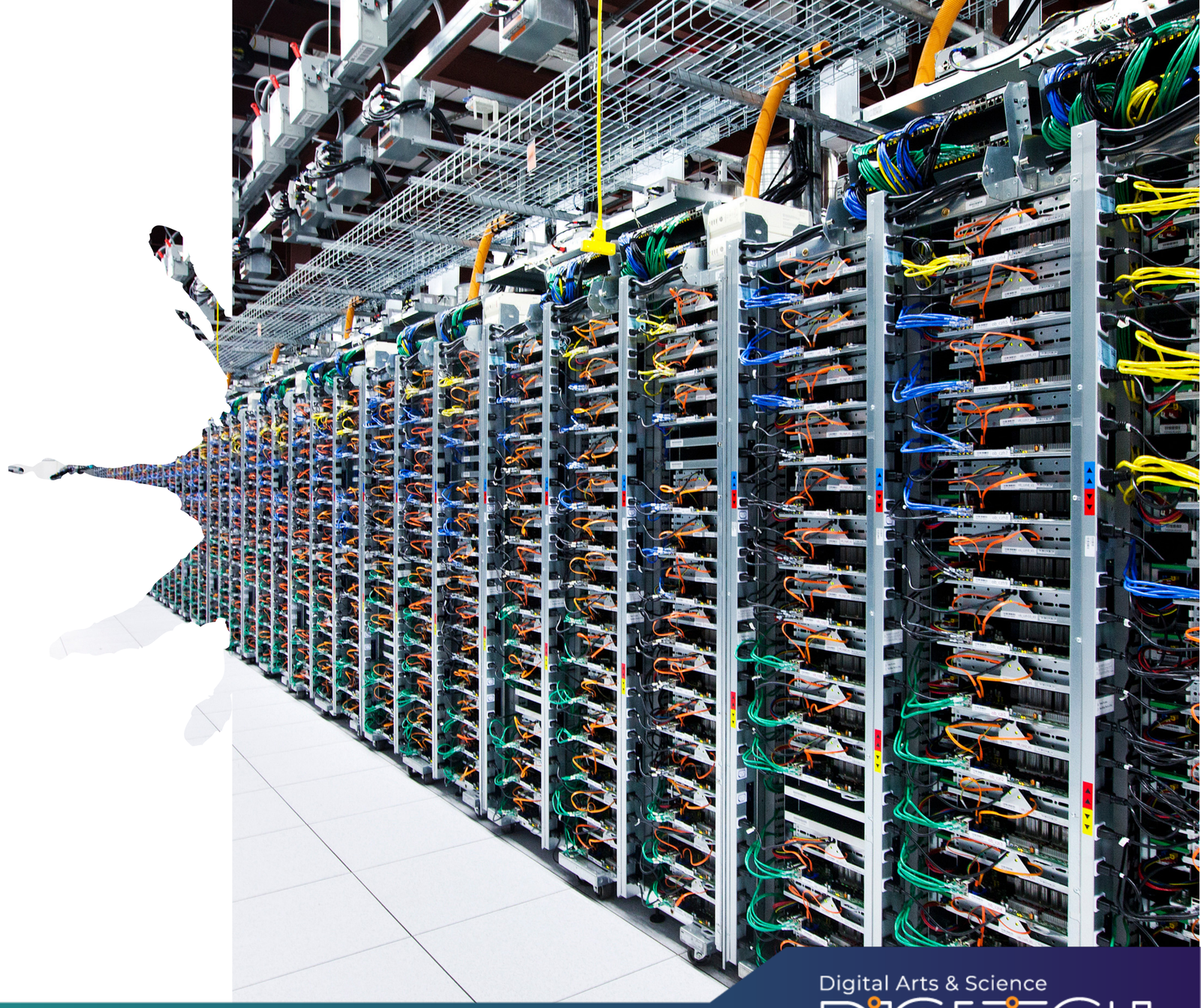
Big Data

Thara Angskun (angskun@sut.ac.th)

Digital Arts & Science
DIGITECH

Learning Objective

" explain big data and Hadoop foundation "



Question

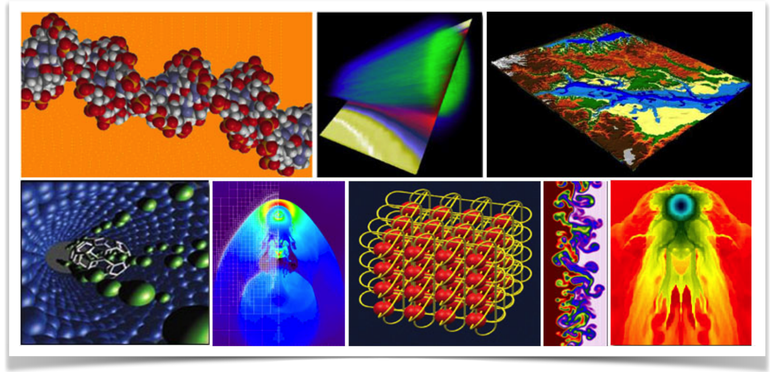
How many CPU hours did it take to render Shrek 3?

Question

- How many CPU hours did it take to render Shrek 3?
  - 20,000,000 CPU hours = 2,283 years

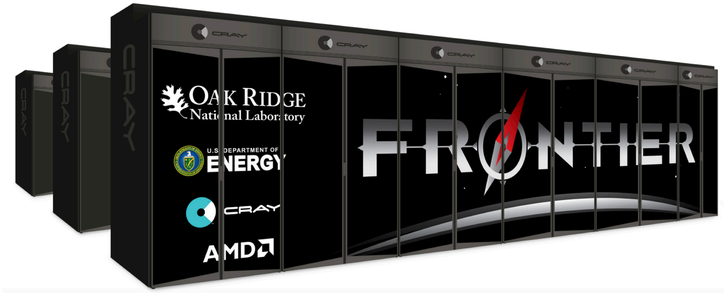
High Performance Computing

- Science & Engineering
  - Weather forecast
  - Deep learning
- Commercial & Business
  - Film Maker
  - Search Engine



The fastest computer in the world (Nov 2023!!)

- Frontier @ ORNL (TN, USA)
  - Speed: 1.102 -> 1.194 Exaflop/s
  - CPU: 8,730,112 -> 8,699,904 cores



The fastest computer in the world

iPhone XS

iPhone11,2-D321AP / 6 cores

Problem size: 500

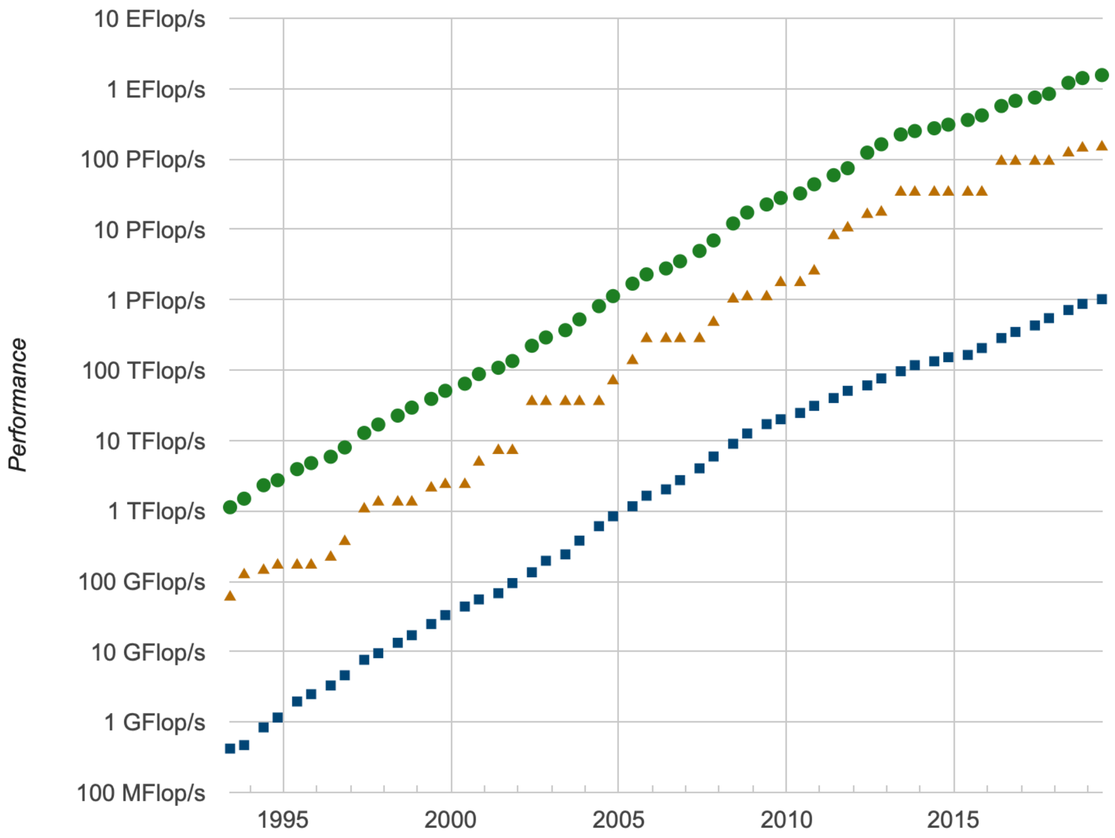
Number of runs: 10

Multithread mode:

Run benchmark

Run: #10
Mflop/s: 16832.87
Time: 0.0466
Norm Res: 5.1700
Precision: 2.22044605e-16

Max Mflop/s: 18995.09
Avg Mflop/s: 16977.87



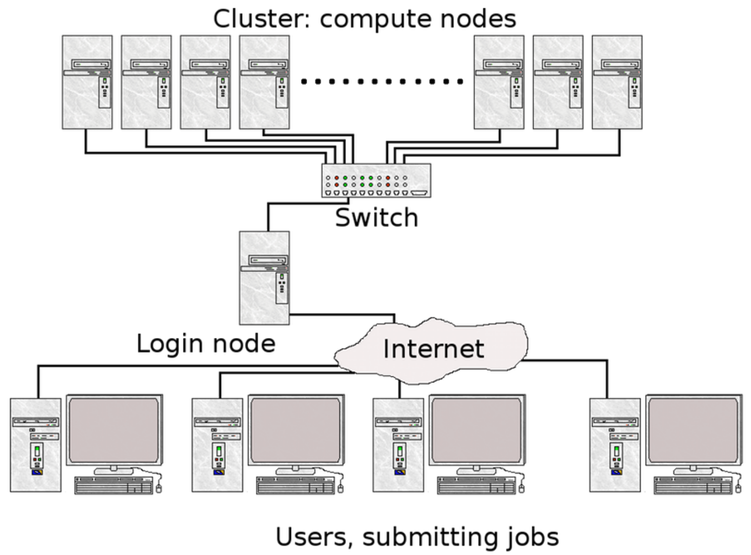
Digital Arts & Science DIGITECH

The fastest computer in Thailand (Nov 2023!!)

- Speed: 8.15 Petaflop/s
- CPU: 87,296 cores

Cluster Computing

- Type of Nodes
  - Front-end: Master, Login, Root
  - Back-end: Worker, Slave, Peon, Compute

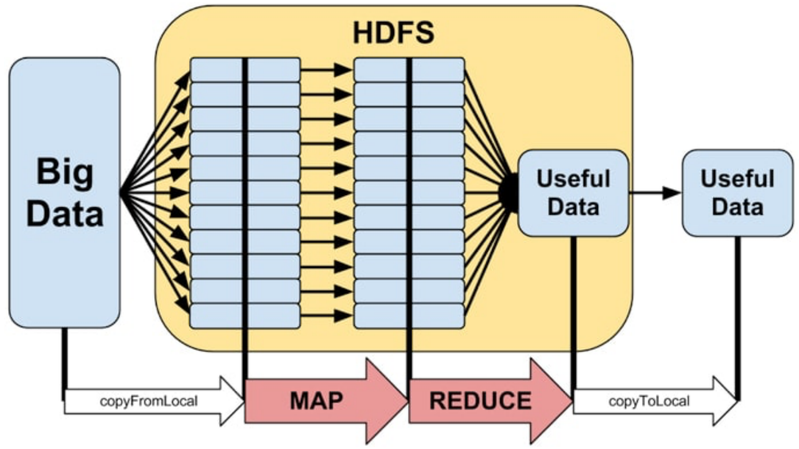


Digital Arts & Science DIGITECH

Cluster Computing

Hadoop

- Cluster Computing Abstraction
- Operating System for Big Data
  - Distributed data storage
  - Parallel computation



- Prerequisite
  - Hadoop cluster is run on open source software
  - Linux and command line are required.
    - Even with Microsoft Azure!

When (not) to use Hadoop

When to use Hadoop

- For processing really big data
- For storing a diverse set of data
- For parallel data processing

When NOT to use Hadoop

- For real-time data analysis
- For a relational database system
- For a general network file system
- For non-parallel data processing

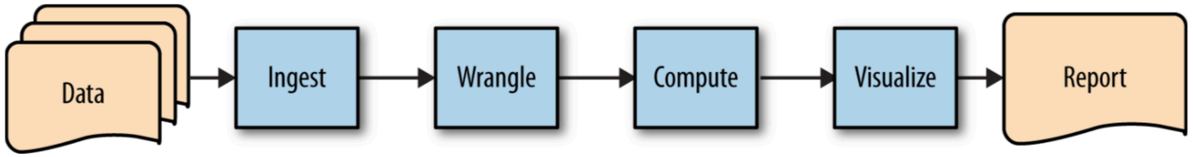
The Age of "Data Product"

- Derives value from data
  - then produce more data, more value, ...
- Data + Statistical Algorithms
  - for inferences or prediction
- Example
  - Amazon recommendation
  - Facebook's People You May Know
  - Nest thermostat

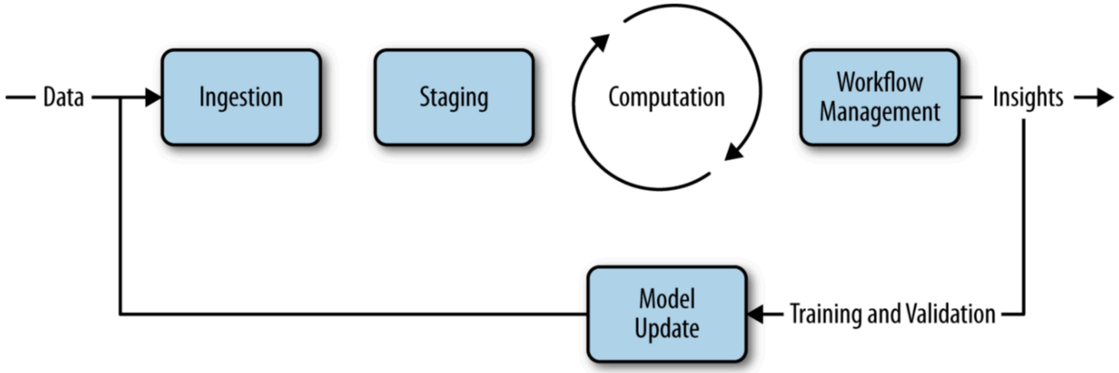
Digital Arts & Science DIGITECH

Data Science vs. Big Data Pipeline

Data Science



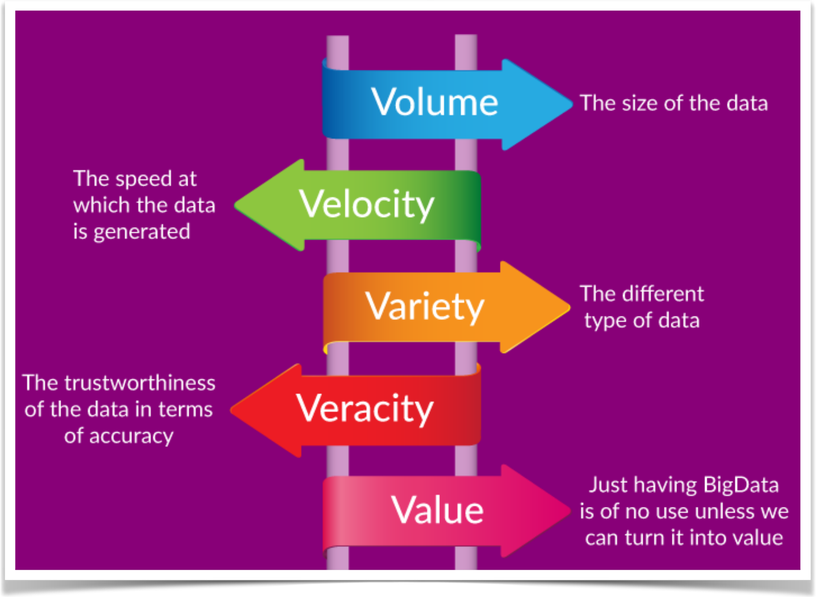
Big Data (= Data Science + Scalability + Automation)



Hadoop as an OS for Big Data

Distributed System Requirements

- Fault Tolerance
- Recoverability
- Consistency
- Scalability

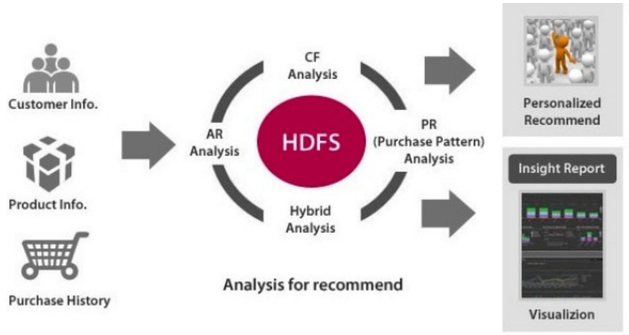


Digital Arts & Science DIGITECH

Hadoop Case Studies

Understand customer behavior

- Mark & Spencer, Royal Mail,
- Expedia (&Trivago, hotels), Yahoo,
- Western Union, Royal Bank of Scotland



Understand gamer behavior

- King.com (e.g., Candy Crush)
  - Analyze every "event/action"

Digital Arts & Science DIGITECH

Hadoop Case Studies

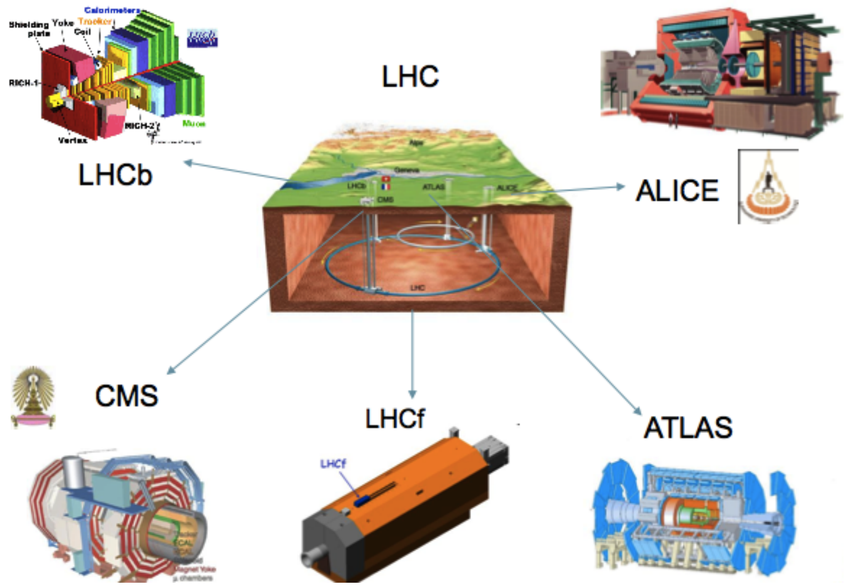
- Tesla: to analyze connected cars
  - Driverless cars

- BT (UK's telecommunication company):
  - identify network problems

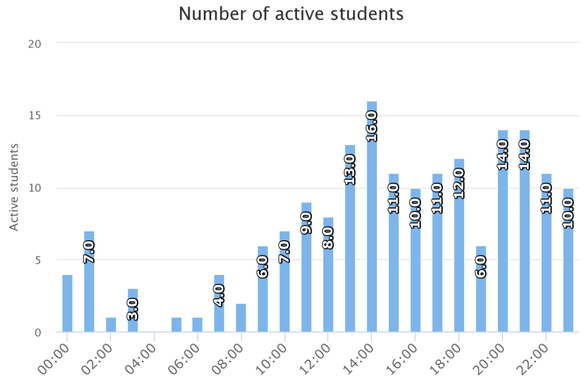
Hadoop Case Studies

CERN: Petabytes per second!!

- ALICE detector



Hadoop Case Studies



B6070014	152	13		8	
B6070229	52	11		5	
B6070243	50	11		7	
B6070250	41	10		7	
B6070267	105	17		7	
B6070281	36	9		7	

Digital Arts & Science DIGITECH

Hadoop History

Doug Cutting adds DFS & MapReduce support to Nutch

Doug Cutting & Mike Cafarella started working on Nutch

NY Times converts 4TB of Image archives over 100 EC2s

YAHOO!
Fastest sort of a TB, 3.5mins over 910 nodes

Fastest sort of a TB, 62secs over 1,460 nodes
Sorted a PB in 16.25hours
Over 3.658 nodes

2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009

Yahoo! hires Cutting, Hadoop spins out of Nutch

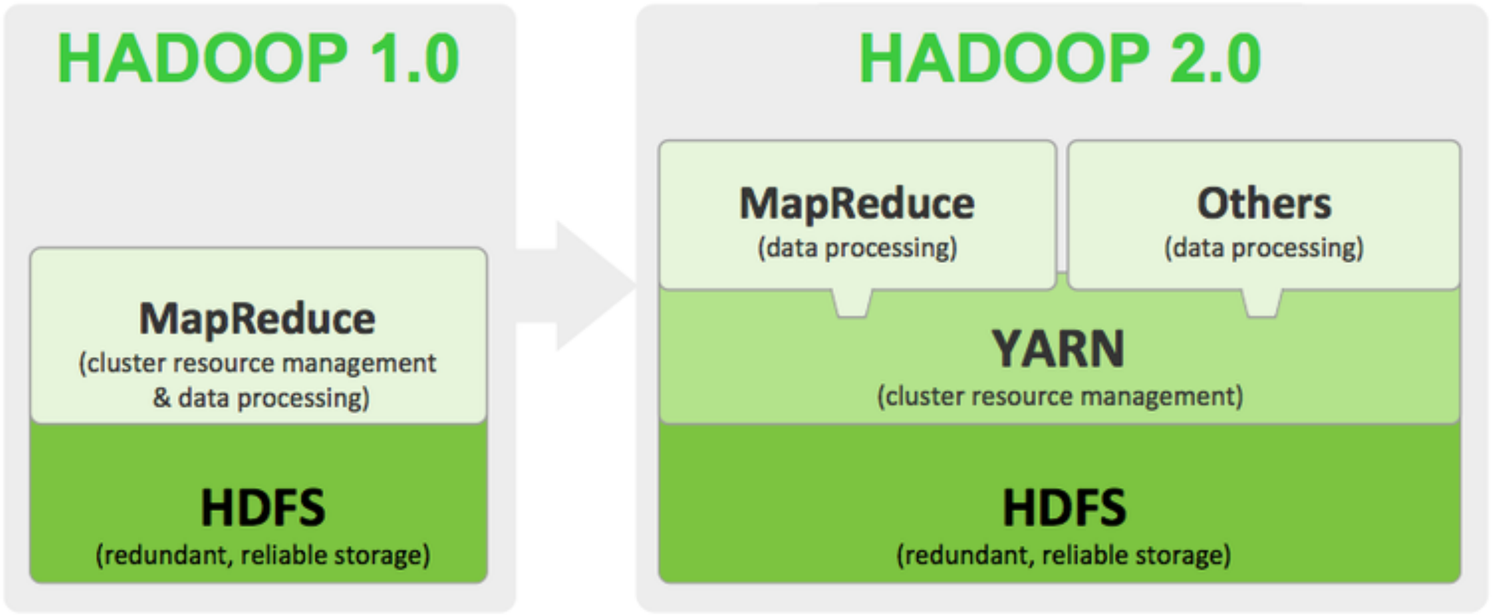
Facebook launches Hive: SQL Support for Hadoop

cloudera
Founded

Doug Cutting Joins Cloudera

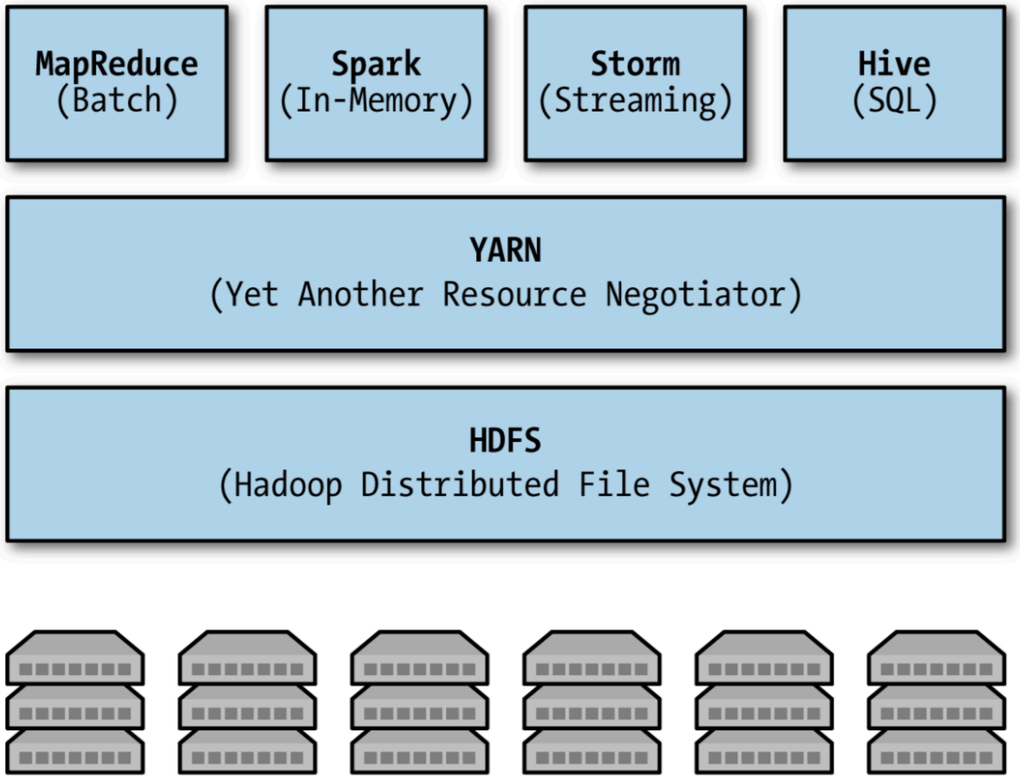
Hadoop Summit 2009, 750 attendees

Hadoop History

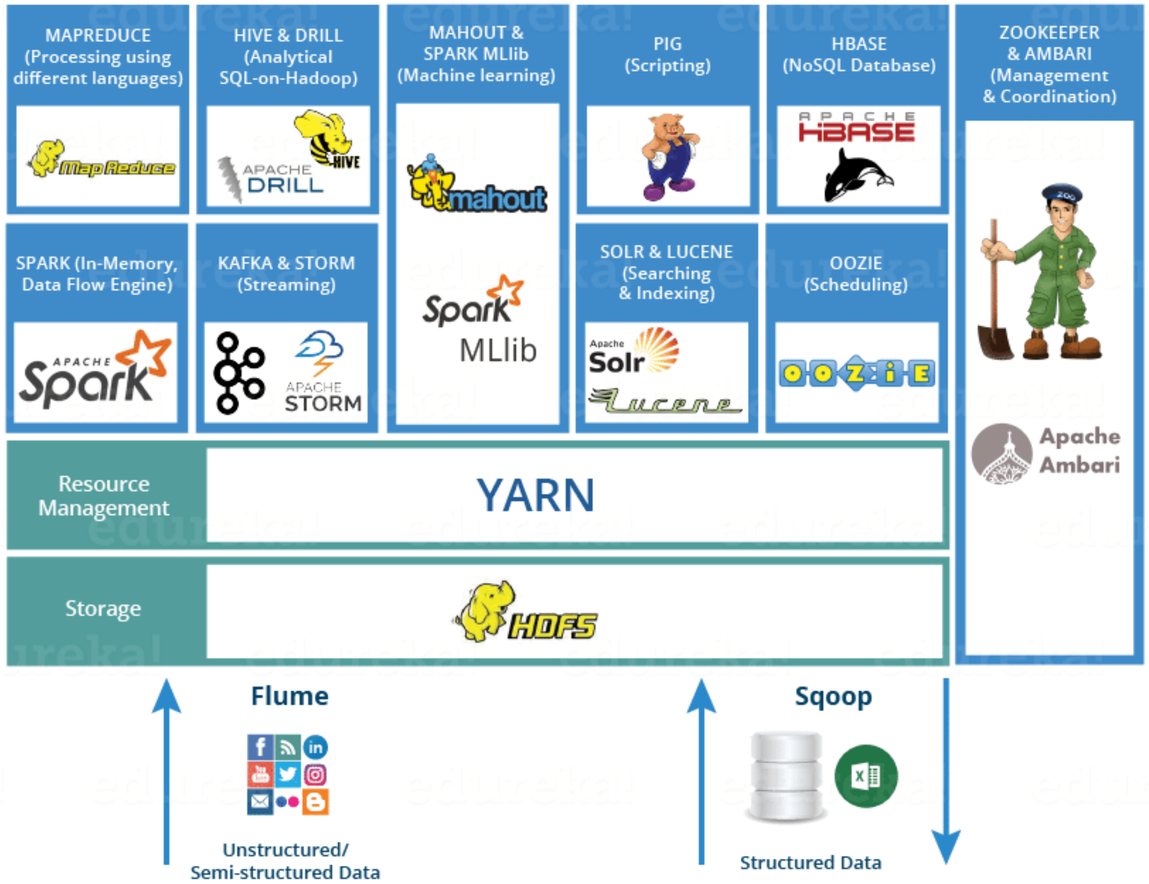


Digital Arts & Science DIGITECH

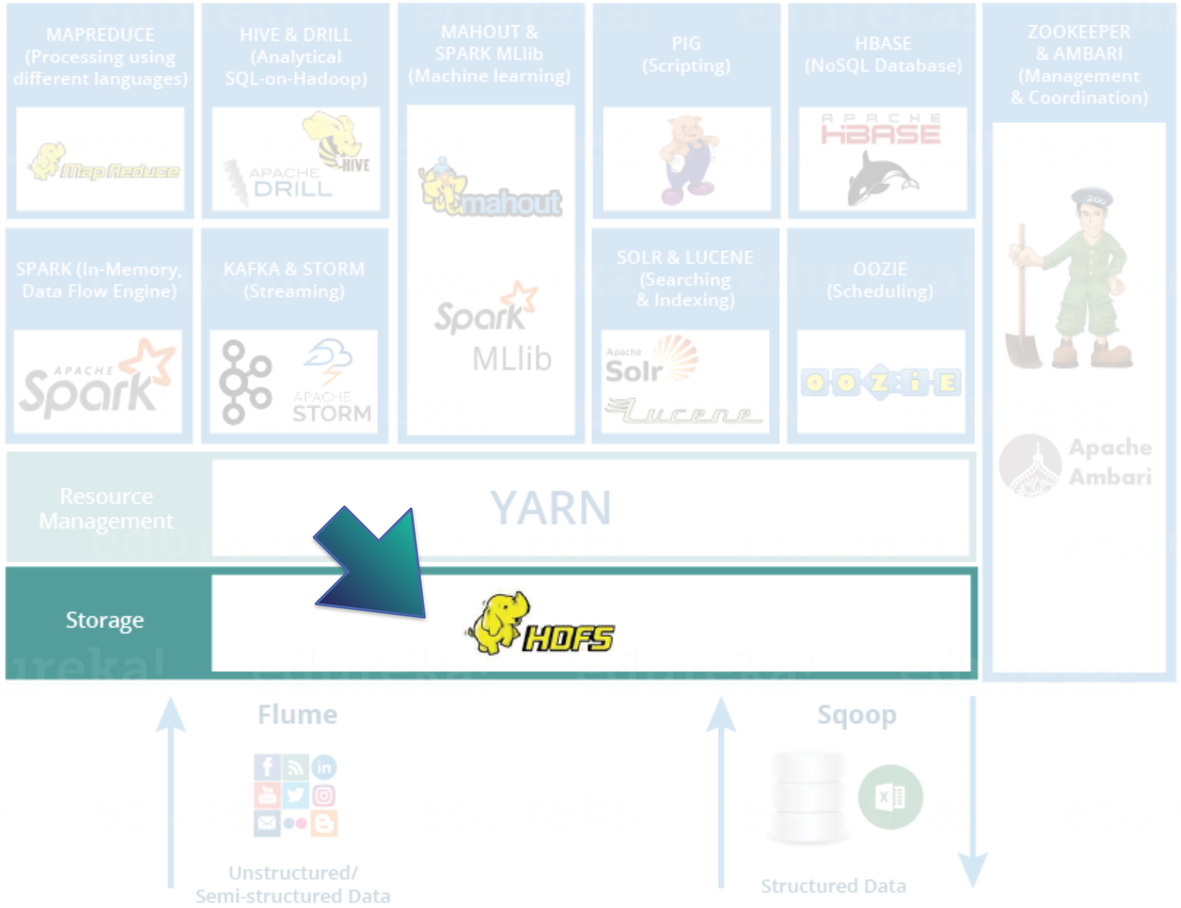
Hadoop Architecture



Hadoop Architecture



HDFS


MAPREDUCE (Processing using different languages)
HIVE & DRILL (Analytical SQL-on-Hadoop)
MAHOUT & SPARK MLlib (Machine learning)
PIG (Scripting)
HBASE (NoSQL Database)
ZOOKEEPER & AMBARI (Management & Coordination)
SPARK (In-Memory, Data Flow Engine)
KAFKA & STORM (Streaming)
SOLR & LUCENE (Searching & Indexing)
OOZIE (Scheduling)
Resource Management
YARN
Storage
HDFS
Flume
Sqoop
Unstructured/ Semi-structured Data
Structured Data

HDFS – Hadoop Distributed File System

NameNode (@master)

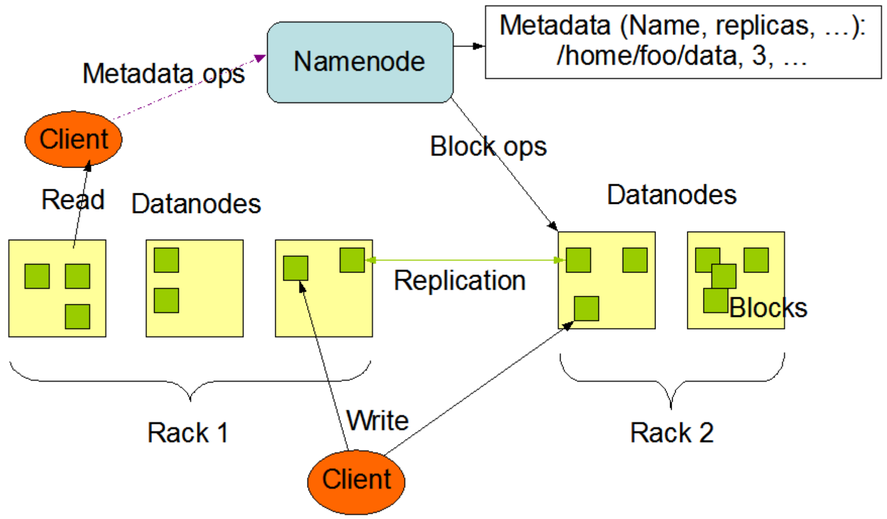
- Metadata
- Single point of failure!
  - Require HA

Secondary NameNode (Optional)

- Not a backup NameNode
- Housekeeping (log)
- Checkpointing (state)

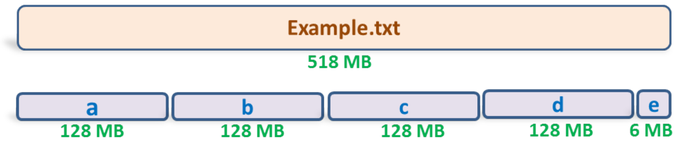
DataNode (@worker)

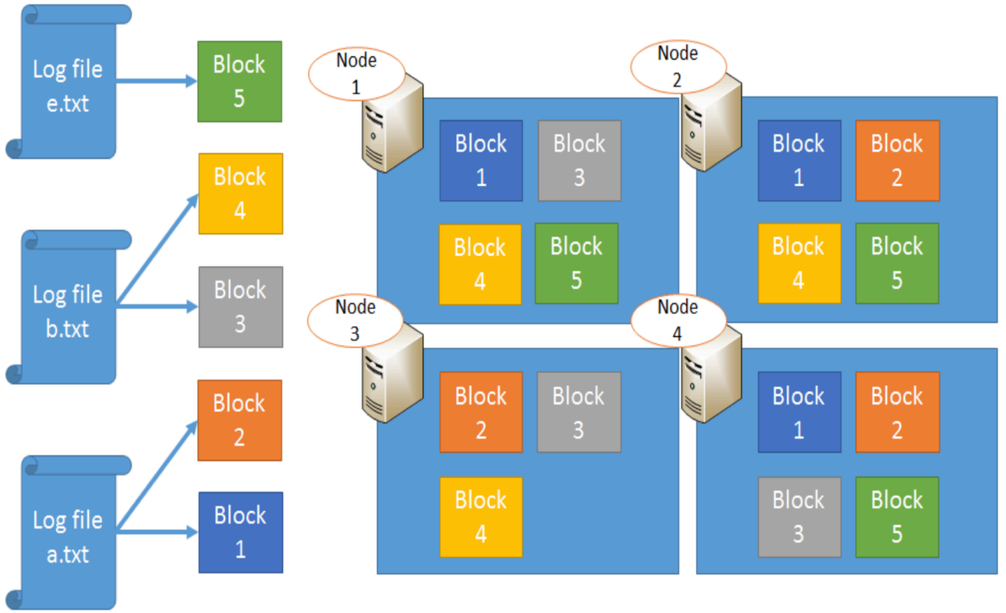
- Store & manage HDFS block



HDFS - Hadoop Distributed File System

- Software on top of a native file system e.g., ext4, xfs
  - Files are split into blocks (64MB, 128MB, or 256MB)
  - By default, each block exists on 3 different nodes





HDFS - Hadoop Distributed File System

- Performs best with large files.
- WORM: Write Once, Read Many (no random, append)
  - "Data Lake!!", Not good for real-time/interactive

Basic File System Operations

Help

- $ hadoop fs -help

Copy from local to hdfs

- $ hadoop fs -copyFromLocal <src> <dst>
- $ hadoop fs -put <src> <dst>

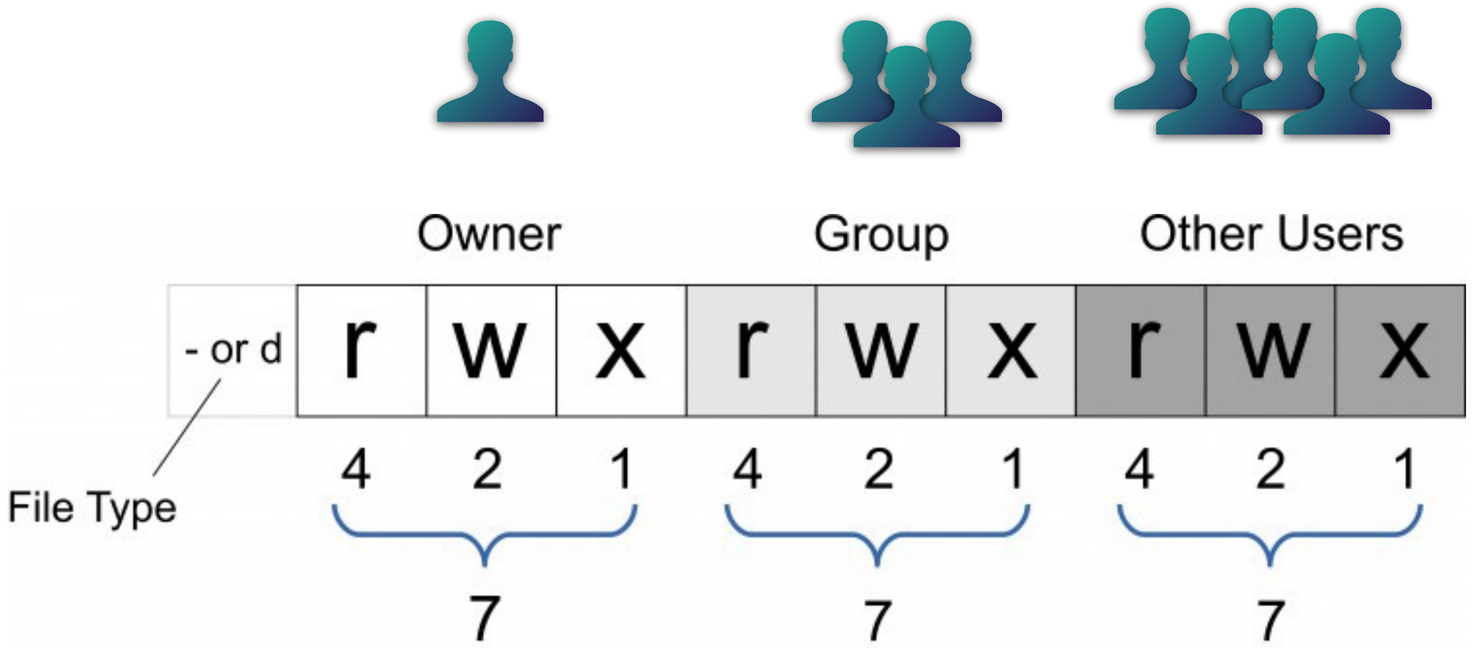
Copy from hdfs to local

- $ hadoop fs -copyToLocal <src> <dst>
- $ hadoop fs -get <src> <dst>

Basic File System Operations

Operation	LINUX Command	Hadoop Command
Making a Directory	mkdir	hadoop fs -mkdir
Listing files	ls	hadoop fs -ls
Moving Files	mv	hadoop fs -mv
Copying Files	cp	hadoop fs -cp
Removing Files	rm	hadoop fs -rm
Display content of Files	cat	hadoop fs -cat

File Permission

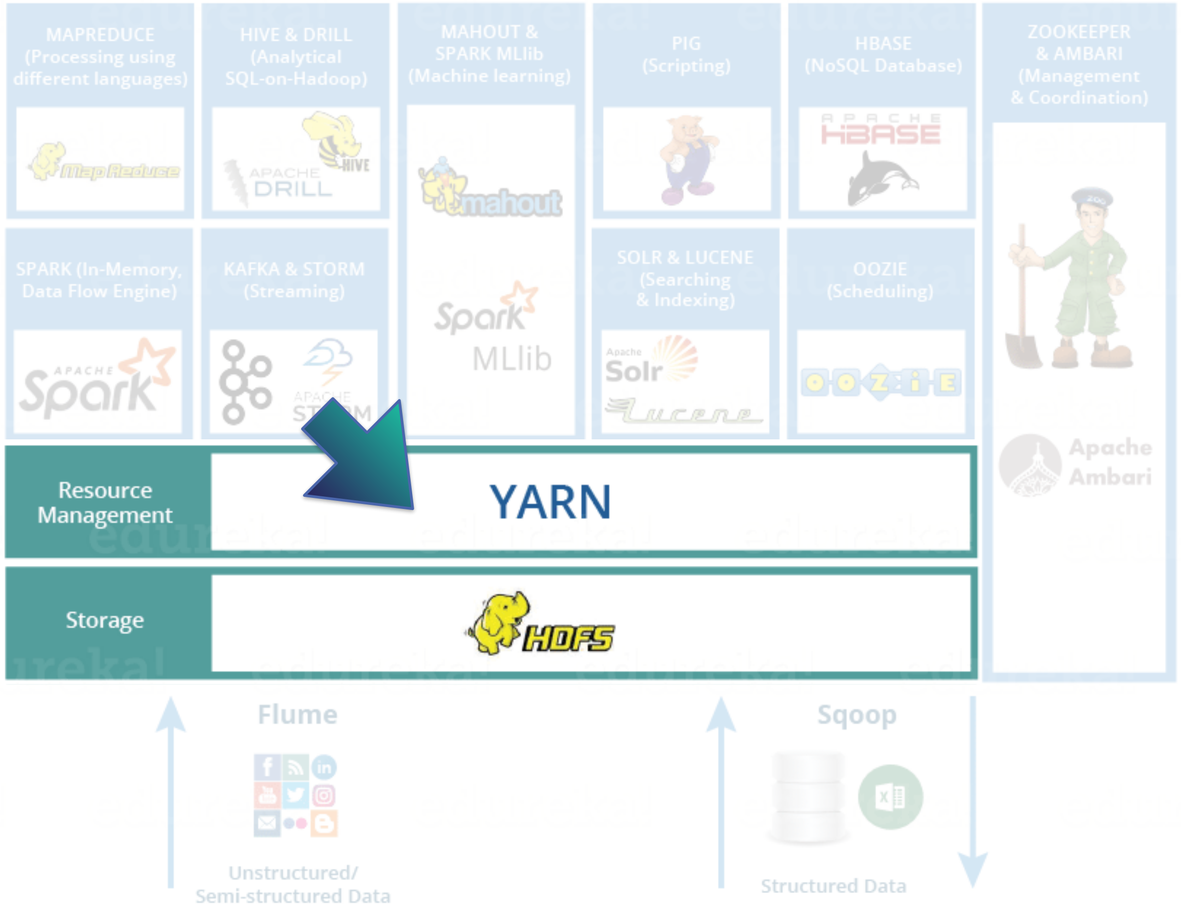


$ hadoop fs -chmod 664 file.txt

Accessing HDFS

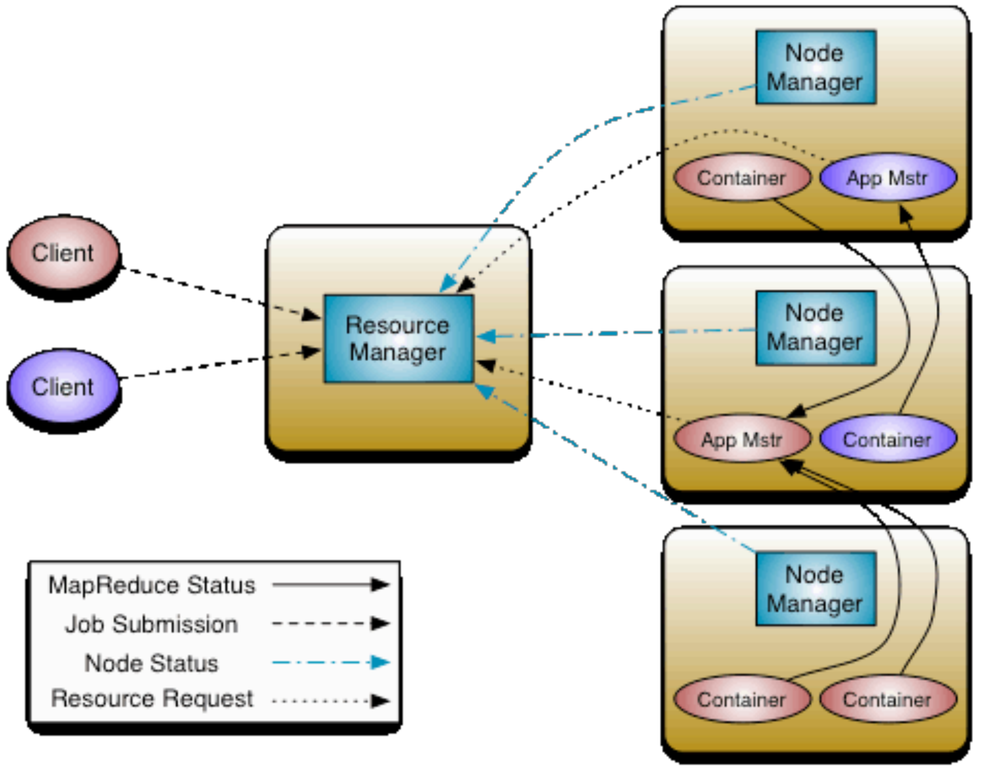
- Java API, Python, FTP, Amazon S3
- HTTP (read only)
  - Namenode: http://127.0.0.1:50070
  - Secondary Namenode: http://127.0.0.1:50090
  - Datanode: http://127.0.0.1:50075

YARN


MAPREDUCE (Processing using different languages)
HIVE & DRILL (Analytical SQL-on-Hadoop)
MAHOUT & SPARK MLlib (Machine learning)
PIG (Scripting)
HBASE (NoSQL Database)
ZOOKEEPER & AMBARI (Management & Coordination)
SPARK (In-Memory, Data Flow Engine)
KAFKA & STORM (Streaming)
SOLR & LUCENE (Searching & Indexing)
OOZIE (Scheduling)
Resource Management
YARN
Storage
HDFS
Flume
Sqoop
Unstructured/ Semi-structured Data
Structured Data

YARN – Yet Another Resource Negotiator

- ResourceManager (@master)
  - monitor resources
  - schedule jobs
- ApplicationMaster
  - coordinate applications
- NodeManager (@worker)
  - run and manage tasks



YARN Operations

Resource Management (http://127.0.0.1:8088)

- Display Node status: $ yarn node -list
- Display Queue: $yarn queue

Job/Application Management

- Run a Jar file : $ yarn jar <jar>
- List job: $ yarn application -list
- Display job status: $ yarn application -status <app-id>
- Kill job: $ yarn application -kill <job-id>

YARN Operations

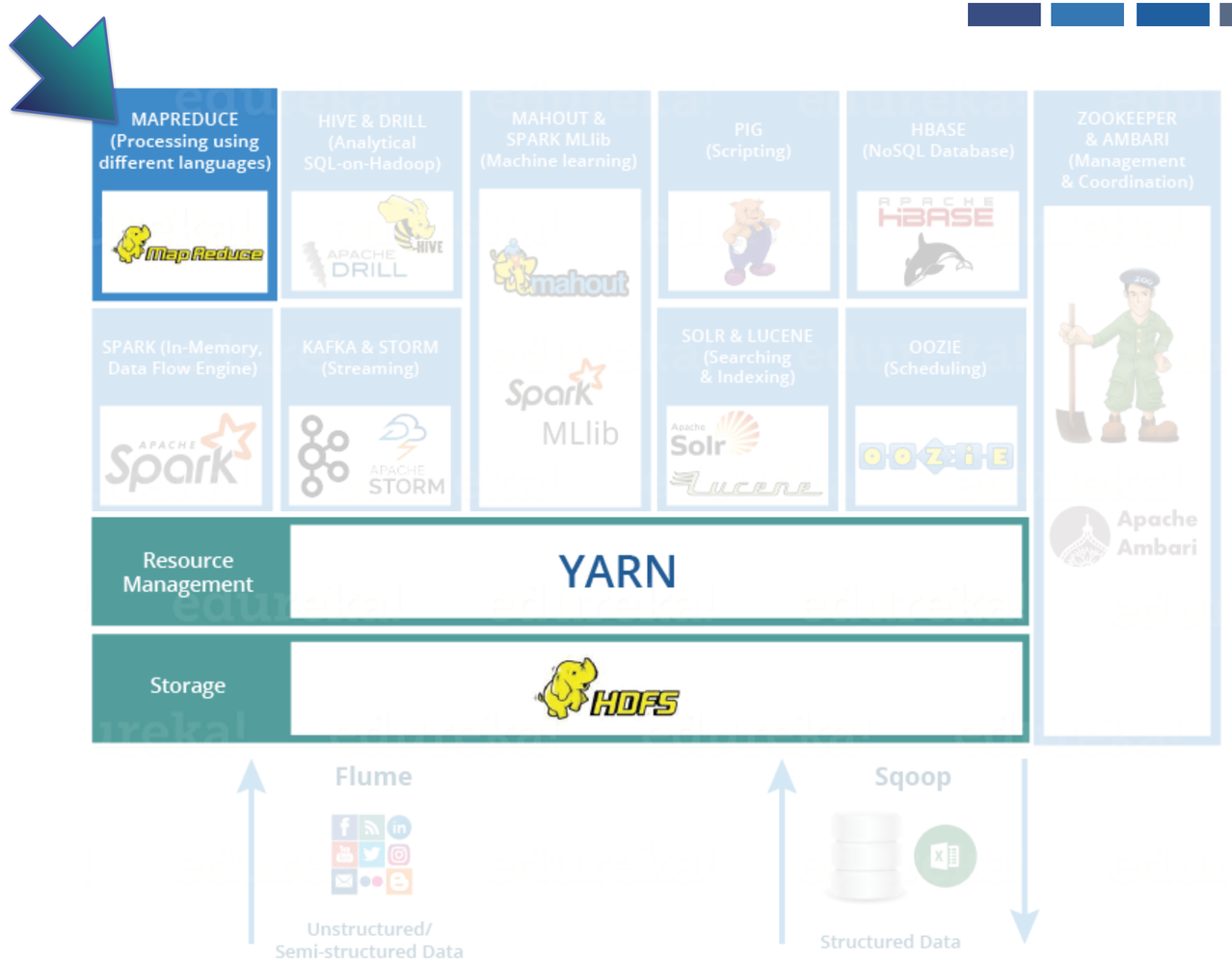
Environment

- Display Classpath: $ yarn classpath
- Display Environment Variables: $yarn envvars

Debug / Log

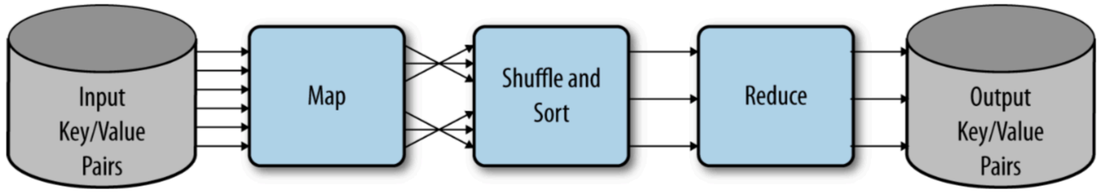
- Print container report: $yarn container
- Dump container logs: $yarn logs
- Log file: /tmp/hadoop-yarn, /var/log/hadoop-yarn
- Display version: $yarn version

MapReduce


MAPREDUCE (Processing using different languages)
HIVE & DRILL (Analytical SQL-on-Hadoop)
MAHOUT & SPARK MLlib (Machine learning)
PIG (Scripting)
HBASE (NoSQL Database)
ZOOKEEPER & AMBARI (Management & Coordination)
SPARK (In-Memory, Data Flow Engine)
KAFKA & STORM (Streaming)
SOLR & LUCENE (Searching & Indexing)
OOZIE (Scheduling)
Resource Management
YARN
Storage
HDFS
Flume
Sqoop
Unstructured/ Semi-structured Data
Structured Data

MapReduce

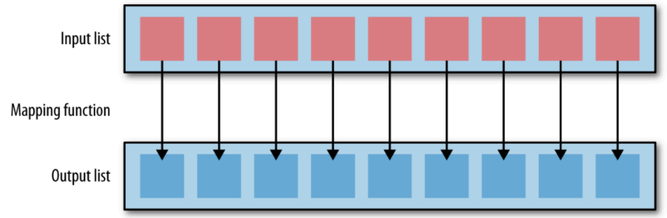
- Fault-tolerant Computation
- Functional Programming (no share state)
- Move computation to data!!
  - Jeffrey Dean and Sanjay Ghemawat (Google)
  - "MapReduce: Simplified Data Processing on Large Clusters".



MapReduce

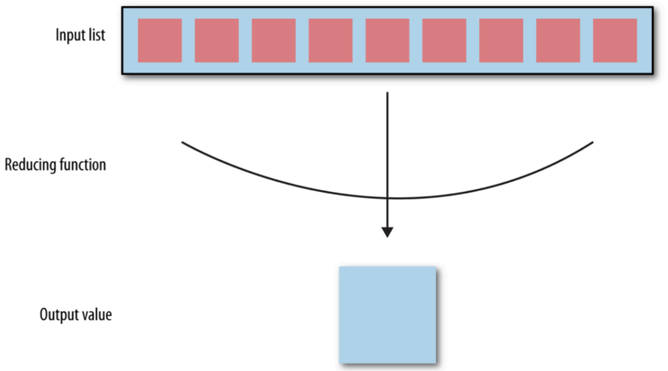
def map(key, value):

- return (intermed_key, intermed_value)

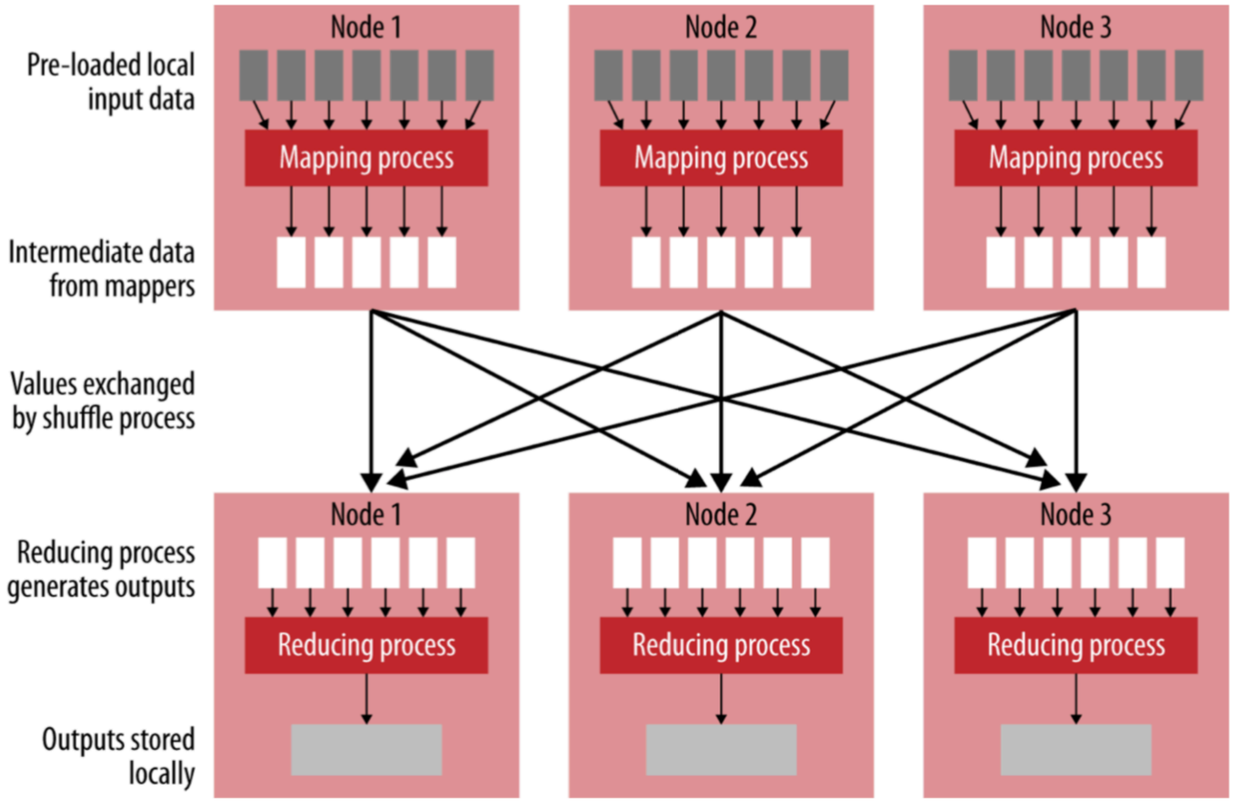


def reduce(intermed_key, values):

- return (key, output)

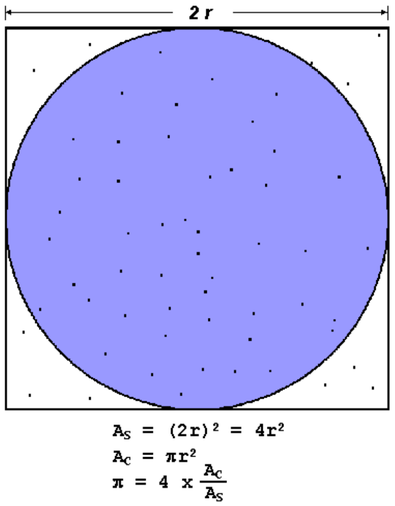


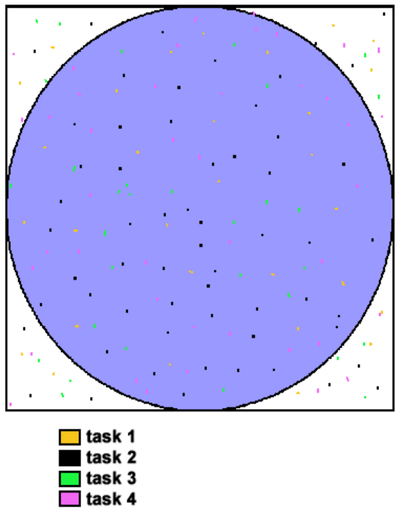
MapReduce on a Cluster

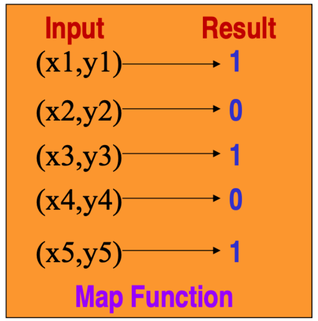


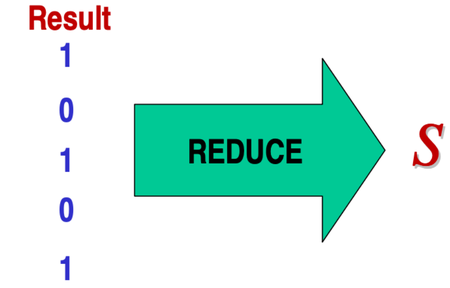
MapReduce: Pi Estimation

- Create random (x, y) pair values
  - Evaluate if it is in the circle or not (x2 + y2 < r) [MAP]
  - Collect results, evaluate the ratio, calculate PI [REDUCE]



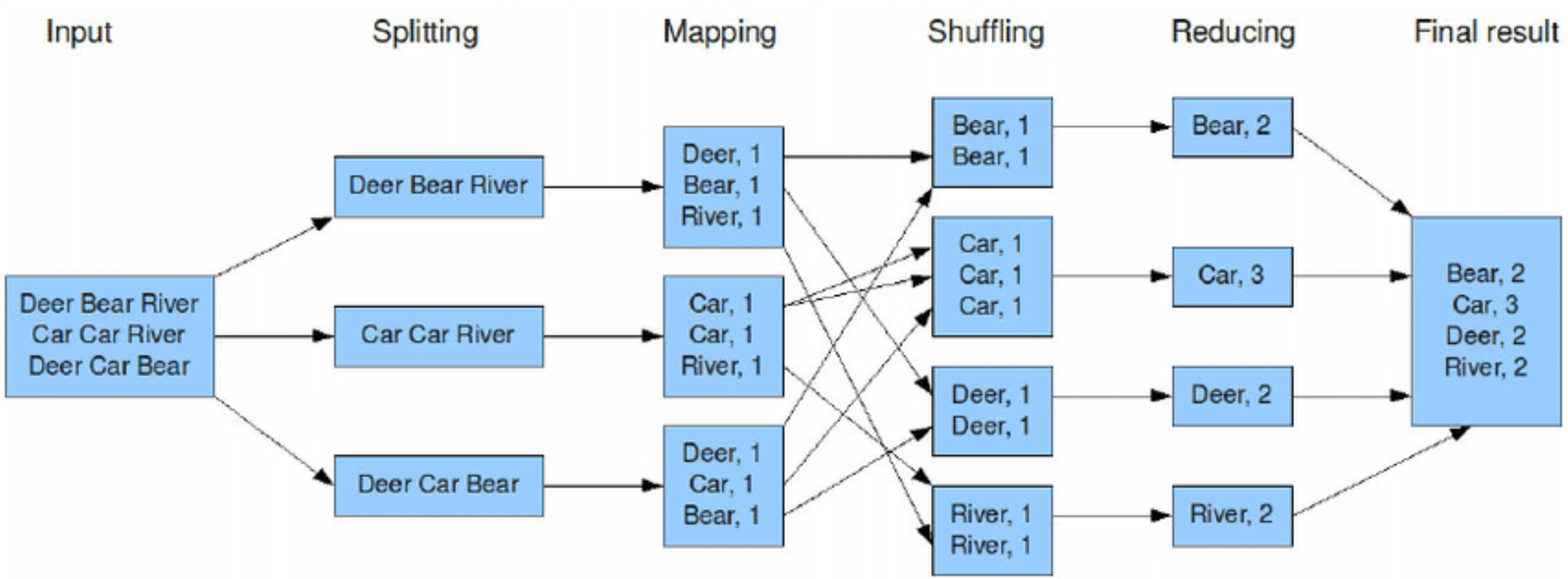




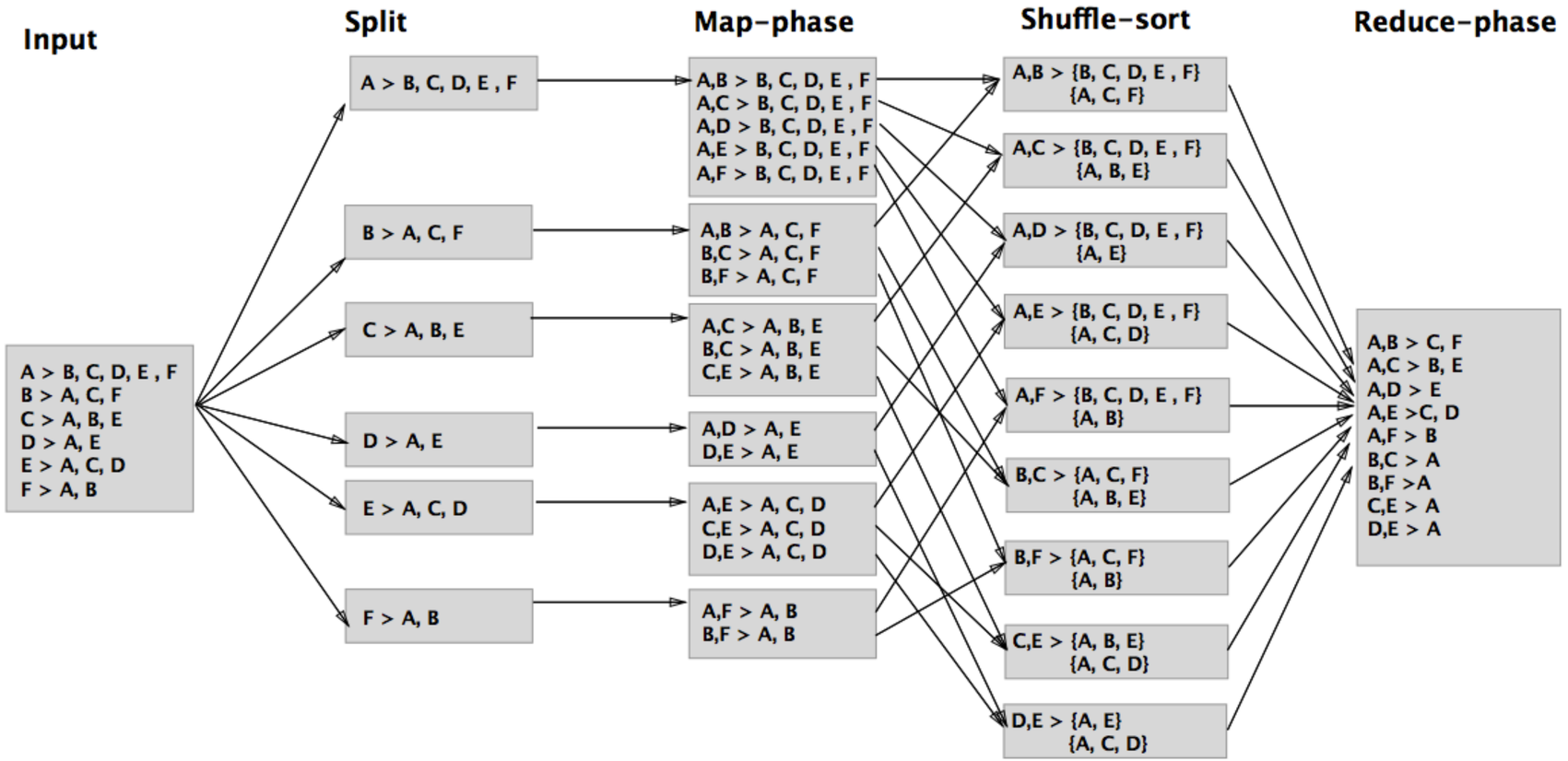


$$4\left(\frac{S}{N}\right) = \pi$$

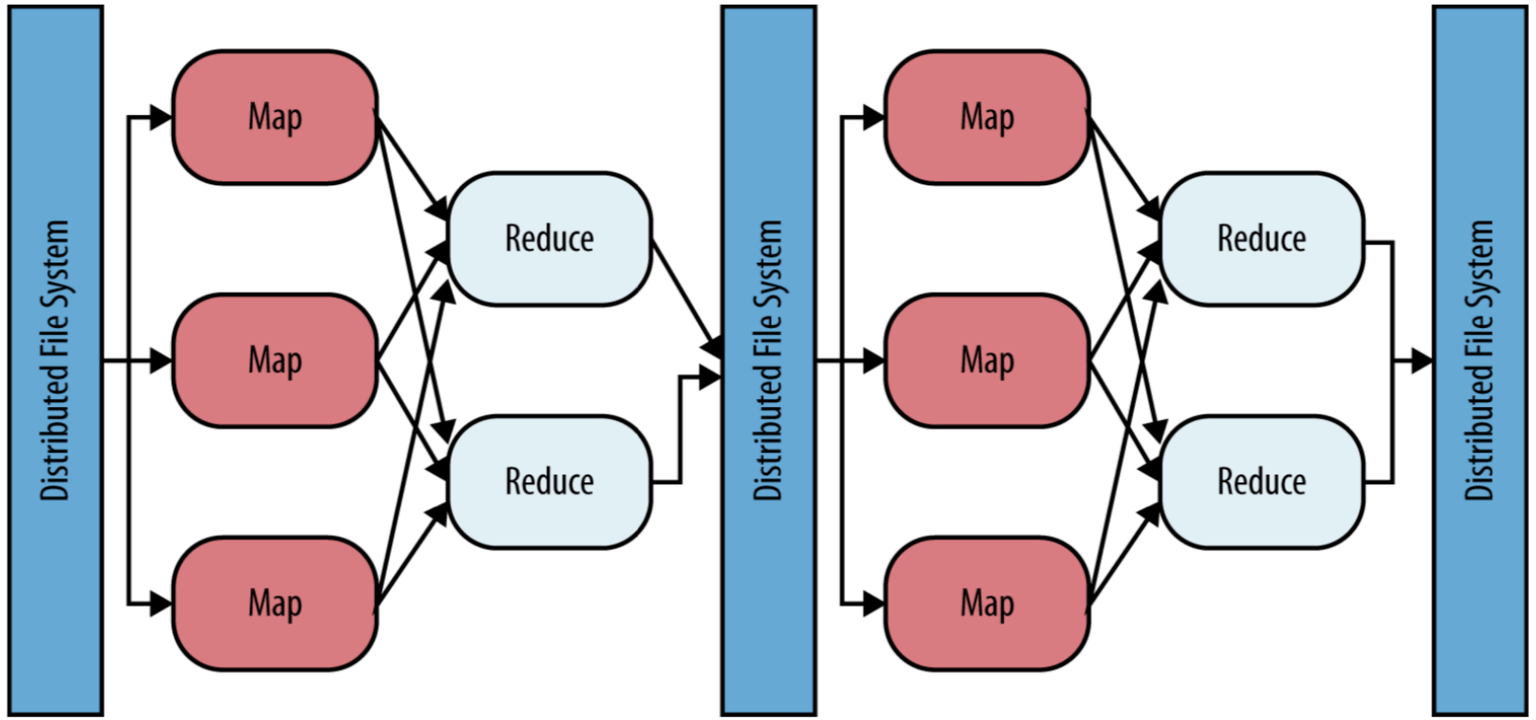
MapReduce: WordCount



MapReduce: Friend-of-friend



MapReduce: Complex Applications




Digital Arts & Science
DIGITECH
SURINAME UNIVERSITY OF TECHNOLOGY