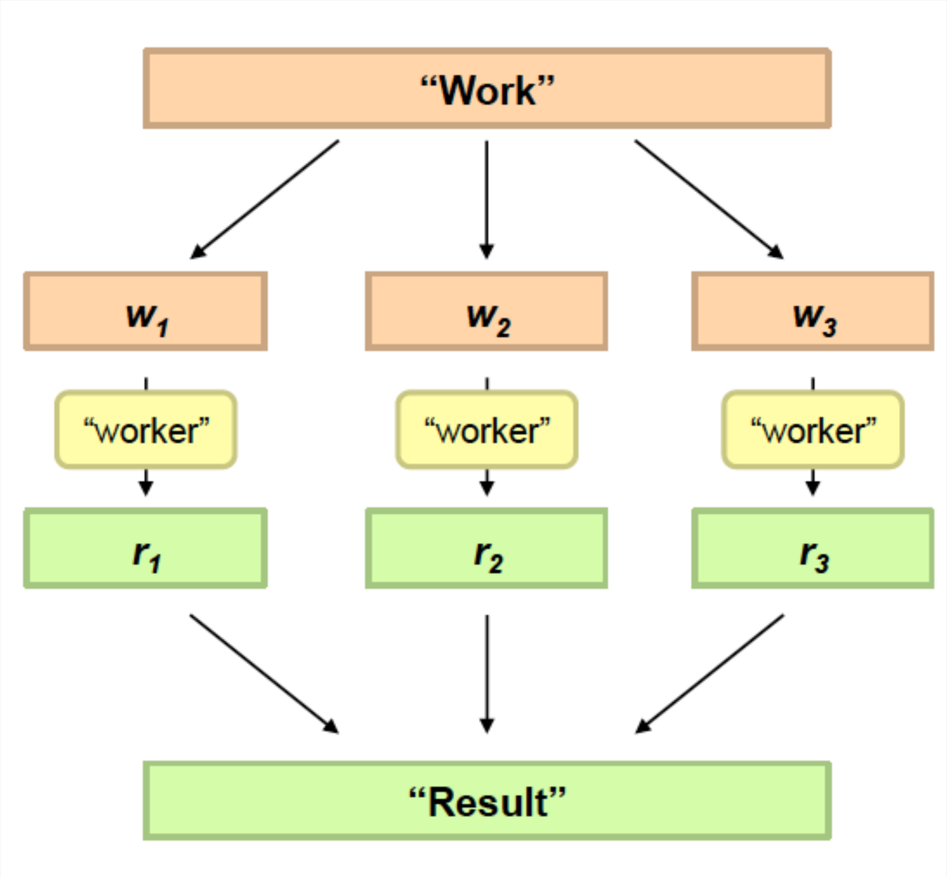
DDAM
INTRODUCTION TO HADOOP

Docente: Patrizio Dazzi

Mail: patrizio.dazzi@isti.cnr.it

- Philosophy to Scale for Big Data?

WORKLOAD DECOMPOSITION



Divide Work

Combine Results

DISTRIBUTED PROCESSING IS NON-TRIVIAL

- How to assign tasks to different workers in an efficient way?
- What happens if tasks fail?
- How do workers exchange results?
- How to synchronize distributed tasks allocated to different workers?

Image courtesy of Master isolated images at FreeDigitalPhotos.net

(PERFORMANT) BIG DATA STORAGE IS CHALLENGING

- Data Volumes are massive
- Reliability of Storing PBs of data is challenging
- All kinds of failures: Disk/Hardware/Network Failures
- Probability of failures simply increase with the number of machines …

- Performance, performance, performance

ONE POPULAR SOLUTION: HADOOP*

Hadoop Cluster at Yahoo! (Credit: Yahoo)

* = but is not the only one

HADOOP* OFFERS

- Redundant, Fault-tolerant data storage
- Parallel computation framework
- Job coordination

* = as well as analogous tools

HADOOP* OFFERS

Programmers

No longer need to worry about

Q: Where file is located?

Q: How to handle failures & data lost?

Q: How to divide computation?

There ain't **no** such thing as a **free lunch**

~~HADOOP IS THE SOLUTION~~

- HADOOP is NOT magic
- Heuristics work… often, not always
- High Performances are not for free
- We wil see in the future how to deal with and implement optimizations

A REAL WORLD EXAMPLE OF NEW YORK TIMES

- **Goal:** Make entire archive of articles available online: 11 million, from 1851
- **Task:** Translate 4 TB TIFF images to PDF files
- **Solution:** Used Amazon Elastic Compute Cloud (EC2) and Simple Storage System (S3)
- **Time: ?**
- **Costs: ?**

A REAL WORLD EXAMPLE OF NEW YORK TIMES

- **Goal:** Make entire archive of articles available online: 11 million, from 1851
- **Task:** Translate 4 TB TIFF images to PDF files
- **Solution:** Used Amazon Elastic Compute Cloud (EC2) and Simple Storage System (S3)
- **Time:** **< 24 hours**
- **Costs:** **$240**

A LITTLE HISTORY ON HADOOP

- Hadoop is an open-source implementation based on **Google File System** (GFS) and **MapReduce** from Google
- Hadoop was created by **Doug Cutting** and **Mike Cafarella** in 2005
- Hadoop was donated to **Apache** in 2006

WHO ARE USING HADOOP?

Social

twitter

Twitter for Business

Small business advertisers, learn how to get $100 in free Twitter advertising from American Express.

ABCs

Learn the Basics

What is Twitter?

Twitter Glossary

Best Practices

Twitter on the Go

Optimize

User Tracking & Engagement

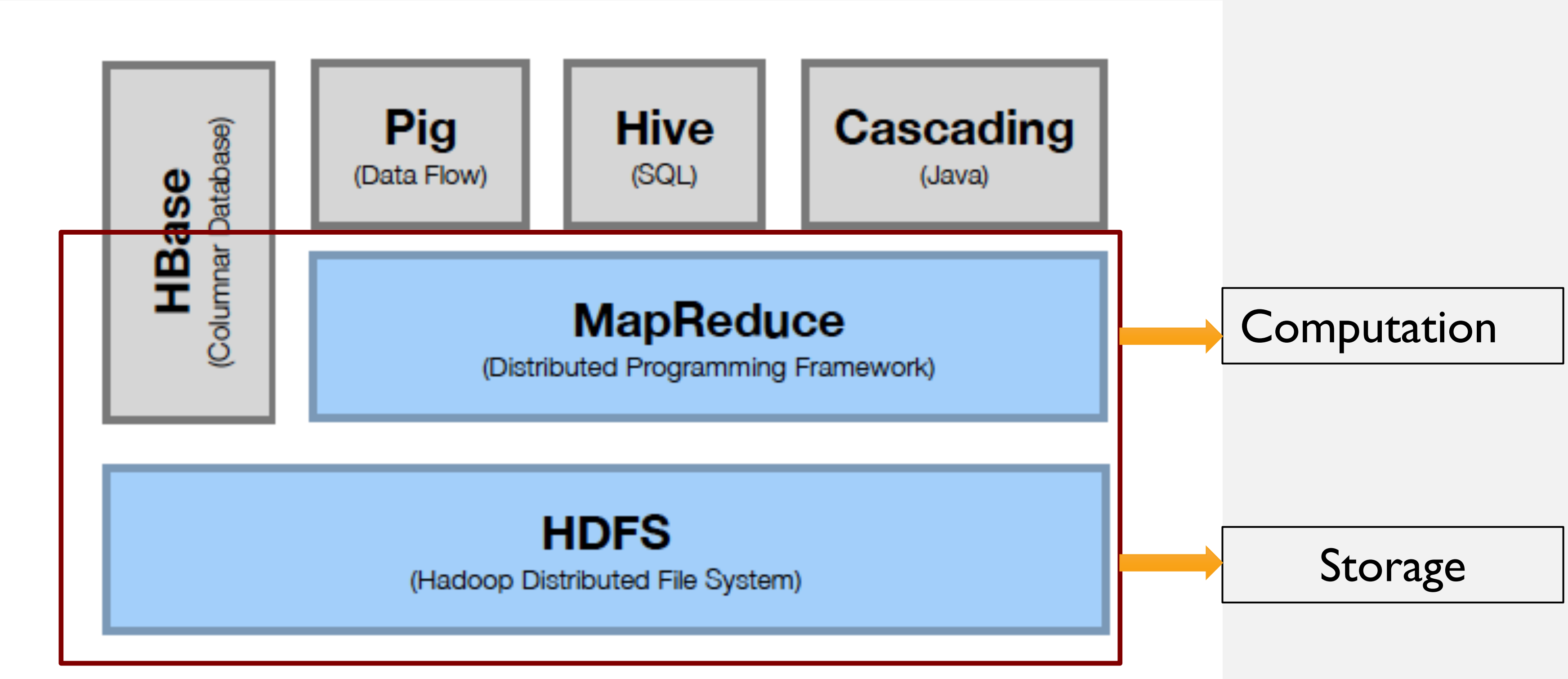
Homeland Security

eCommerce

Financial Services

Real Time Search

HADOOP STACK



HADOOP RESOURCES

- Hadoop at ND:

http://ccl.cse.nd.edu/operations/hadoop/

- Apache Hadoop Documentation:

http://hadoop.apache.org/docs/current/

- Data Intensive Text Processing with Map-Reduce

http://lintool.github.io/MapReduceAlgorithms/

- Hadoop Definitive Guide:

http://www.amazon.com/Hadoop-Definitive-Guide-Tom-White/dp/1449311520

HDFS

HADOOP DISTRIBUTED FILE SYSTEM

MOTIVATION QUESTIONS

- **Problem 1:** Data is too big to store on one machine.

- **HDFS:** Store the data on multiple machines!

MOTIVATION QUESTIONS

- **Problem 2:** Very high end machines are too expensive

- **HDFS:** Run on commodity hardware!

MOTIVATION QUESTIONS

- **Problem 3:** Commodity hardware will fail!
- **HDFS:** Software is intelligent enough to handle hardware failure!

MOTIVATION QUESTIONS

- **Problem 4:** What happens to the data if the machine stores the data fails?

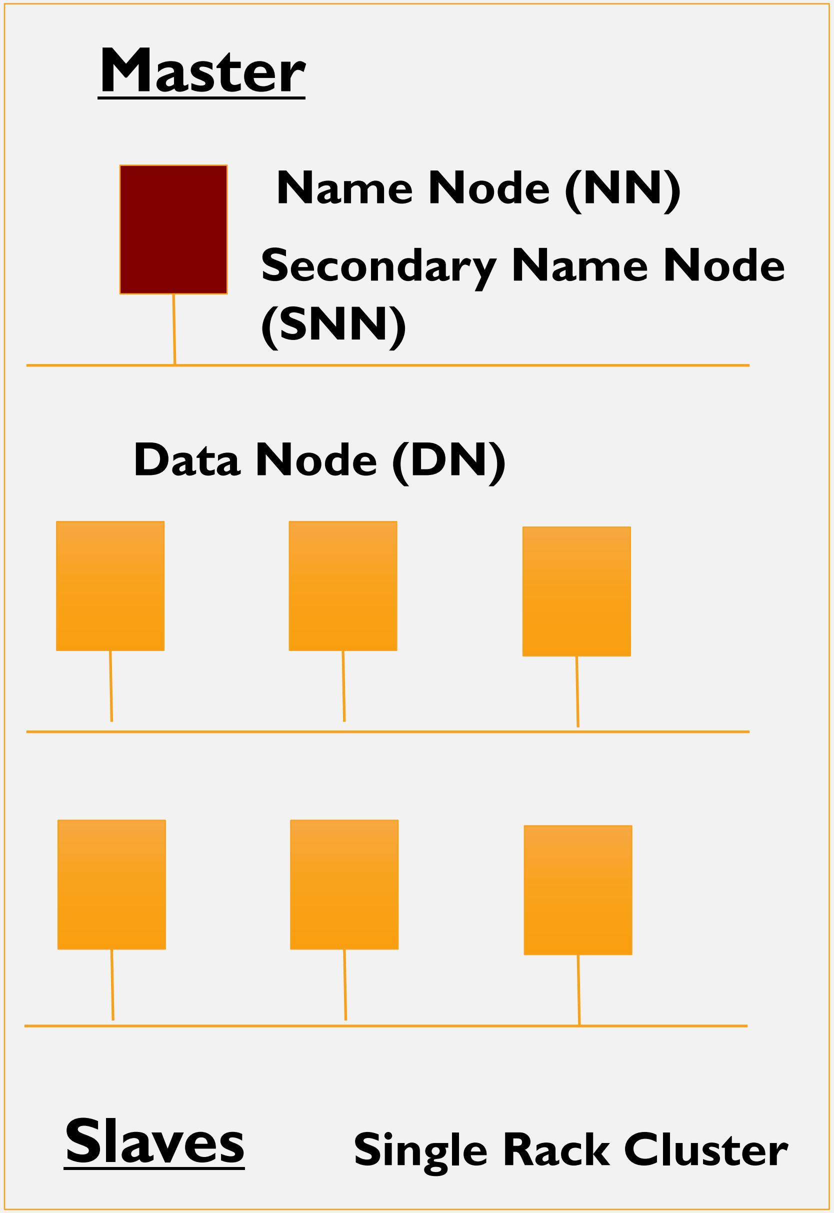
- **HDFS:** Replicate the data!

MOTIVATION QUESTIONS

- **Problem 5:** How can distributed machines organize the data in a coordinated way?

- **HDFS:** Master-Slave Architecture!

HDFS ARCHITECTURE: MASTER-SLAVE



Master

Name Node (NN)

Secondary Name Node (SNN)

Data Node (DN)

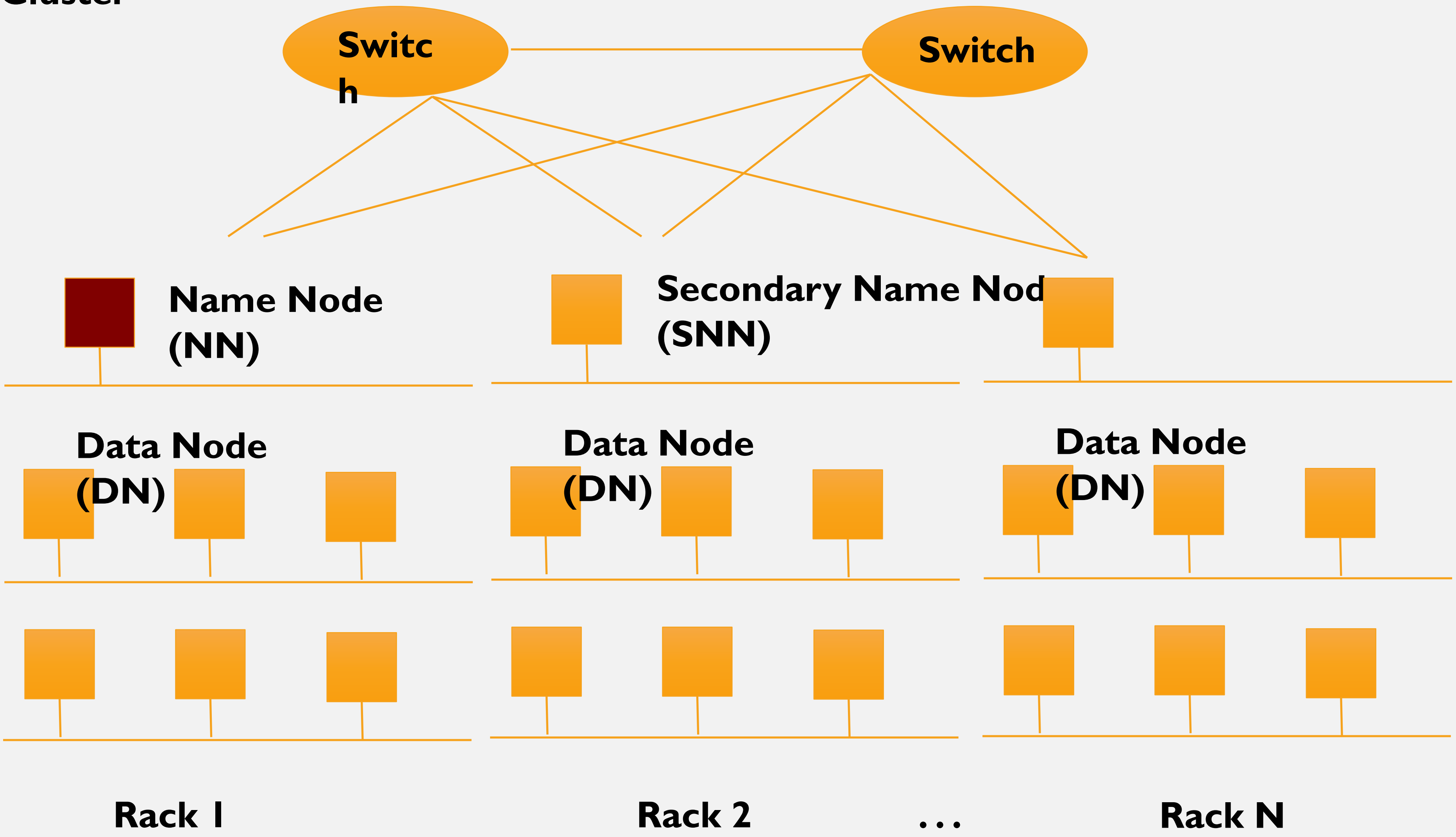
Slaves

Single Rack Cluster

- Name Node: Controller
  - File System Name Space Management
  - Block Mappings
- Data Node: Work Horses
  - Block Operations
  - Replication
- Secondary Name Node:
  - Checkpoint node

HDFS ARCHITECTURE: MASTER-SLAVE

Multiple-Rack Cluster



Switc
h

Switch

Name Node
(NN)

Secondary Name Nod
(SNN)

Data Node
(DN)

Data Node
(DN)

Data Node
(DN)

Rack 1

Rack 2

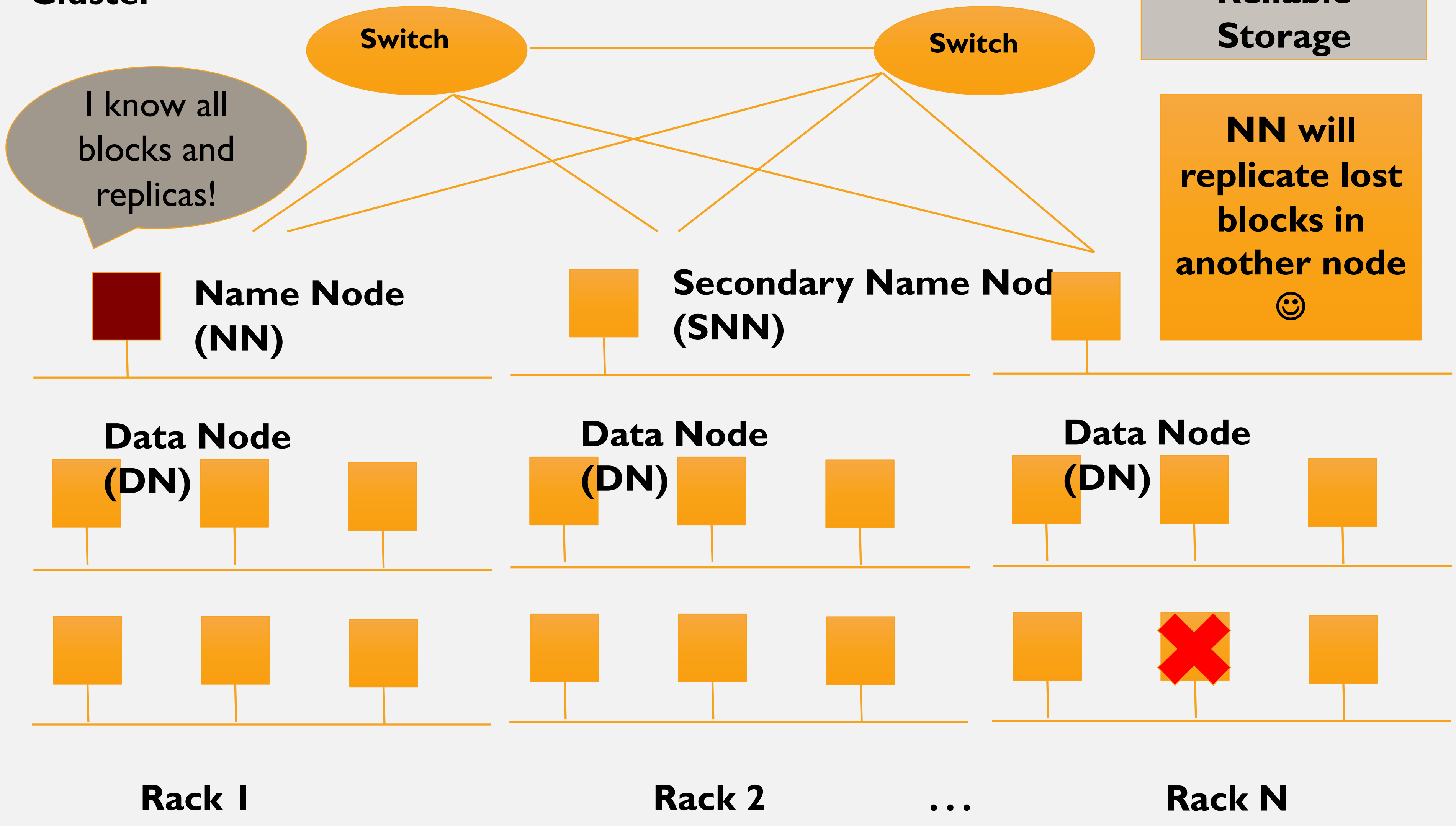
. . .

Rack N

HDFS ARCHITECTURE: MASTER-SLAVE

Multiple-Rack Cluster

Reliable Storage



Switch

Switch

I know all blocks and replicas!

Name Node (NN)

Secondary Name Nod (SNN)

NN will replicate lost blocks in another node ☺

Data Node (DN)

Data Node (DN)

Data Node (DN)

Rack 1

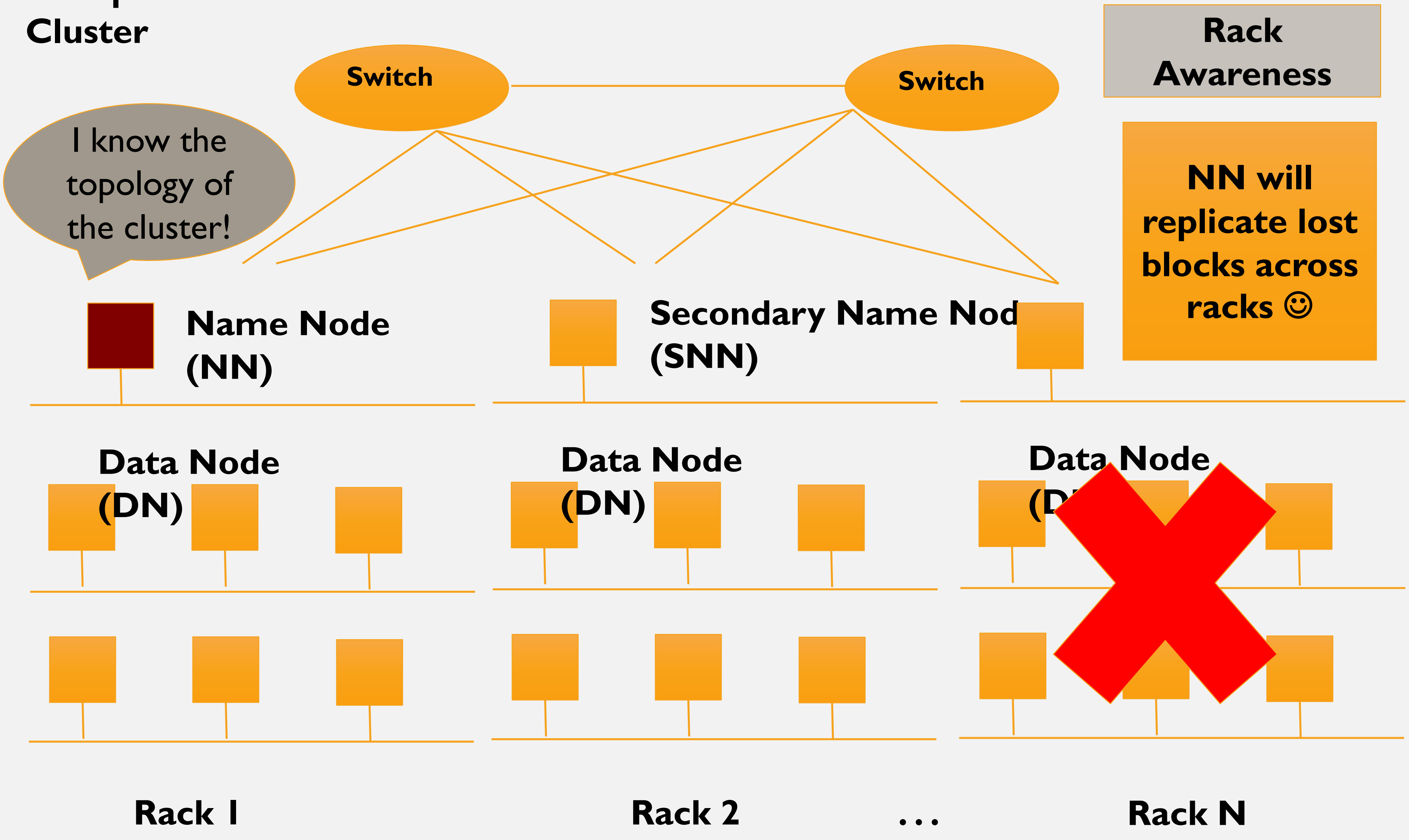
Rack 2

. . .

Rack N

HDFS ARCHITECTURE: MASTER-SLAVE

Multiple-Rack Cluster



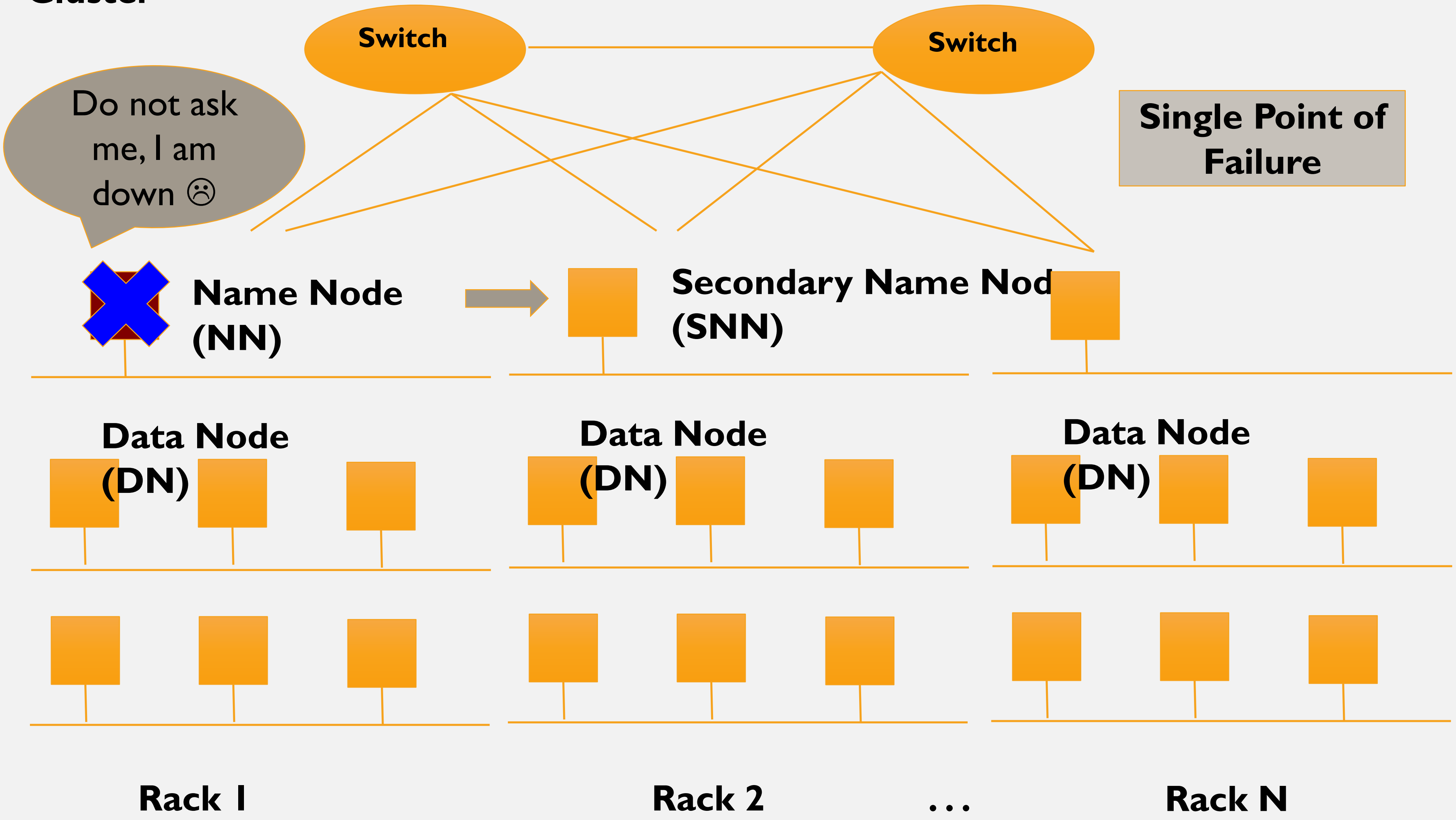
Rack Awareness

NN will replicate lost blocks across racks ☺

I know the topology of the cluster!

HDFS ARCHITECTURE: MASTER-SLAVE

Multiple-Rack Cluster



Switch

Switch

Do not ask me, I am down ☹

Single Point of Failure

Name Node (NN)

Secondary Name Nod (SNN)

Data Node (DN)

Data Node (DN)

Data Node (DN)

Rack 1

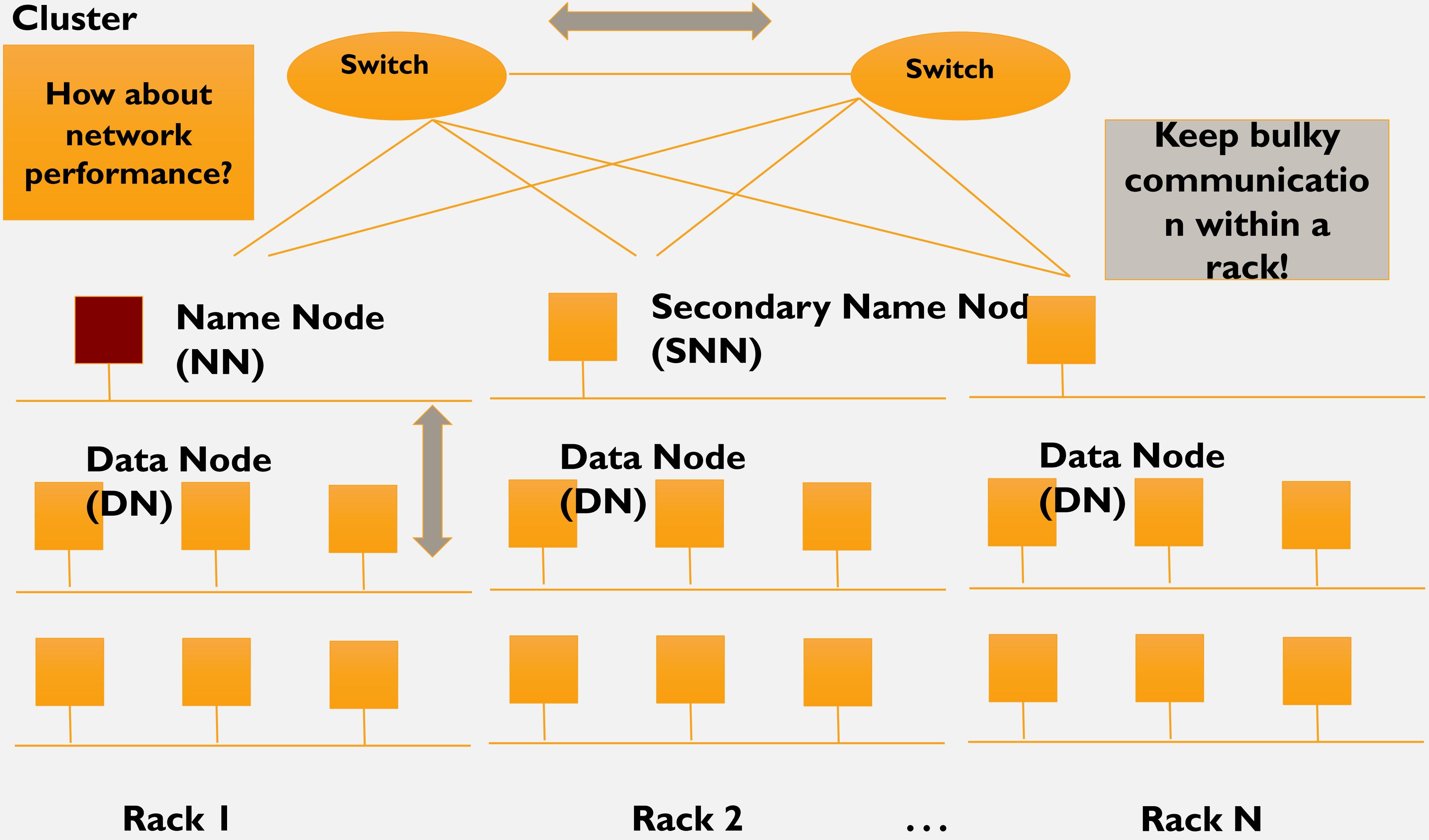
Rack 2

. . .

Rack N

HDFS ARCHITECTURE: MASTER-SLAVE

Multiple-Rack Cluster



How about network performance?

Keep bulky communication within a rack!

HDFS INSIDE: NAME NODE

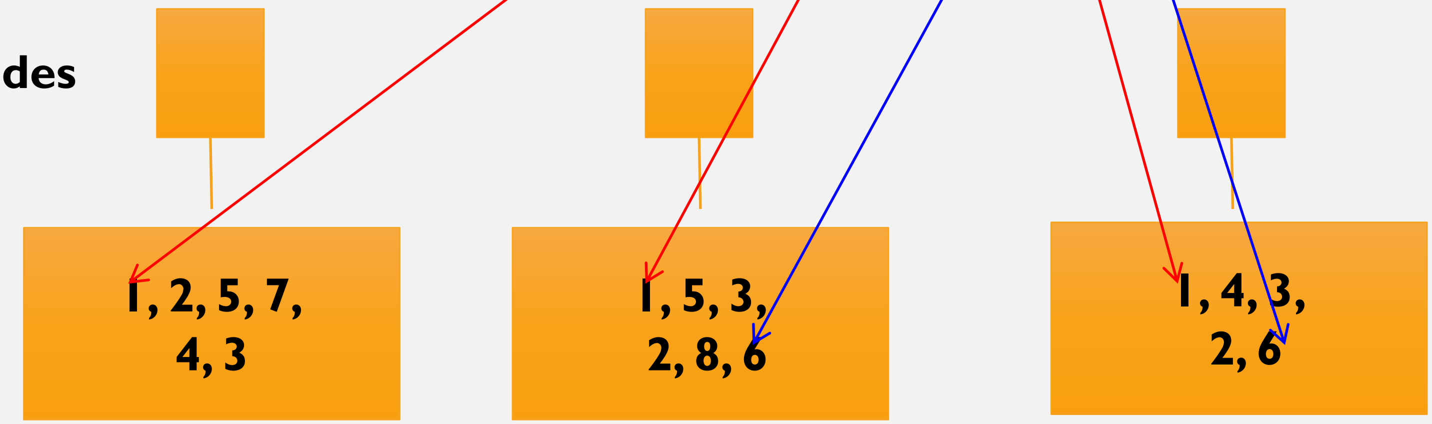
Name Node

Snapshot of FS

Edit log: record changes to FS

Filename	**Replication factor**	**Block ID**
File 1	3	[1, 2, 3]
File 2	2	[4, 5, 6]
File 3	1	[7,8]

Data Nodes



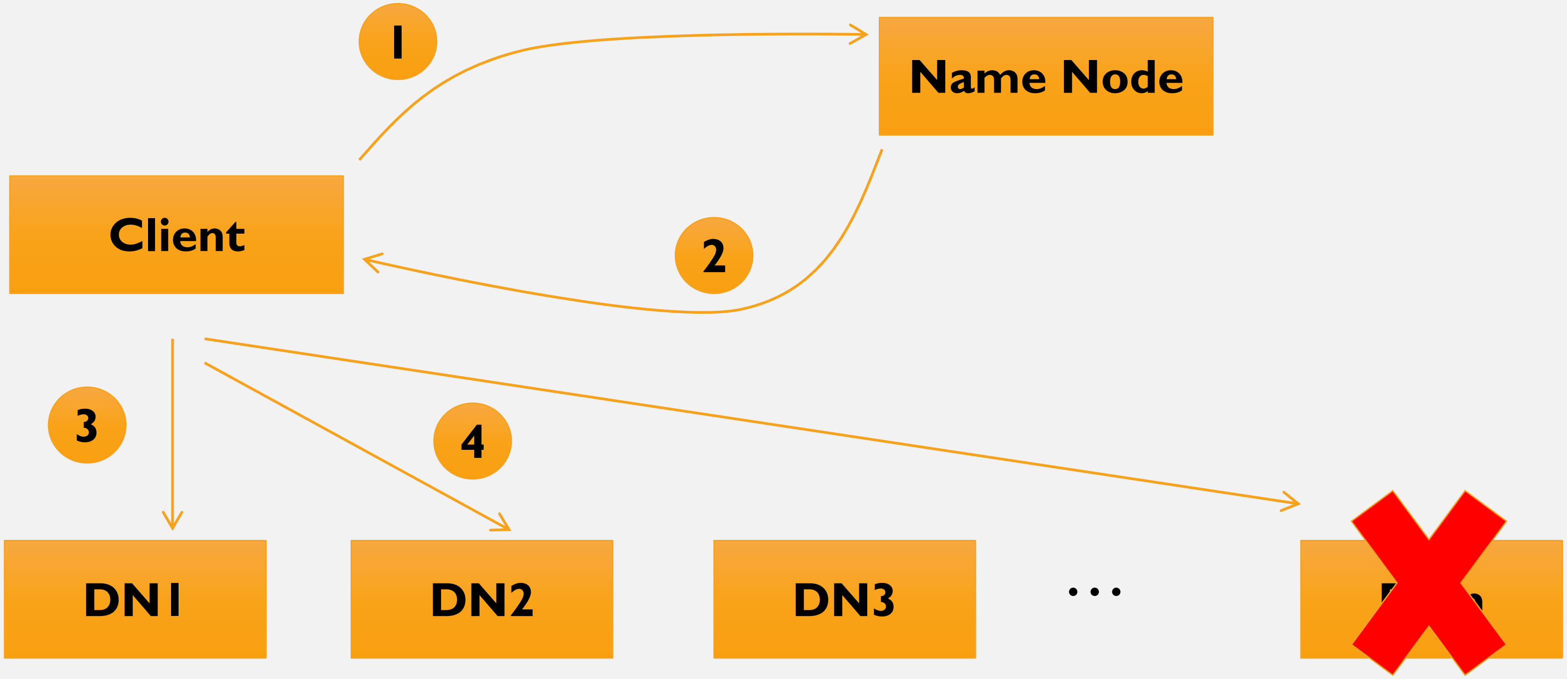
HDFS INSIDE: BLOCKS

- Q:Why do we need the abstraction "Blocks" in addition to "Files"?
- Reasons:
  - File can be larger than a single disk
  - Block is of fixed size, easy to manage and manipulate
  - Easy to replicate and do more fine grained load balancing

HDFS INSIDE: BLOCKS

- HDFS Block size is by default **64 MB**, why it is much larger than regular file system block?
- Reasons:
  - Minimize overhead: disk seek time is almost constant

HDFS INSIDE: READ



1. Client connects to NN to read data
2. NN tells client where to find the data blocks
3. Client reads blocks directly from data nodes (without going through NN)
4. In case of node failures, client connects to another node that serves the missing block

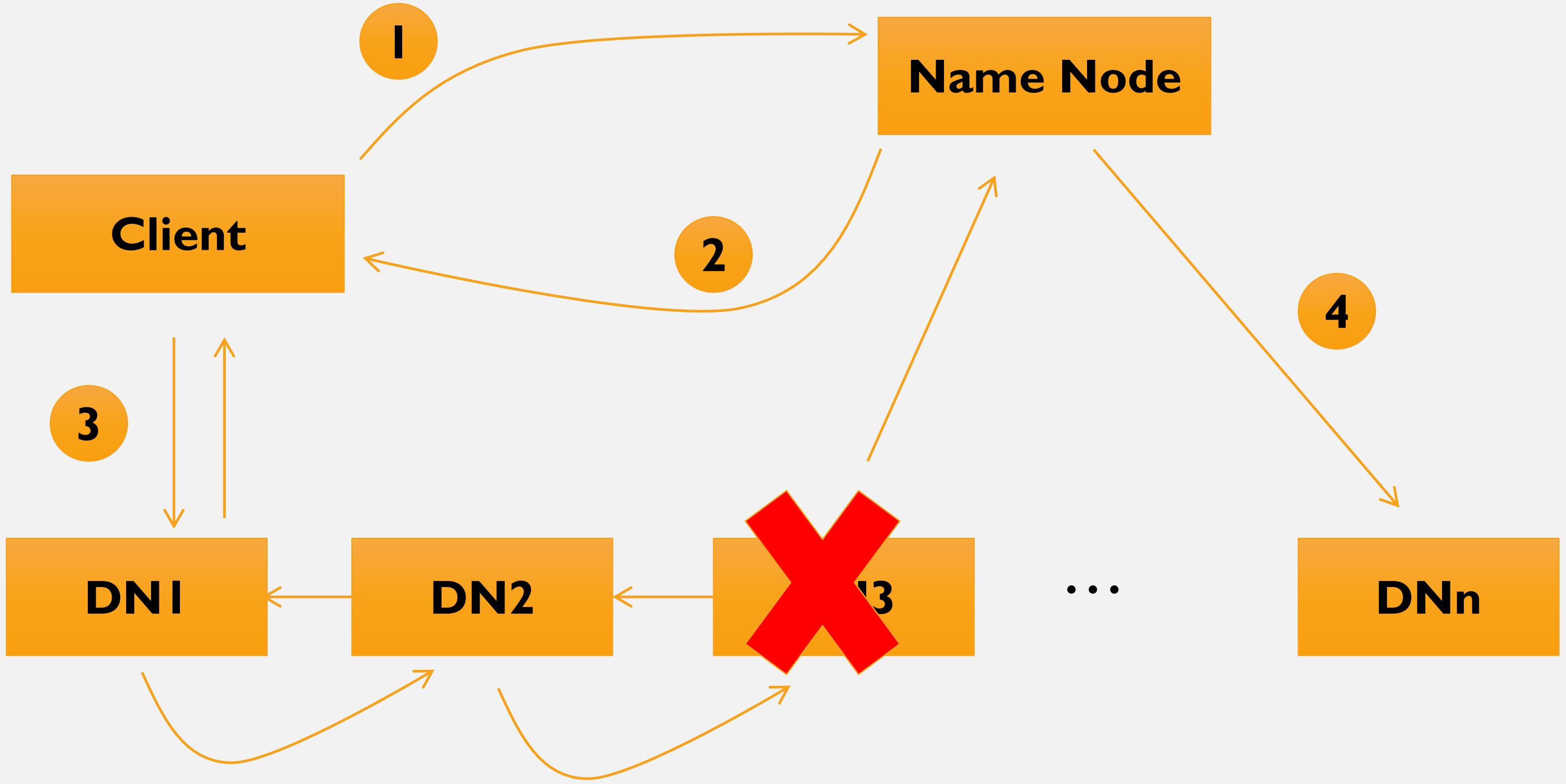
HDFS INSIDE: READ

- Q: Why does HDFS choose such a design for read? Why not ask client to read blocks through NN?
- Reasons:
  - Prevent NN from being the bottleneck of the cluster
  - Allow HDFS to scale to large number of concurrent clients
  - Spread the data traffic across the cluster

HDFS INSIDE: READ

- Q: Given multiple replicas of the same block, how does NN decide which replica the client should read?
- HDFS Solution:
  - Rack awareness based on network topology

HDFS INSIDE: WRITE



1. Client connects to NN to write data
2. NN tells client write these data nodes
3. Client writes blocks directly to data nodes with desired replication factor
4. In case of node failures, NN will figure it out and replicate the missing blocks

HDFS INSIDE: WRITE

- Q: Where should HDFS put the three replicas of a block? What tradeoffs we need to consider?
- Tradeoffs:
  - Reliability
  - Write Bandwidth
  - Read Bandwidth

Q: What are some possible strategies?

HDFS INSIDE: WRITE

- Replication Strategy vs Tradeoffs

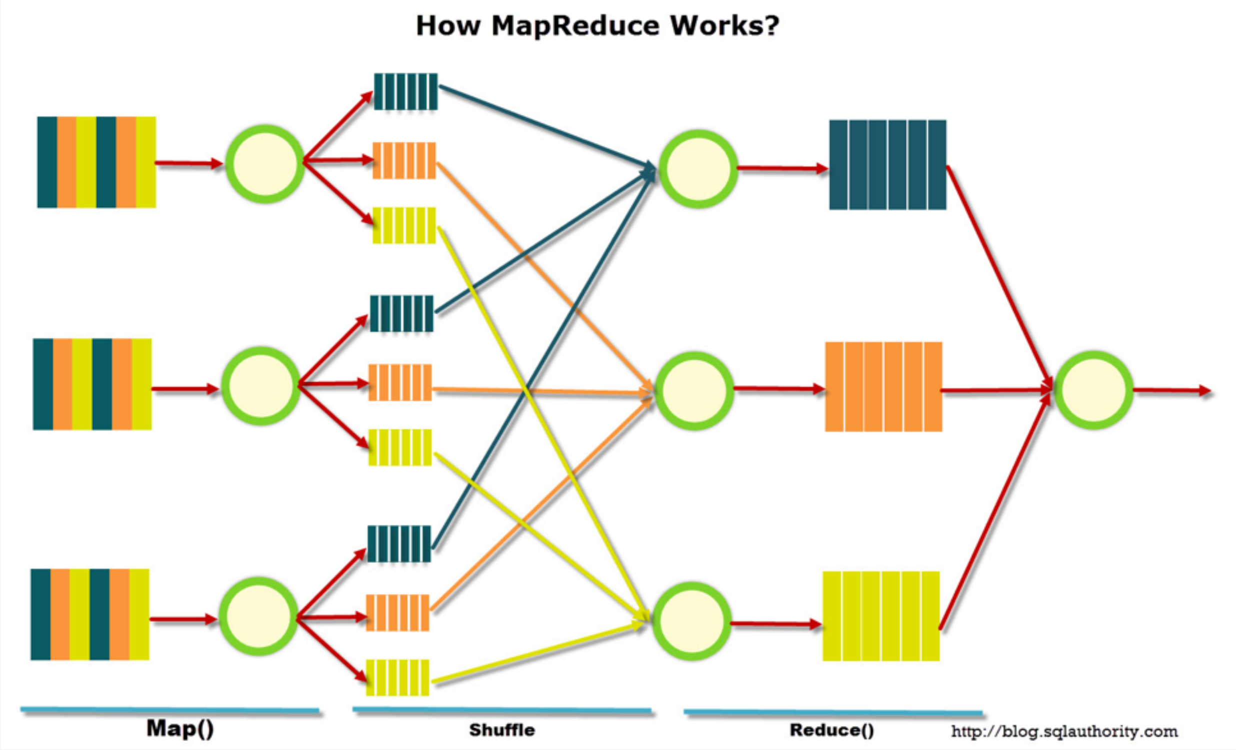
	Reliability	Write Bandwidth	Read Bandwidth
Put all replicas on one node	☹	🙂	☹
Put all replicas on different racks	🙂	☹	☹

HDFS INSIDE: WRITE

- Replication Strategy vs Tradeoffs

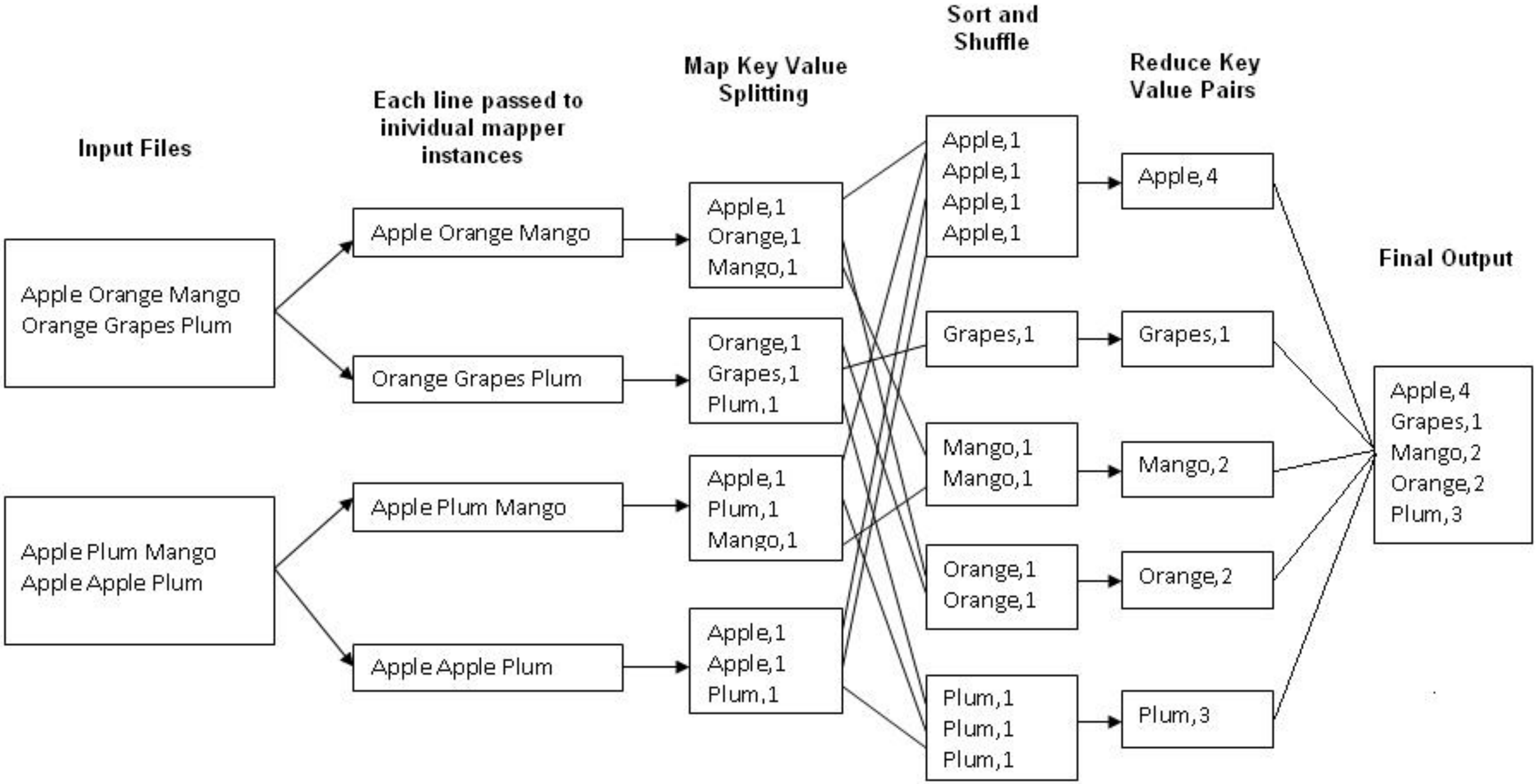
	Reliability	Write Bandwidth	Read Bandwidth
Put all replicas on one node	☹	☺	☹
Put all replicas on different racks	☺	☹	☹
HDFS: 1-> same node as client 2-> a node on different rack 3-> a different node on the same rack as 2	☺	OK	OK

MAPREDUCE



Map
extract something you
care about from each
record

Reduce
aggregate,
summarize, filter,
or transform



Input Files

Apple Orange Mango
Orange Grapes Plum

Apple Plum Mango
Apple Apple Plum

Each line passed to inividual mapper instances

Apple Orange Mango

Orange Grapes Plum

Apple Plum Mango

Apple Apple Plum

Map Key Value Splitting

Apple,1
Orange,1
Mango,1

Orange,1
Grapes,1
Plum,1

Apple,1
Plum,1
Mango,1

Apple,1
Apple,1
Plum,1

Sort and Shuffle

Apple,1
Apple,1
Apple,1
Apple,1

Grapes,1

Mango,1
Mango,1

Orange,1
Orange,1

Plum,1
Plum,1
Plum,1

Reduce Key Value Pairs

Apple,4

Grapes,1

Mango,2

Orange,2

Plum,3

Final Output

Apple,4
Grapes,1
Mango,2
Orange,2
Plum,3

MAPPER

- Reads in input pair <Key,Value>
- Outputs a pair <K', V'>
  - Let's count number of each word in user queries (or Tweets/Blogs)
  - The input to the mapper will be <queryID, QueryText>:

    ```
    <Q1,"The teacher went to the store. The store was closed; the store
    opens in the morning. The store opens at 9am." >
    ```

  - The output would be:

    ```
    <The, 1> <teacher, 1> <went, 1> <to, 1>
    <the, 1> <store,1> <the, 1> <store, 1>
    <was, 1> <closed, 1> <the, 1> <store,1>
    <opens, 1> <in, 1> <the, 1> <morning,
    1> <the 1> <store, 1> <opens, 1> <at,
    1> <9am, 1>
    ```

REDUCER

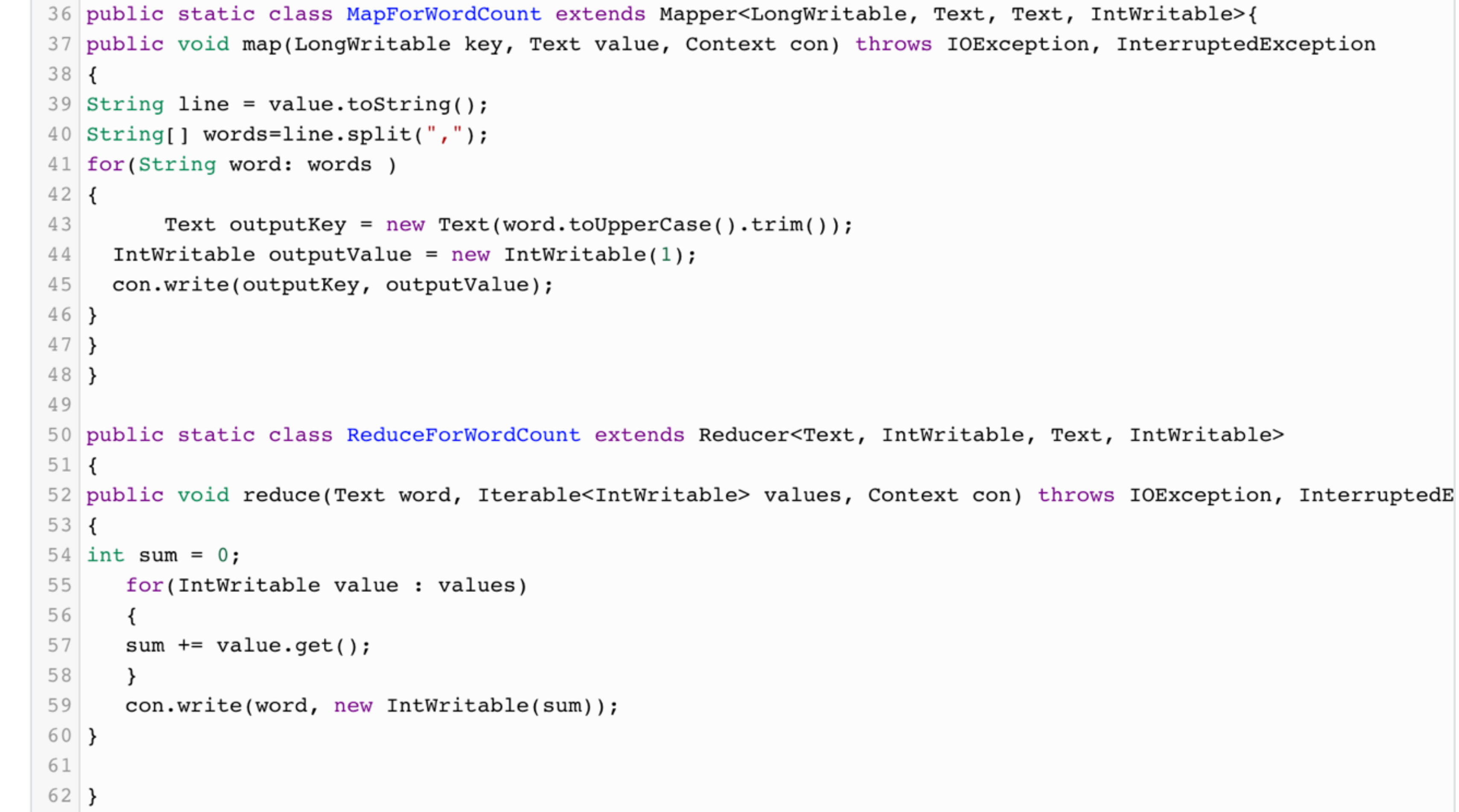
- Accepts the Mapper output, and aggregates values on the key
  - For our example, the reducer input would be:

    <store, 1> <store, 1> <store, 1><store, 1>

  - The output would be:

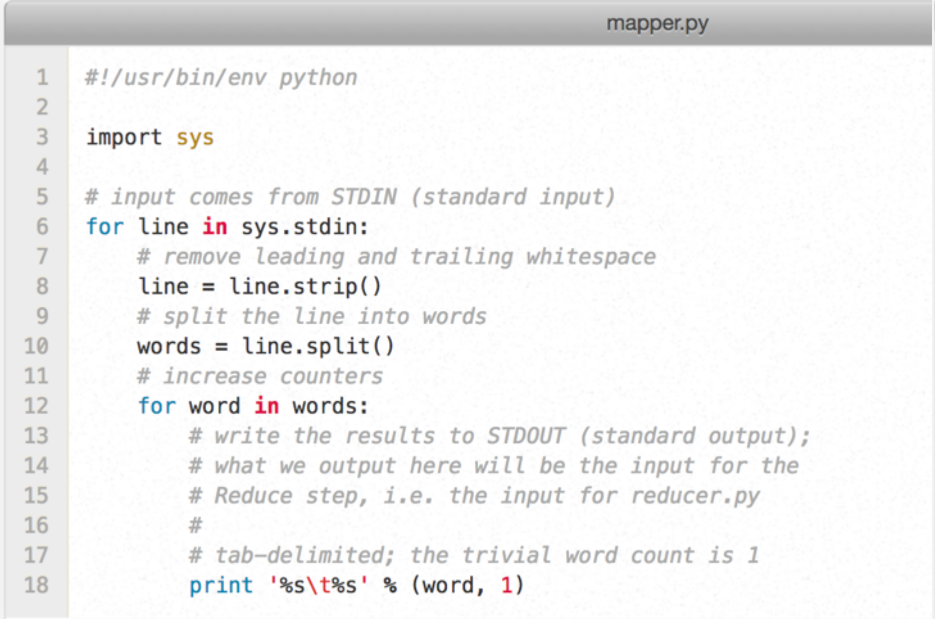
    <store, 4>

JAVA MAP-REDUCE



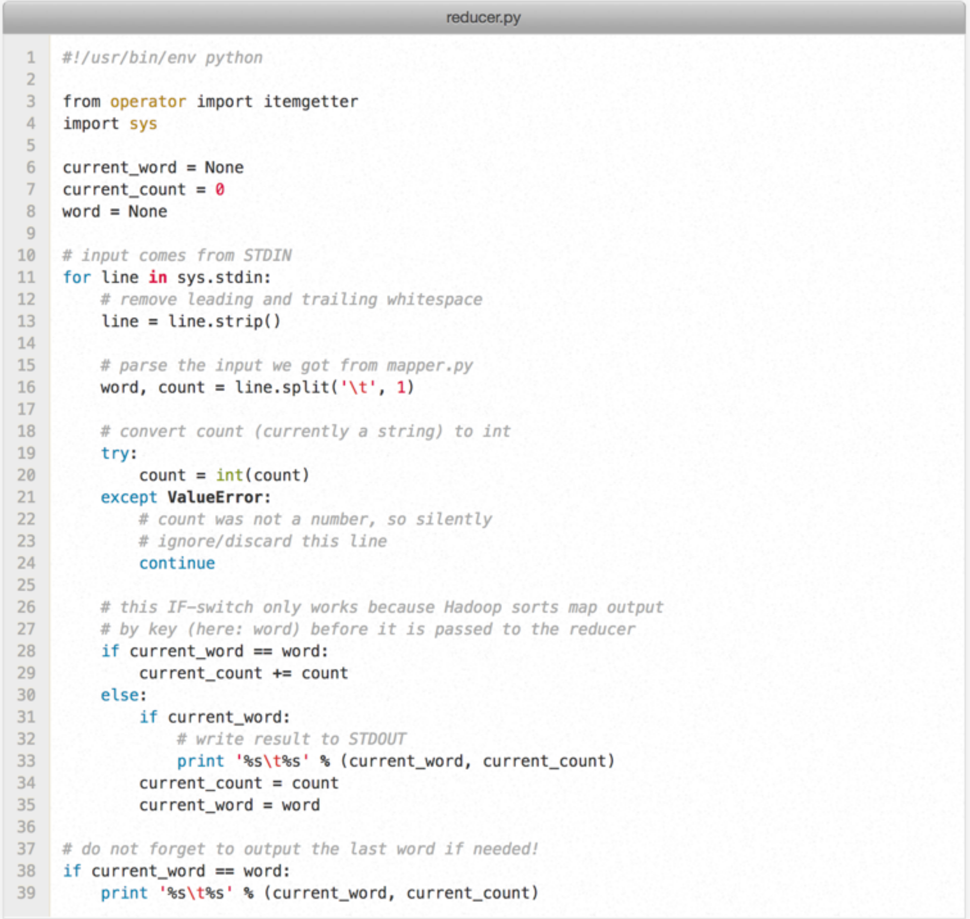
```java
public static class MapForWordCount extends Mapper<LongWritable, Text, Text, IntWritable>{
public void map(LongWritable key, Text value, Context con) throws IOException, InterruptedException
{
String line = value.toString();
String[] words=line.split(",");
for(String word: words )
{
      Text outputKey = new Text(word.toUpperCase().trim());
  IntWritable outputValue = new IntWritable(1);
  con.write(outputKey, outputValue);
}
}
}

public static class ReduceForWordCount extends Reducer<Text, IntWritable, Text, IntWritable>
{
public void reduce(Text word, Iterable<IntWritable> values, Context con) throws IOException, InterruptedE
{
int sum = 0;
   for(IntWritable value : values)
   {
   sum += value.get();
   }
   con.write(word, new IntWritable(sum));
}

}
```

PYTHON MAP-REDUCE



mapper.py

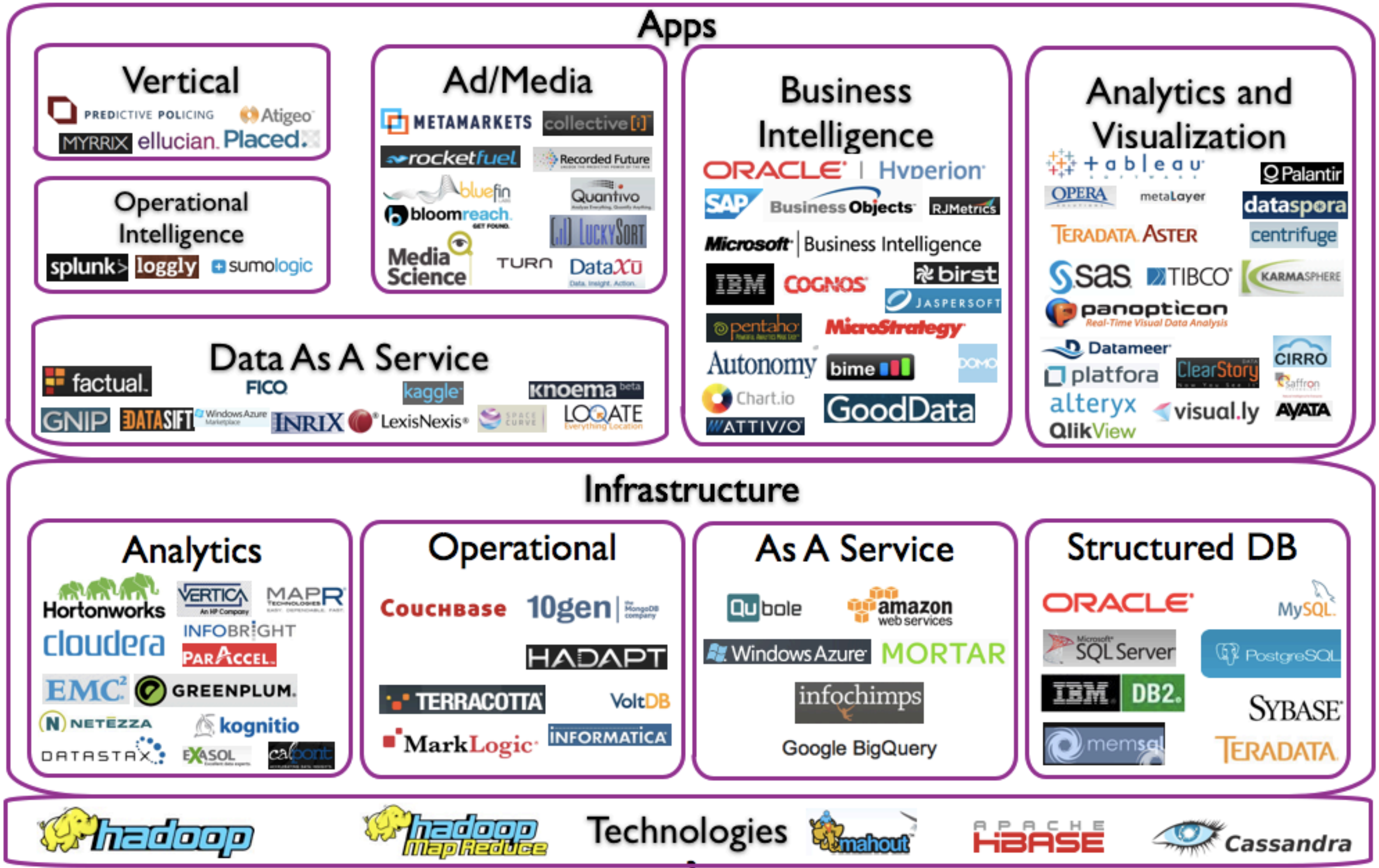
```python
#!/usr/bin/env python

import sys

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    # increase counters
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        print '%s\t%s' % (word, 1)
```



reducer.py

```python
#!/usr/bin/env python

from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to STDOUT
            print '%s\t%s' % (current_word, current_count)
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
    print '%s\t%s' % (current_word, current_count)
```

The Big Data Landscape



Apps

Vertical

PREDICTIVE POLICING, Atigeo, MYRRIX, ellucian., Placed.

Operational Intelligence

splunk>, loggly, sumologic

Ad/Media

METAMARKETS, collective[i], rocketfuel, Recorded Future, bluefin, Quantivo, bloomreach, Lucky Sort, Media Science, TURN, DataXu

Business Intelligence

ORACLE | Hyperion, SAP BusinessObjects, RJMetrics, Microsoft | Business Intelligence, IBM, COGNOS, birst, JASPERSOFT, pentaho, MicroStrategy, Autonomy, bime, DOMO, Chart.io, ATTIVIO, GoodData

Analytics and Visualization

tableau, Palantir, OPERA, metaLayer, dataspora, TERADATA ASTER, centrifuge, sas, TIBCO, KARMASPHERE, panopticon, Real-Time Visual Data Analysis, Datameer, CIRRO, platfora, ClearStory, Saffron, alteryx, visual.ly, AYATA, QlikView

Data As A Service

factual., FICO, kaggle, knoema beta, GNIP, DATASIFT, Windows Azure Marketplace, INRIX, LexisNexis, SPACE CURVE, LOQATE

Infrastructure

Analytics

Hortonworks, VERTICA An HP Company, MAPR, cloudera, INFOBRIGHT, PARACCEL, EMC², GREENPLUM, NETEZZA, kognitio, DATASTAX, EXASOL, calpont

Operational

COUCHBASE, 10gen | the MongoDB company, HADAPT, TERRACOTTA, VoltDB, MarkLogic, INFORMATICA

As A Service

Qubole, amazon web services, Windows Azure, MORTAR, infochimps, Google BigQuery

Structured DB

ORACLE, MySQL, Microsoft SQL Server, PostgreSQL, IBM DB2, SYBASE, memsql, TERADATA

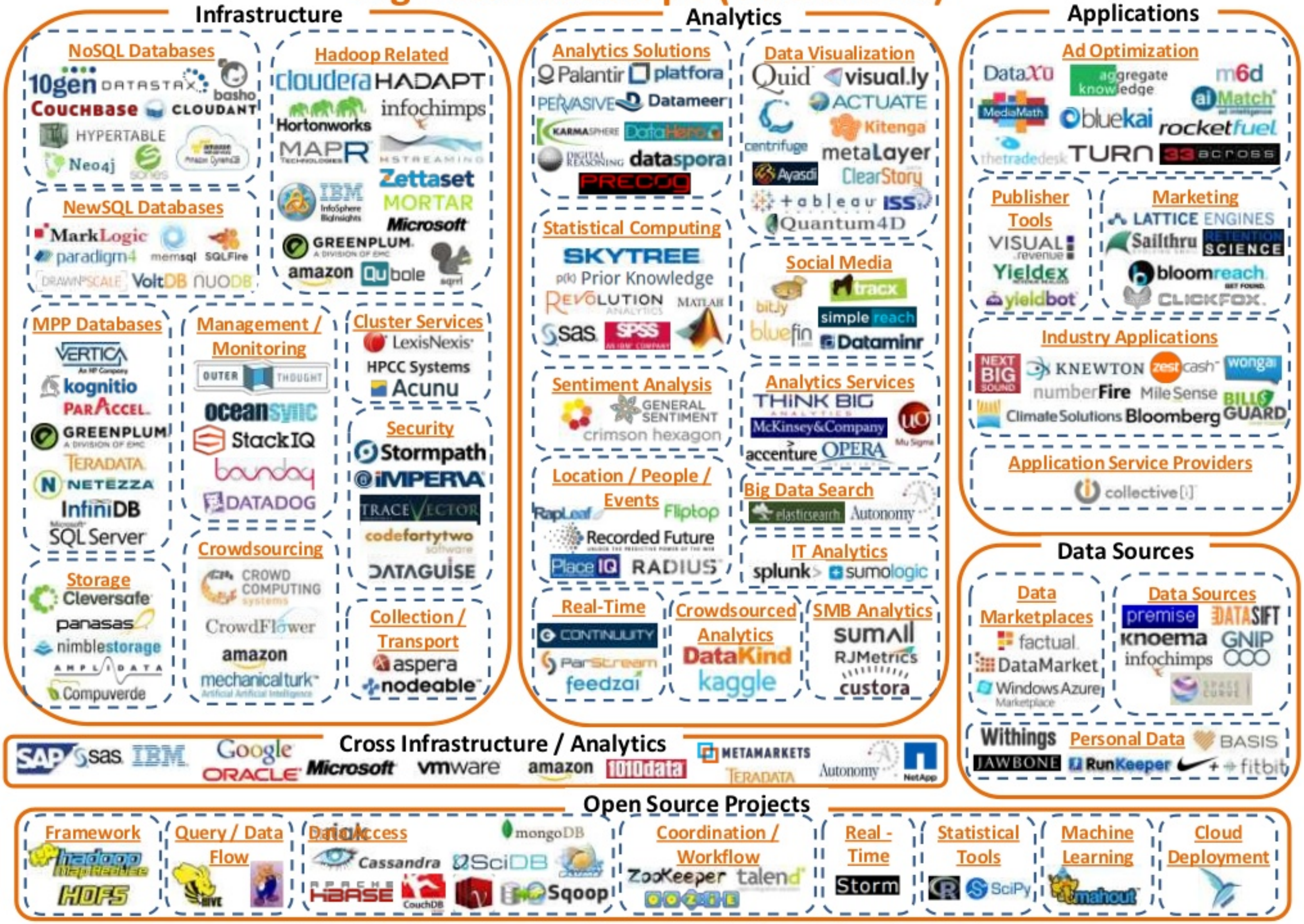
Technologies

hadoop, hadoop MapReduce, mahout, APACHE HBASE, Cassandra

dave@vcdave.com

bigdatalandcsape.com

Big Data Landscape (Version 2.0)



Infrastructure

NoSQL Databases
10gen, DATASTAX, basho, Couchbase, CLOUDANT, HYPERTABLE, Neo4j, sones, Amazon DynamoDB

Hadoop Related
cloudera, HADAPT, Hortonworks, infochimps, MAPR Technologies, HSTREAMING, IBM InfoSphere BigInsights, Zettaset, MORTAR, Microsoft, GREENPLUM A DIVISION OF EMC, amazon, Qubole, sqrrl

NewSQL Databases
MarkLogic, paradigm4, memsql, SQLFire, DRAWNTOSCALE, VoltDB, NUODB

MPP Databases
VERTICA An HP Company, kognitio, ParAccel, GREENPLUM A DIVISION OF EMC, TERADATA, NETEZZA, InfiniDB, Microsoft SQL Server

Management / Monitoring
OUTER THOUGHT, oceansync, StackIQ, boundary, DATADOG

Cluster Services
LexisNexis, HPCC Systems, Acunu

Security
Stormpath, IMPERVA, TRACEVECTOR, codefortytwo software, DATAGUISE

Storage
Cleversafe, panasas, nimblestorage, AMPLIDATA, Compuverde

Crowdsourcing
CROWD COMPUTING systems, CrowdFlower, amazon mechanicalturk Artificial Artificial Intelligence

Collection / Transport
aspera, nodeable

Analytics

Analytics Solutions
Palantir, platfora, PERVASIVE, Datameer, KARMASPHERE, DataHero, DIGITAL REASONING, dataspora, PRECOG

Data Visualization
Quid, visual.ly, ACTUATE, Kitenga, centrifuge, metaLayer, Ayasdi, ClearStory, tableau, ISS, Quantum4D

Statistical Computing
SKYTREE, Prior Knowledge, REVOLUTION ANALYTICS, MATLAB, sas, SPSS AN IBM COMPANY

Social Media
bitly, tracx, simple reach, bluefin, Dataminr

Sentiment Analysis
GENERAL SENTIMENT, crimson hexagon

Analytics Services
THINK BIG ANALYTICS, McKinsey&Company, Mu Sigma, accenture, OPERA

Location / People / Events
RapLeaf, Fliptop, Recorded Future, PlaceIQ, RADIUS

Big Data Search
elasticsearch, Autonomy

IT Analytics
splunk, sumologic

Real-Time
CONTINUUITY, ParStream, feedzai

Crowdsourced Analytics
DataKind, kaggle

SMB Analytics
SUMALL, RJMetrics, custora

Applications

Ad Optimization
DataXu, aggregate knowledge, m6d, MediaMath, bluekai, aiMatch, rocketfuel, thetradedesk, TURN, 33across

Publisher Tools
VISUAL revenue, Yieldex, yieldbot

Marketing
LATTICE ENGINES, Sailthru, RETENTION SCIENCE, bloomreach, CLICKFOX

Industry Applications
NEXT BIG SOUND, KNEWTON, zestcash, wonga, numberFire, MileSense, BILLGUARD, Climate Solutions, Bloomberg

Application Service Providers
collective[i]

Data Sources

Data Marketplaces
factual, DataMarket, Windows Azure Marketplace

Data Sources
premise, DATASIFT, knoema, GNIP, infochimps, SPACE CURVE

Personal Data
Withings, BASIS, JAWBONE, RunKeeper, fitbit

Cross Infrastructure / Analytics
SAP, sas, IBM, Google, ORACLE, Microsoft, vmware, amazon, 1010data, METAMARKETS, TERADATA, Autonomy, NetApp

Open Source Projects

Framework
hadoop MapReduce, HDFS

Query / Data Flow
HIVE

Data Access
mongoDB, Cassandra, SciDB, HBASE, CouchDB, Sqoop

Coordination / Workflow
ZooKeeper, talend, OOZIE

Real - Time
Storm

Statistical Tools
R, SciPy

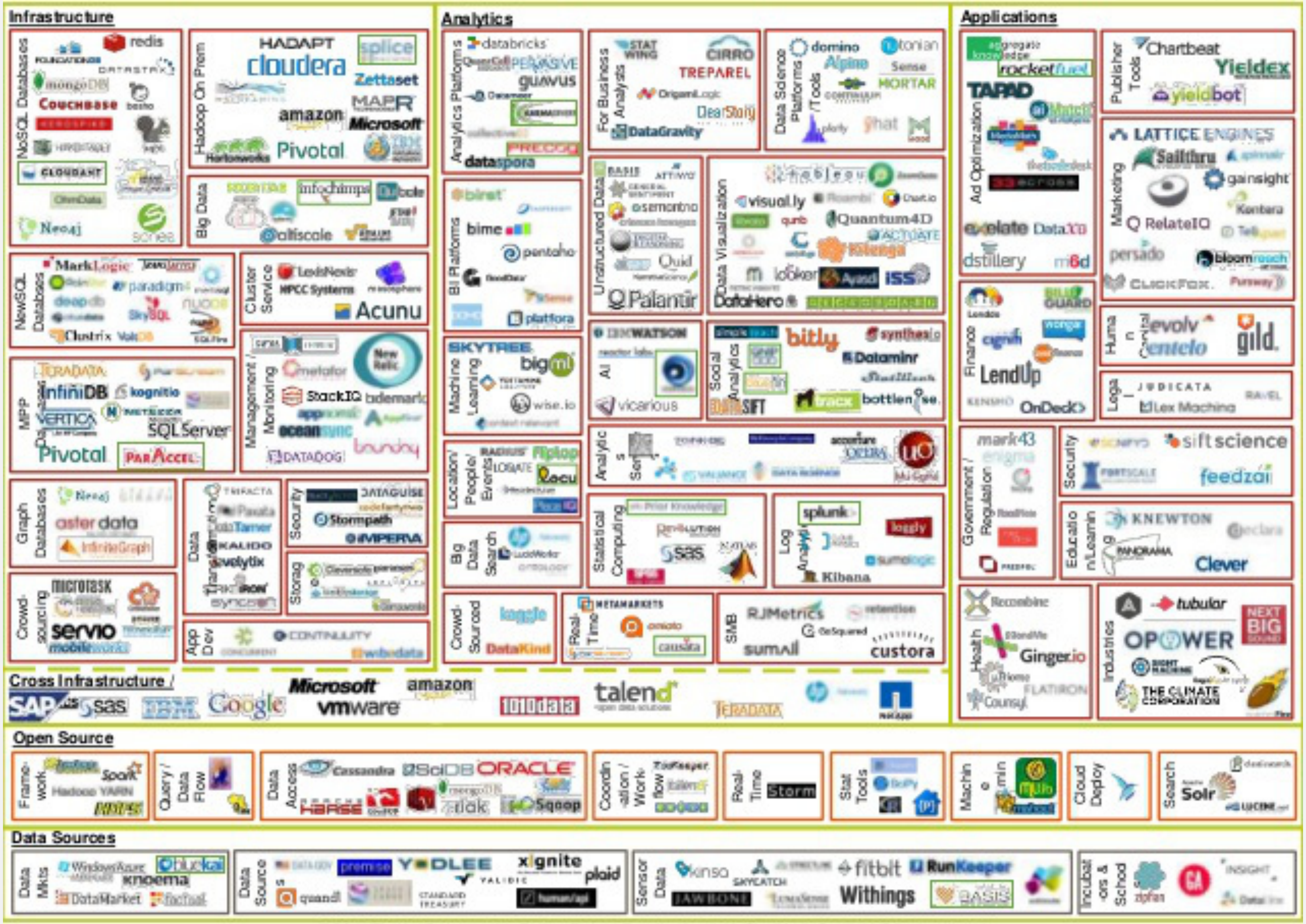
Machine Learning
mahout

Cloud Deployment

© Matt Turck (@mattturck) and ShivonZilis (@shivonz) Bloomberg Ventures

BIG DATA LANDSCAPE, VERSION 3.0

Exited: Acquisition or IPO



Infrastructure

NoSQL Databases

Hadoop On Prem

Big Data

NewSQL Databases

Cluster Service

MPP Databases

Management / Monitoring

Graph Databases

Data Transformation

Security

Crowdsourcing

Storage

App Dev

Analytics

Analytics Platforms

For Business Analysts

Data Science Platforms /Tools

BI Platforms

Unstructured Data

Data Visualization

Machine Learning

AI

Social Analytics

Location/ People/ Events

Analytics Services

Big Data Search

Statistical Computing

Log Analytics

Crowd-Sourced

Real-Time

SMB

Applications

Ad Optimization

Publisher Tools

Marketing

Finance

Human Capital

Legal

Government / Regulation

Security

Education / Learning

Health

Industries

Cross Infrastructure / Analytics

Open Source

Framework

Query / Data Flow

Data Access

Coordination / Workflow

Real-Time

Stat Tools

Machine Learning

Cloud Deploy

Search

Data Sources

Data Mkts

Data Sources

Sensor Data

Incubators & Schools

© Matt Turck (@mattturck), Sutian Dong (@sutiandong) & FirstMark Capital (@firstmarkcap)