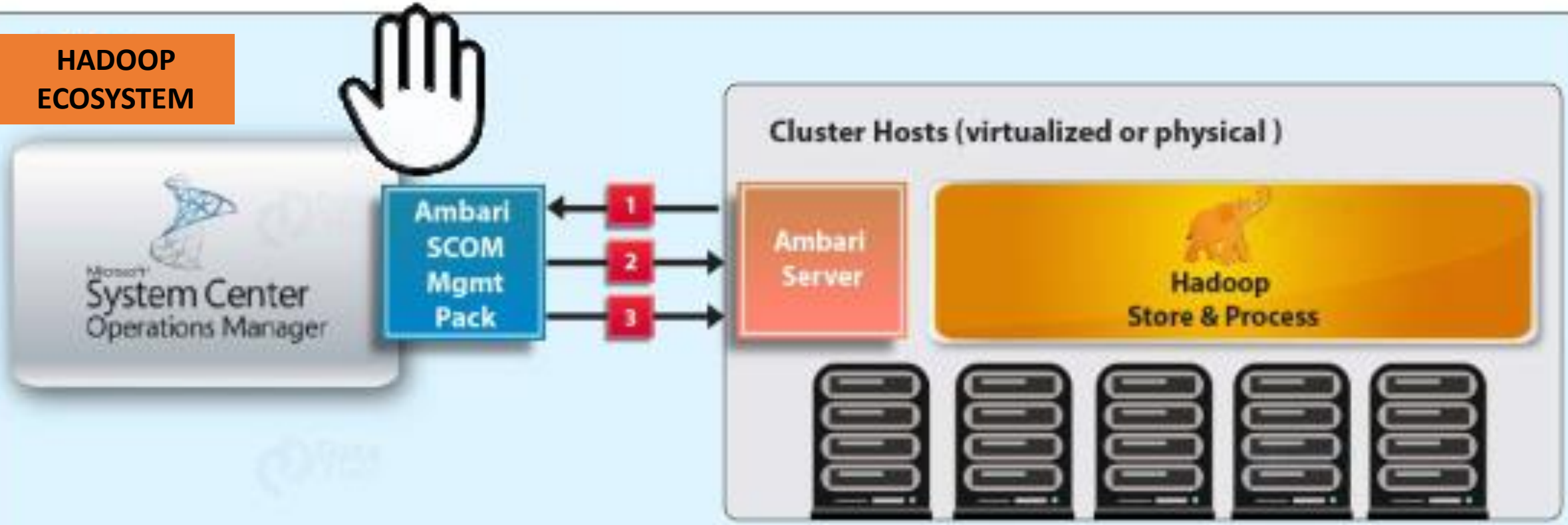


Apache Sqoop Diagram



Apache Flume

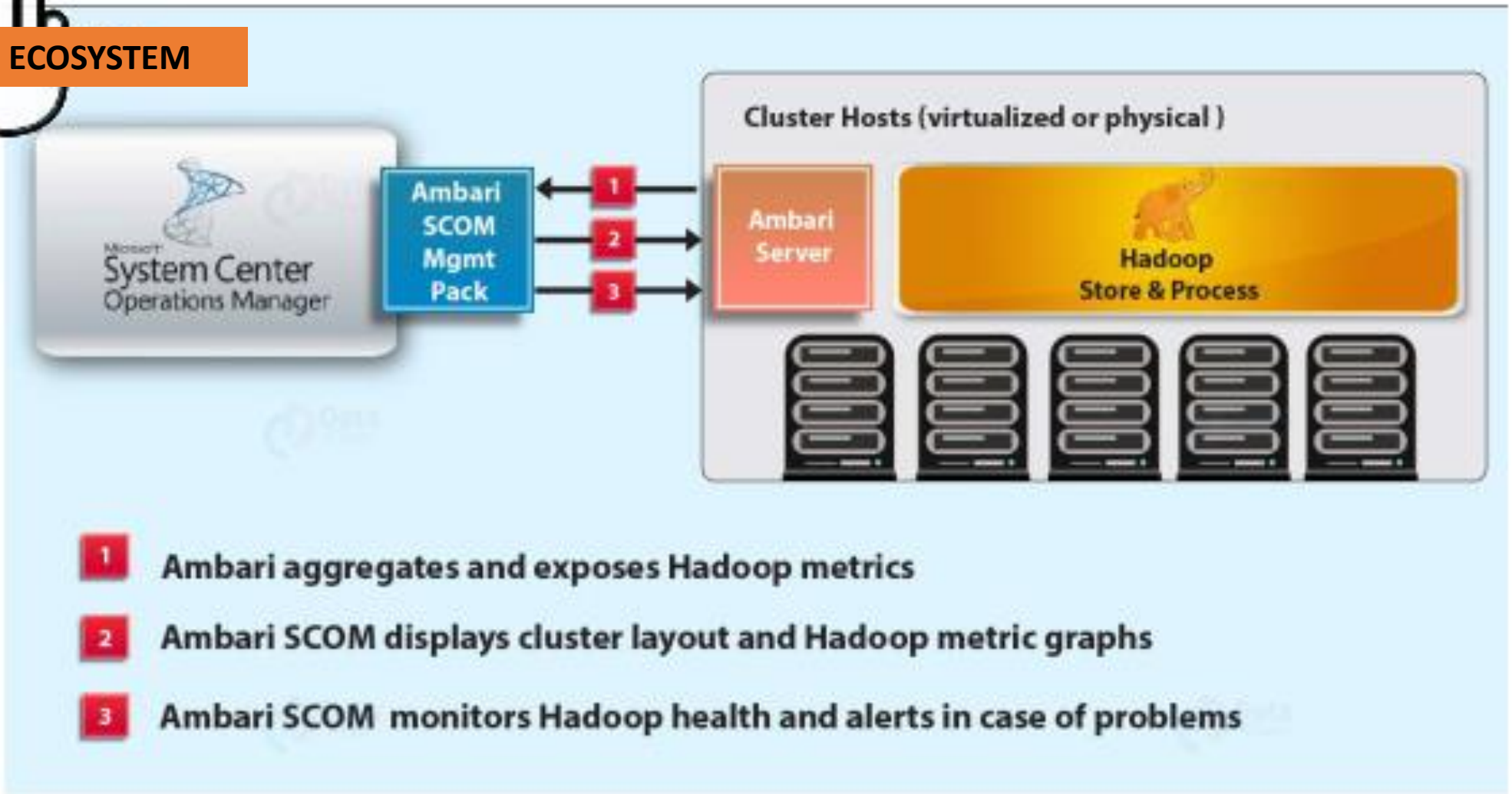
HADOOP ECOSYSTEM



1. Ambari aggregates and exposes Hadoop metrics
2. Ambari SCOM displays cluster layout and Hadoop metric graphs
3. Ambari SCOM monitors Hadoop health and alerts in case of problems

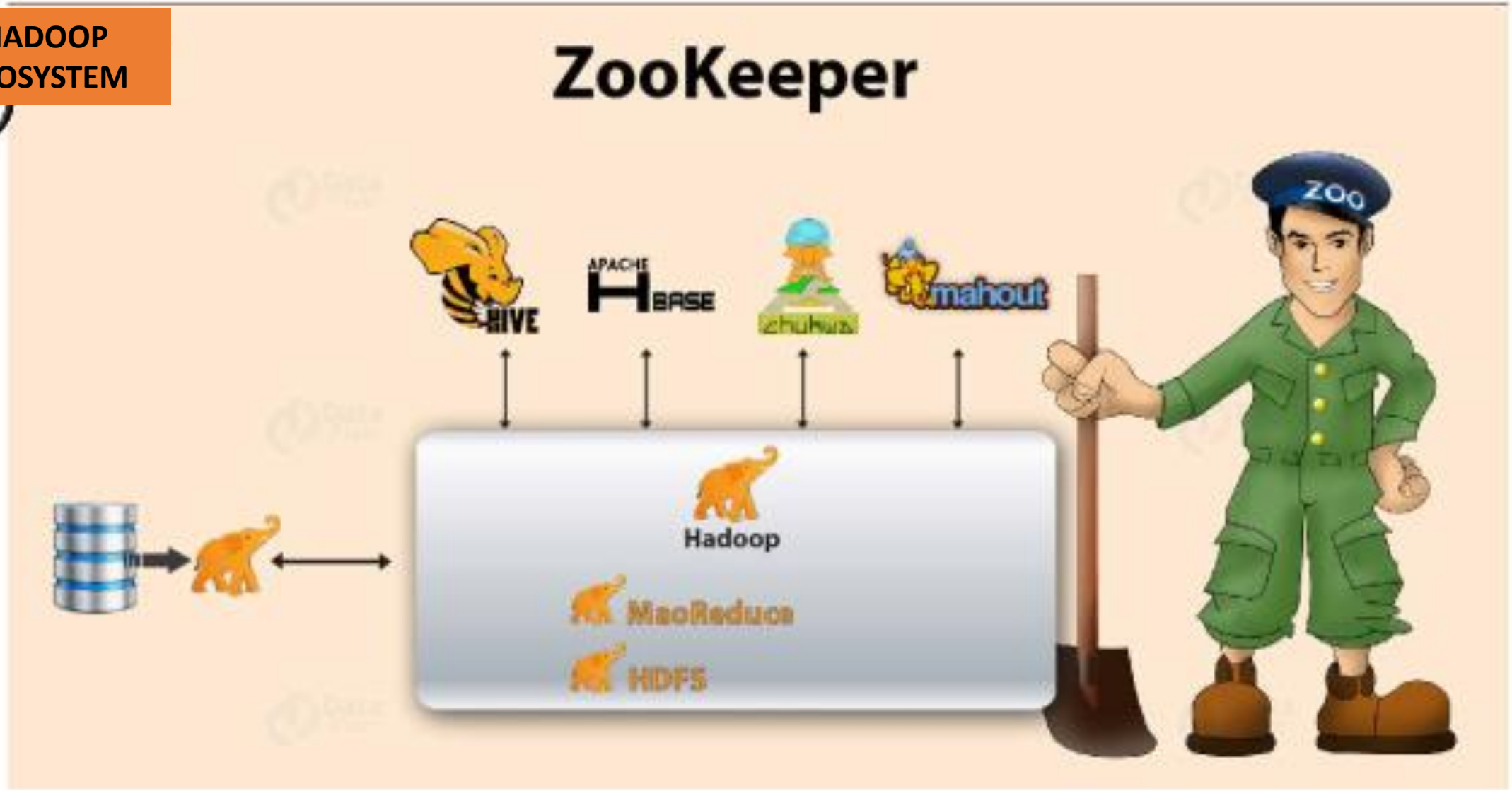
Ambari Diagram

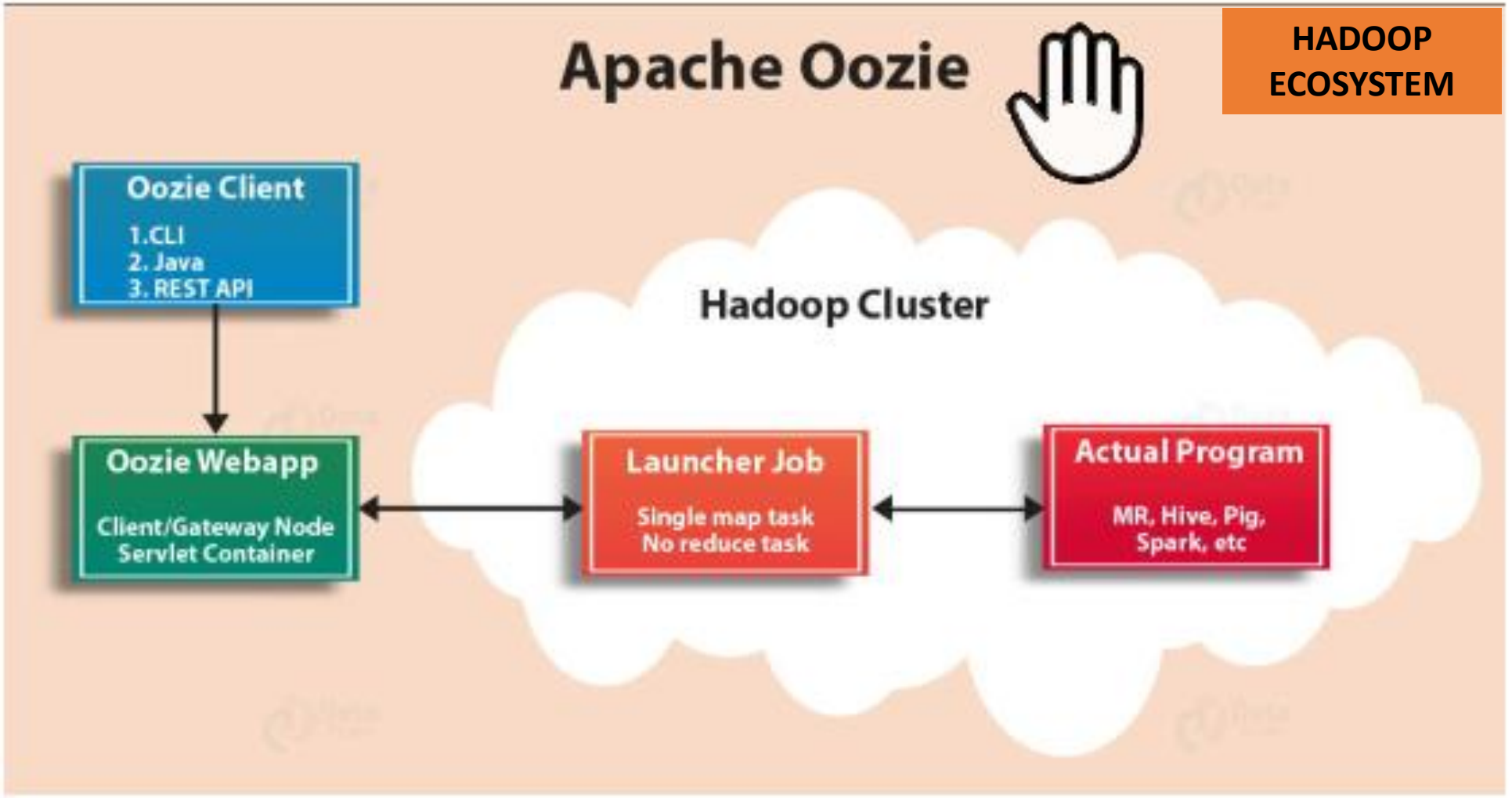
HADOOP ECOSYSTEM



Ambari Diagram

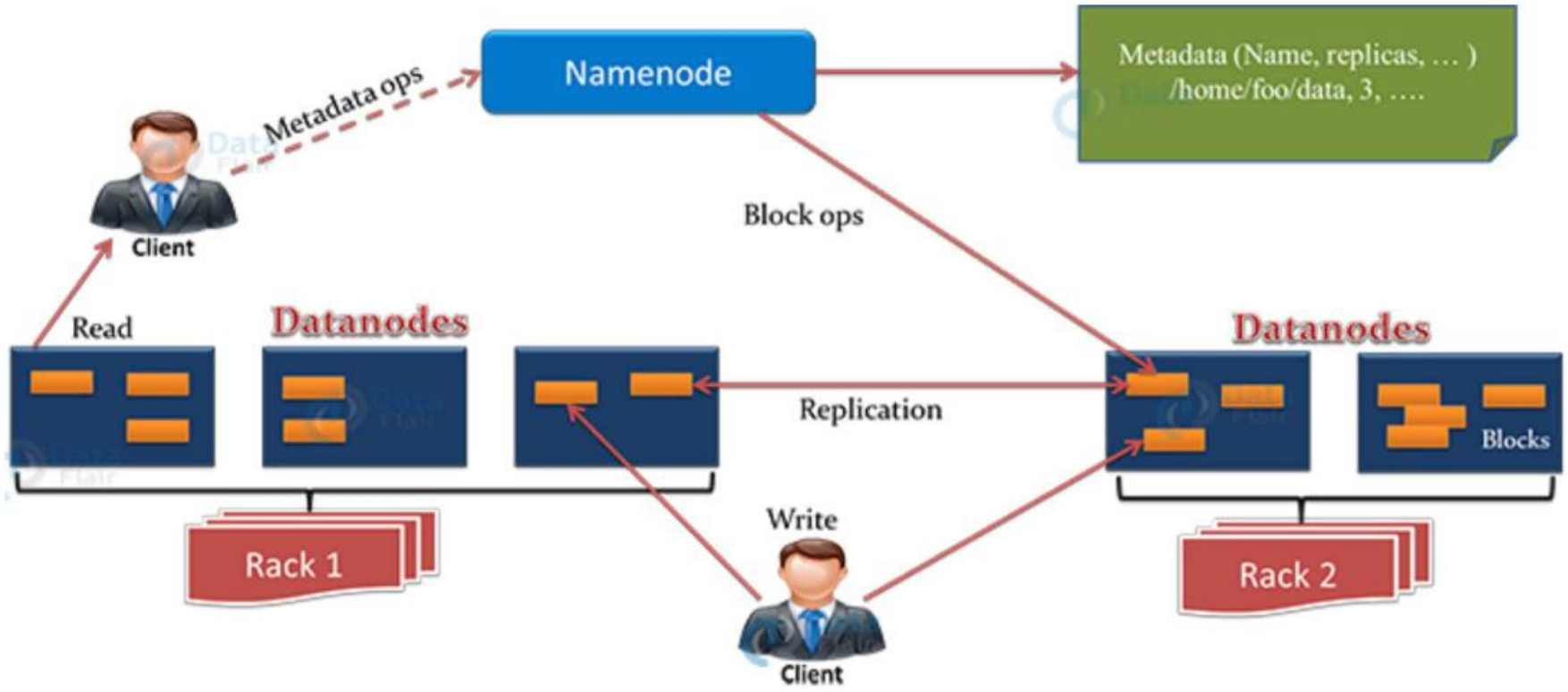
HADOOP ECOSYSTEM



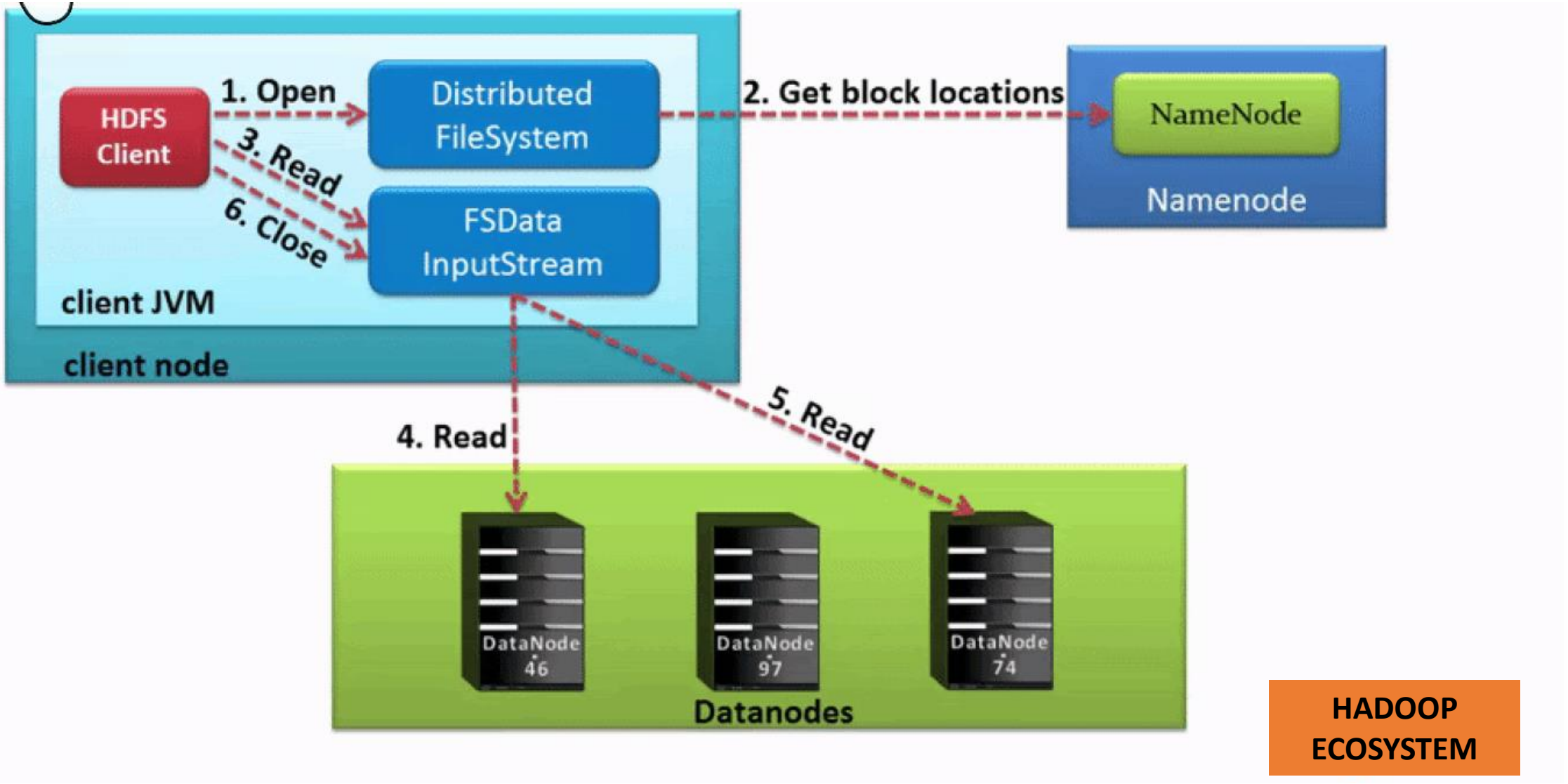


Oozie Diagram

HADOOP ECOSYSTEM

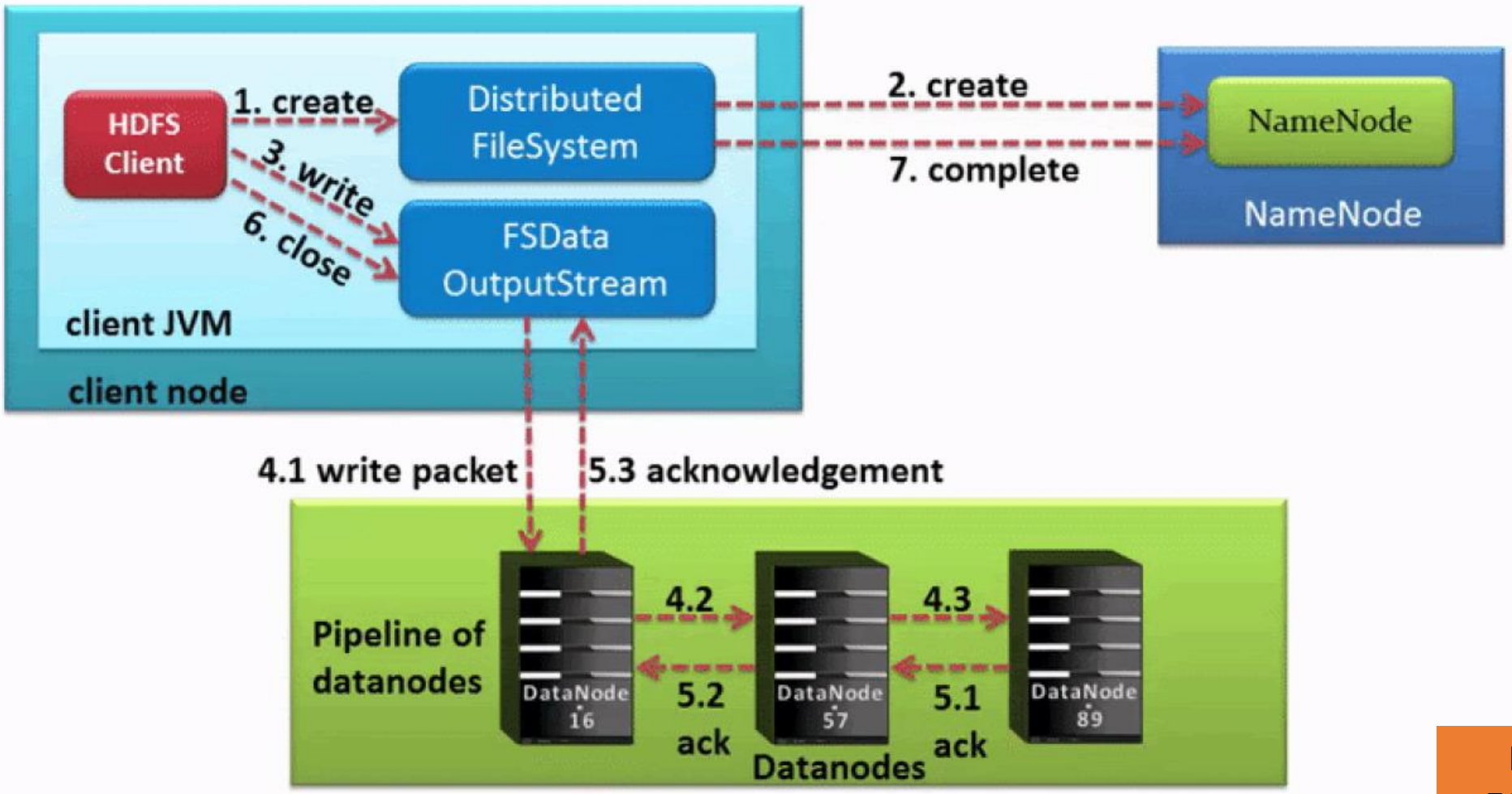


Read Operation in Hadoop HDFS

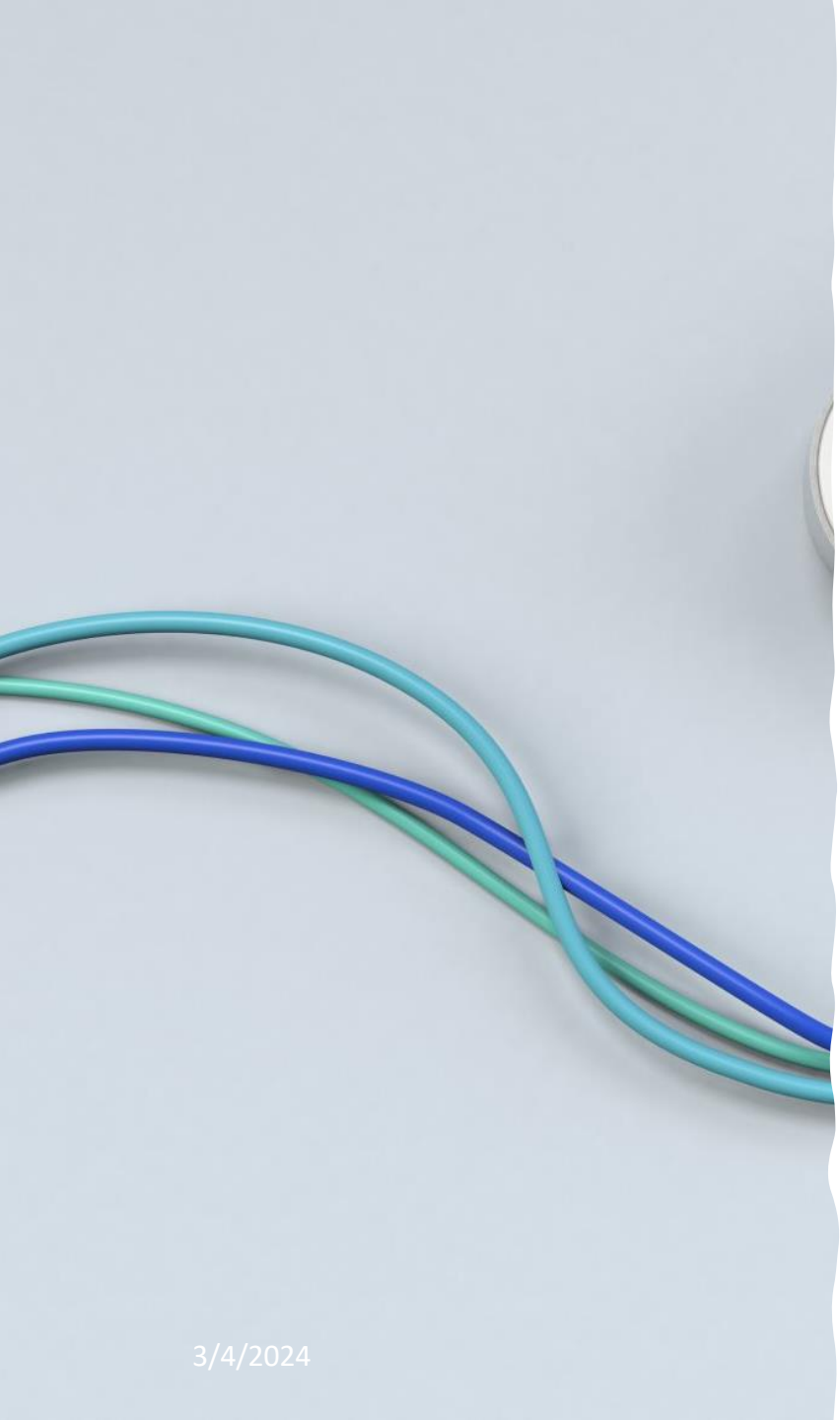


HADOOP ECOSYSTEM

Write Operation in Hadoop HDFS with 2 replications



HADOOP ECOSYSTEM



Summary

- Elastic Map Reduce (EMR) is the cloud-version of MR that is supported by all cloud providers in a platform as a service model (PaaS)
- Hadoop Ecosystem was developed in open-source as a cloud version for Infrastructure as a Services (IaaS) model.
- Handling a few very large data files that have more reads than writes (updates model) is what is used for large data analytics applications.