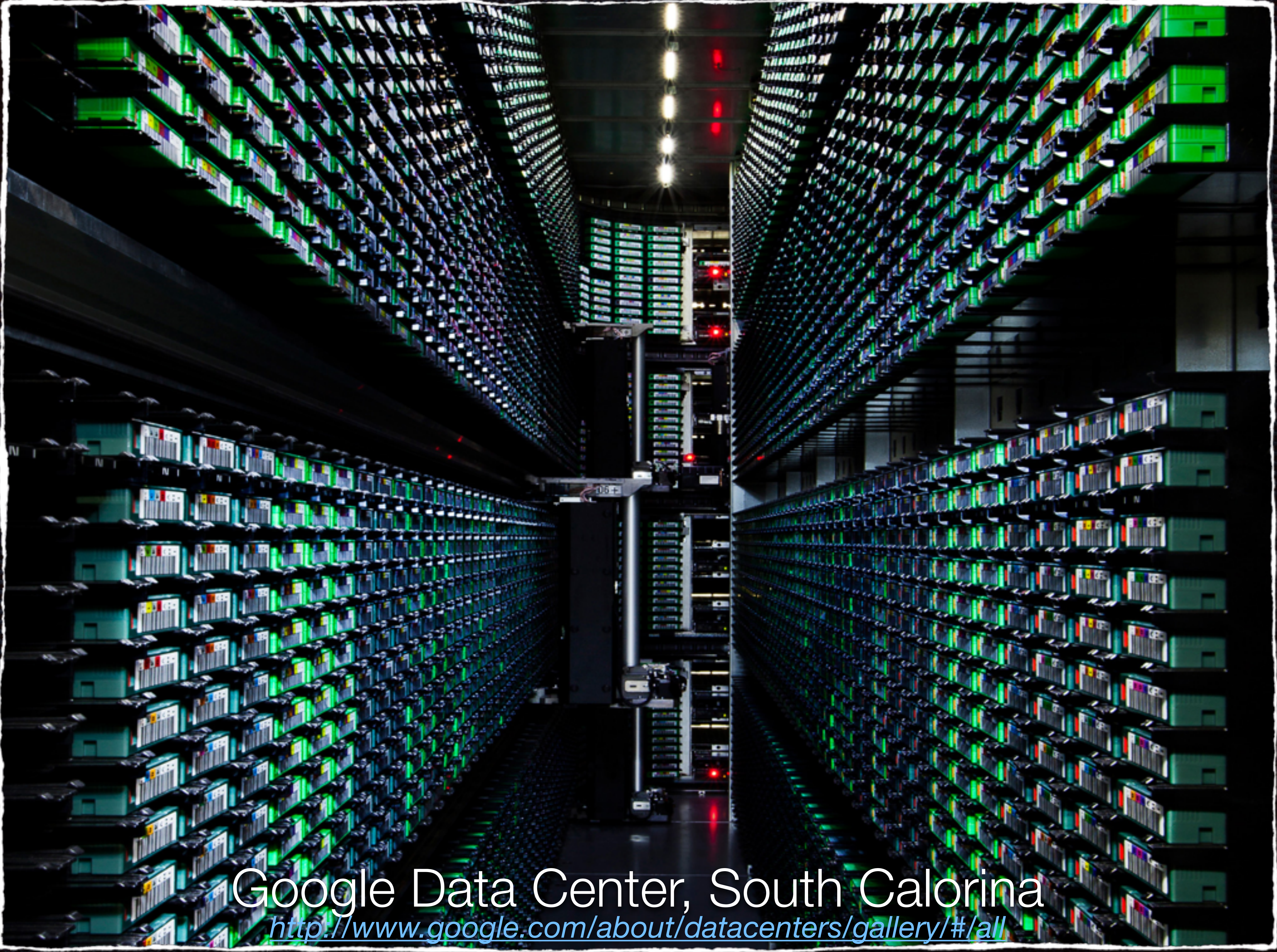


Google Data Center, South Calorina

http://www.google.com/about/datacenters/gallery/#/all

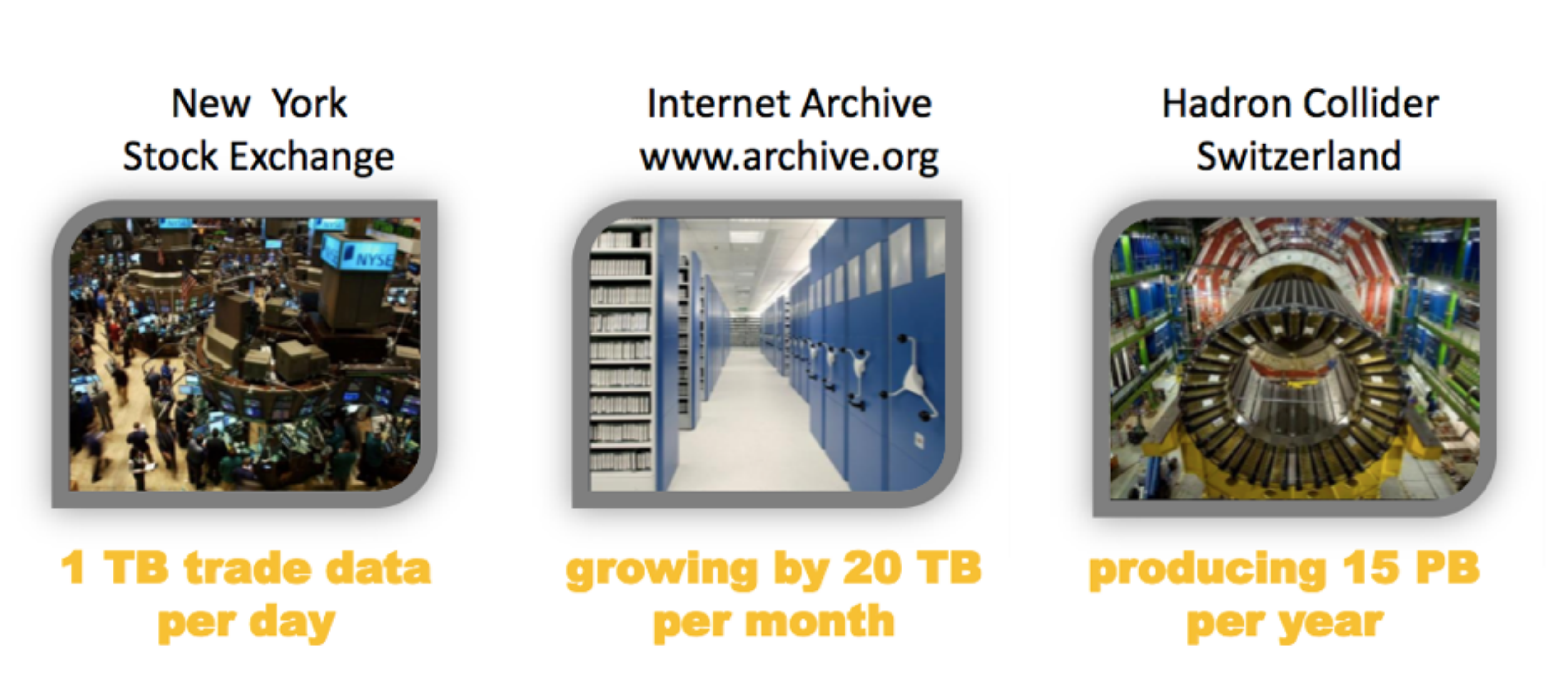
MapReduce: Distributed Computing (the Google Way)

Asst. Prof. Krerk Piromsopa, Ph. D.
Mobile Application and System Services Research Group (MASS)
Department of Computer Engineering,
Chulalongkorn Univeristy

krerk.P@Chula.ac.th

Why do we care?

Why do we care?



New York
Stock Exchange

1 TB trade data per day

Internet Archive
www.archive.org

growing by 20 TB per month

Hadron Collider
Switzerland

producing 15 PB per year

Why do we care?

New York
Stock Exchange

1 TB trade data per day

Internet Archive
www.archive.org

growing by 20 TB per month

Hadron Collider
Switzerland

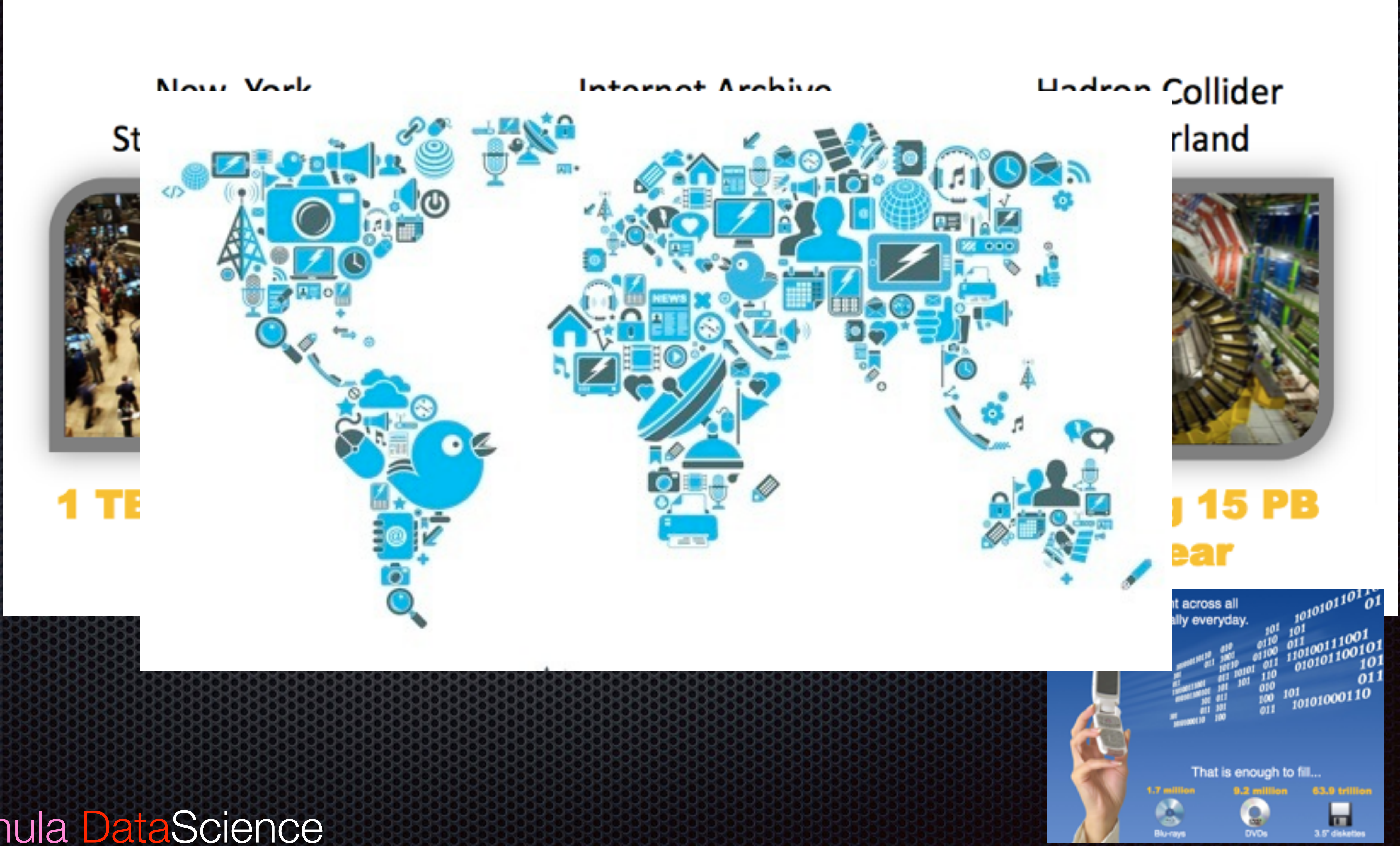
producing 15 PB per year

43.339 TB are sent across all mobile phones globally everyday.

That is enough to fill...

1.7 million	9.2 million	63.9 trillion
Blu-rays	DVDs	3.5" diskettes

Why do we care?



Cost/Performance

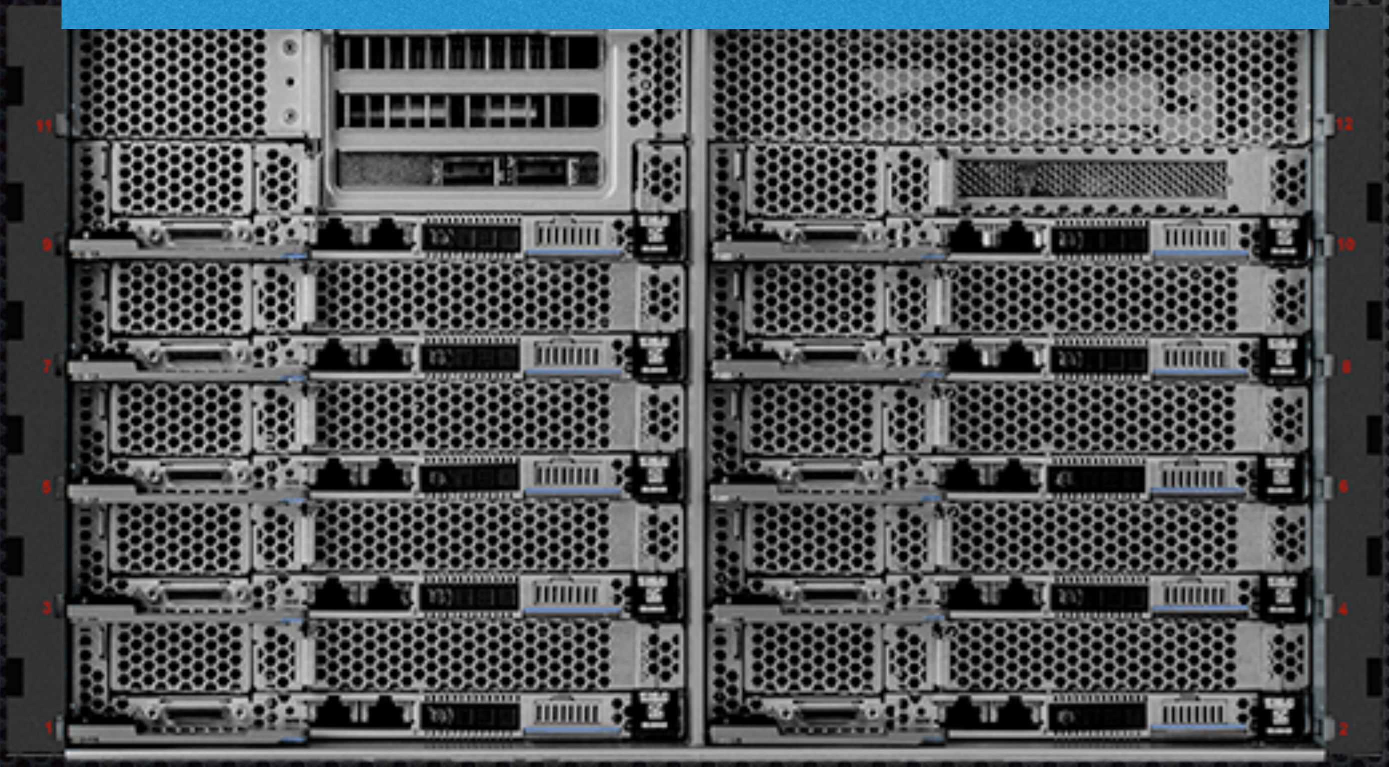
Cost/Performance

Cost/Performance

IBM NextScale Server M5

intel XEON (Up to 18 cores)

RAM 512Gb



Cost/Performance

IBM NextScale Server M5
intel XEON (Up to 18 cores)
RAM 512Gb

$14,432.00 USD

Cost/Performance

IBM NextScale Server M5
intel XEON (Up to 18 cores)
RAM 512Gb

$14,432.00 USD

Cost/Performance

IBM NextScale Server M5
intel XEON (Up to 18 cores)
RAM 512Gb

$14,432.00 USD

Intel NUC
intel Core i5 (4cores)
RAM 16 Gb

Cost/Performance

IBM NextScale Server M5
intel XEON (Up to 18 cores)
RAM 512Gb

$14,432.00 USD

Intel NUC
intel Core i5 (4cores)
RAM 16 Gb

$700.00 USD

Cost/Performance

IBM NextScale Server M5
intel XEON (Up to 18 cores)
RAM 512Gb

$14,432.00 USD

Intel NUC
intel Core i5 (4cores)
RAM 16 Gb

$700.00 USD

20-30 units

Chula DataScience

Cost/Performance

IBM NextScale Server M5
intel XEON (Up to 18 cores)
RAM 512Gb

$14,432.00 USD

Intel NUC
intel Core i5 (4cores)
RAM 16 Gb

$700.00 USD

20-30 units

80 cores
RAM 320 Gb

Hardware vs. Software

IBM NextScale Server M5

- Hardware: Reliable
- Software: easy

Intel NUC farm

- Hardware: Vulnerable
- Software : ????

Time to Product/Analysis

Time to Product/Analysis

Before

Development : weeks
Run: days/month

Time to Product/Analysis

Before

Development : weeks
Run: days/month

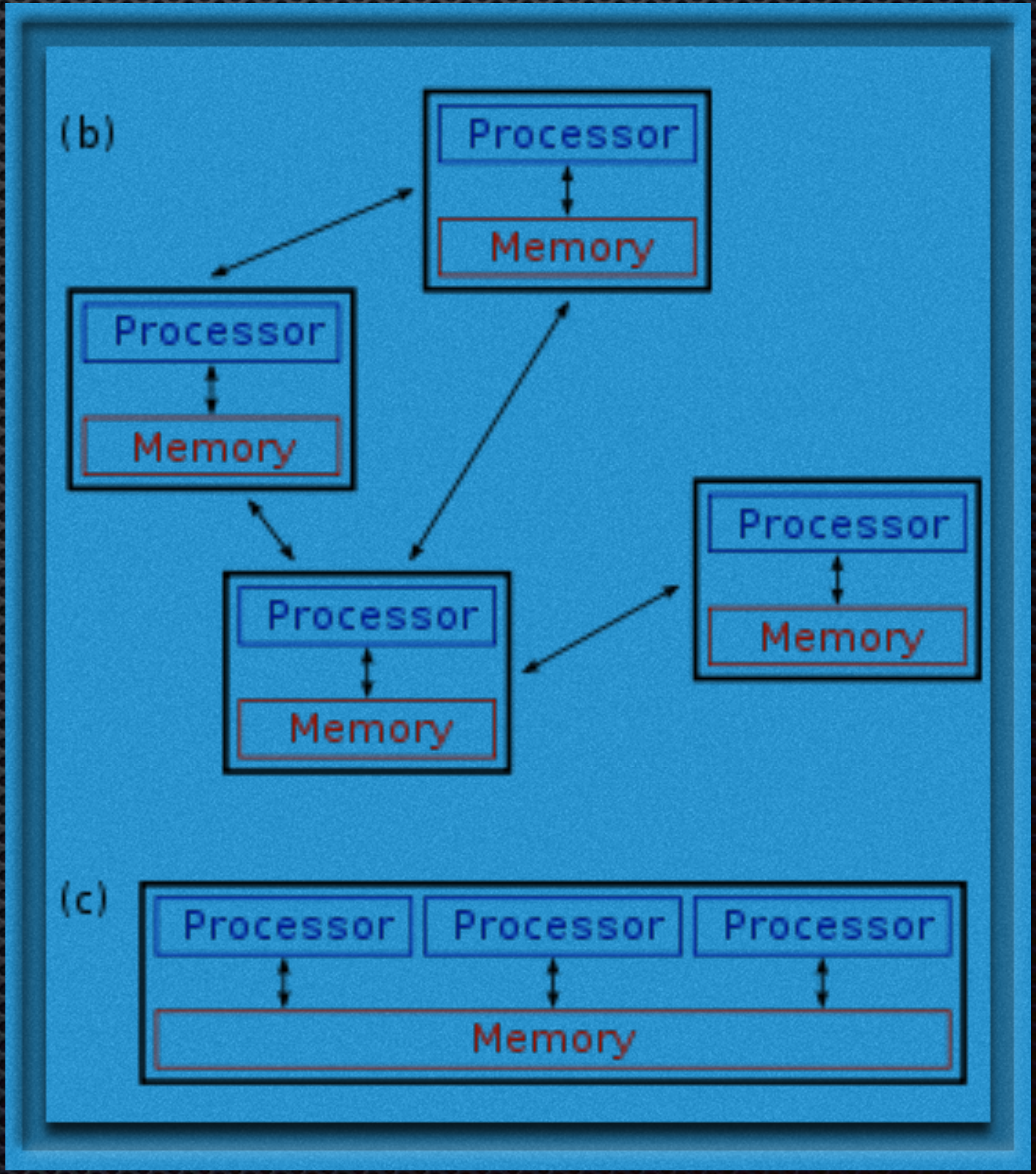
Today

Development : few days
Run: minutes

Why MapReduce?
Why not Grid/Cluster?

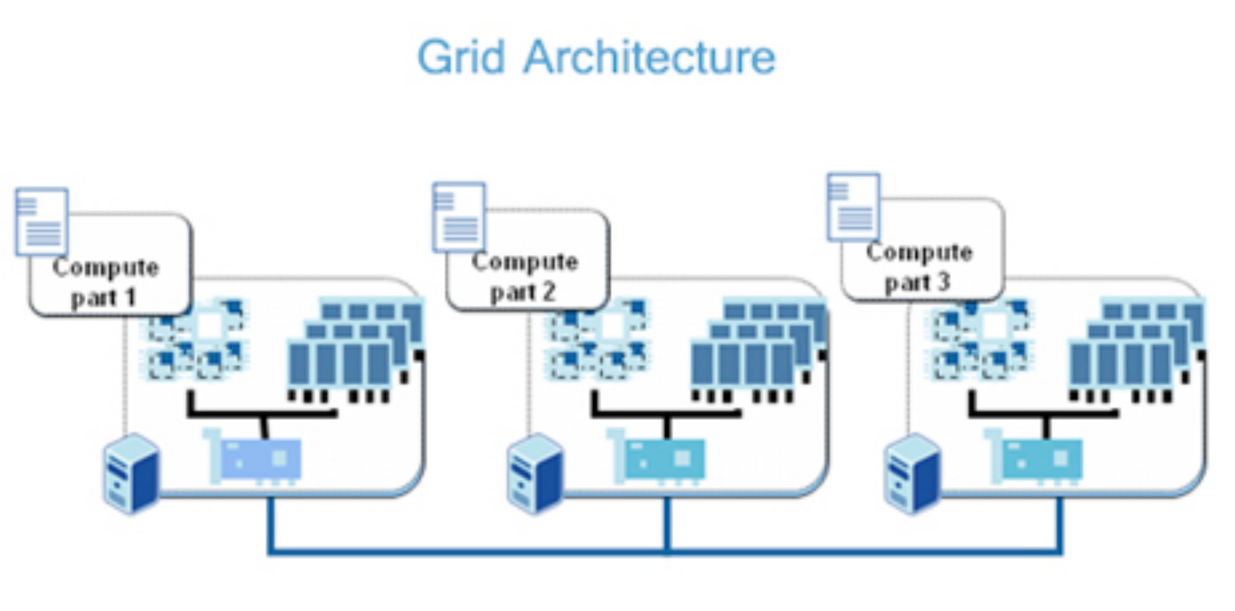
Distributed vs. Parallel Computing

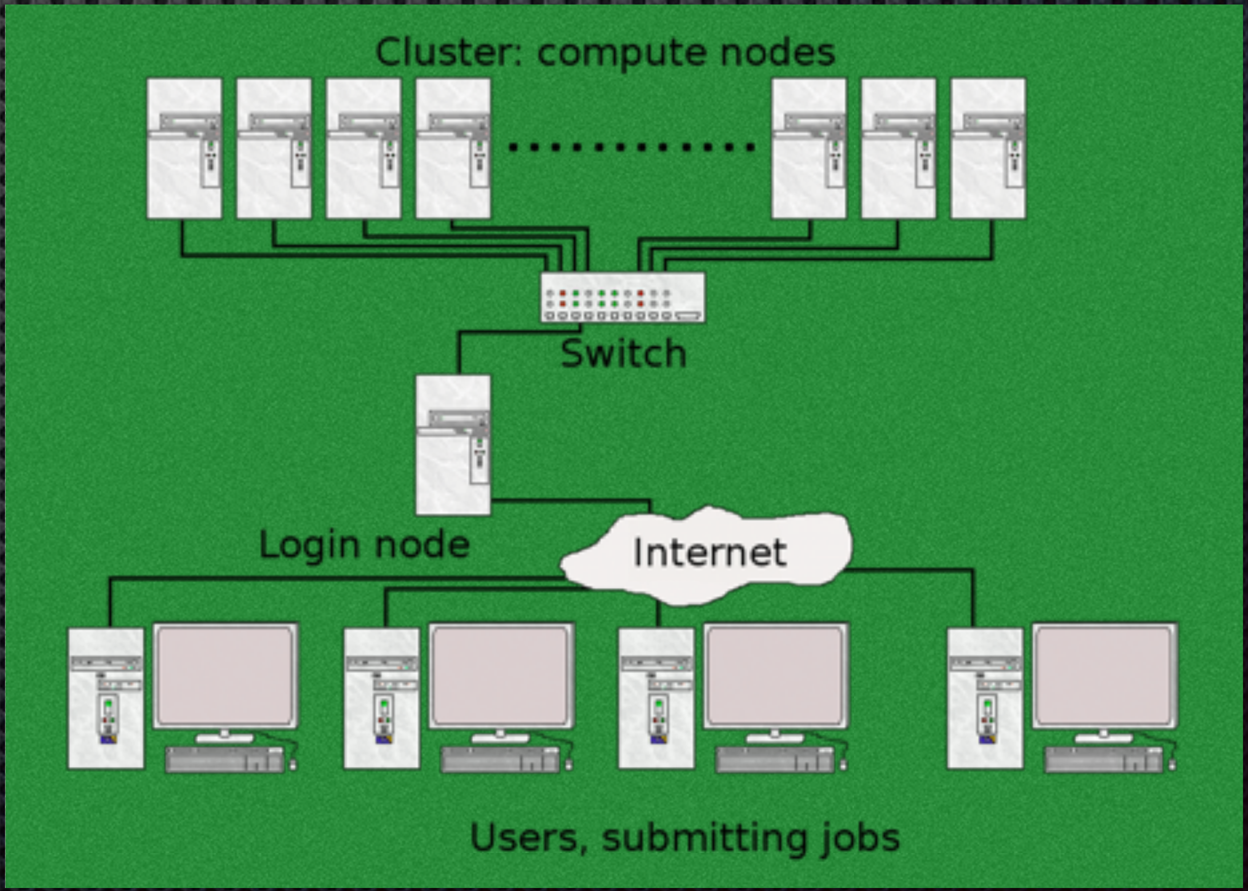
- Parallel Computing: everything in memory
- Distributed Computing: beyond one computer



A picture from wikipedia

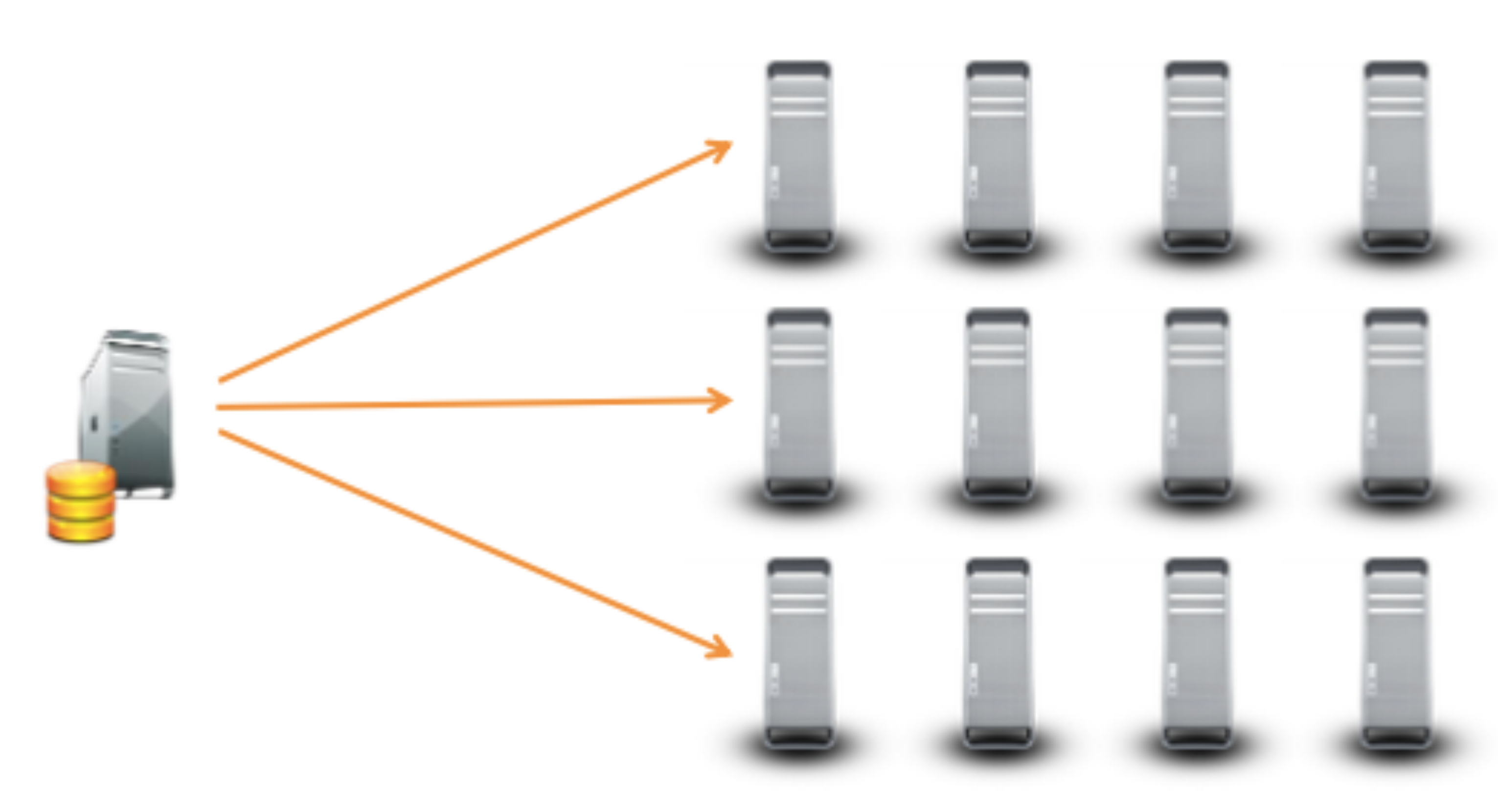
Traditional Distributed Systems





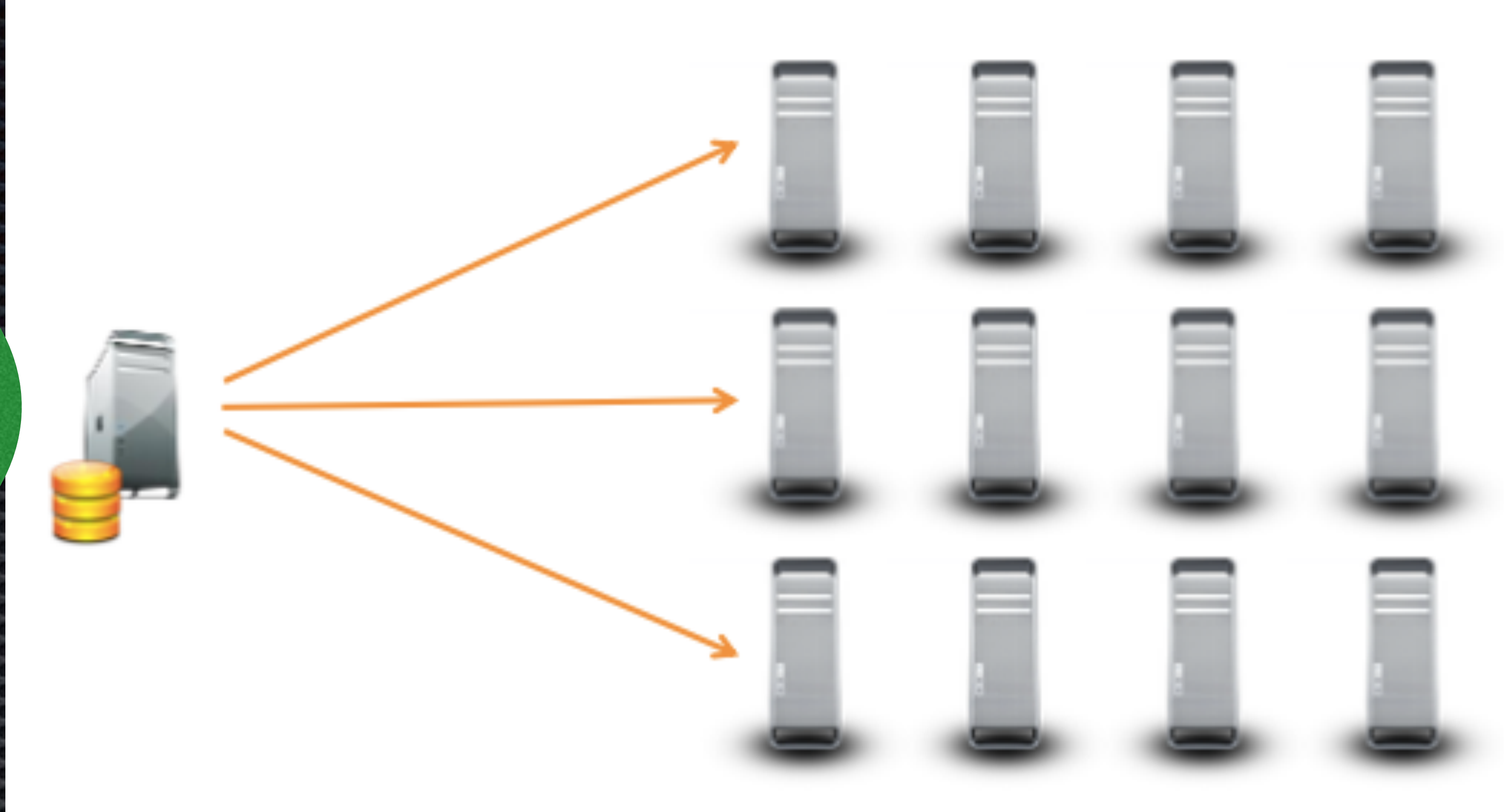
- Grid
- Clusters

Where is the problem?



- Traditional Grid/Cluster good for distributed workload
  - One storage (SAN/NAS), multiple machines
  - small data, long process

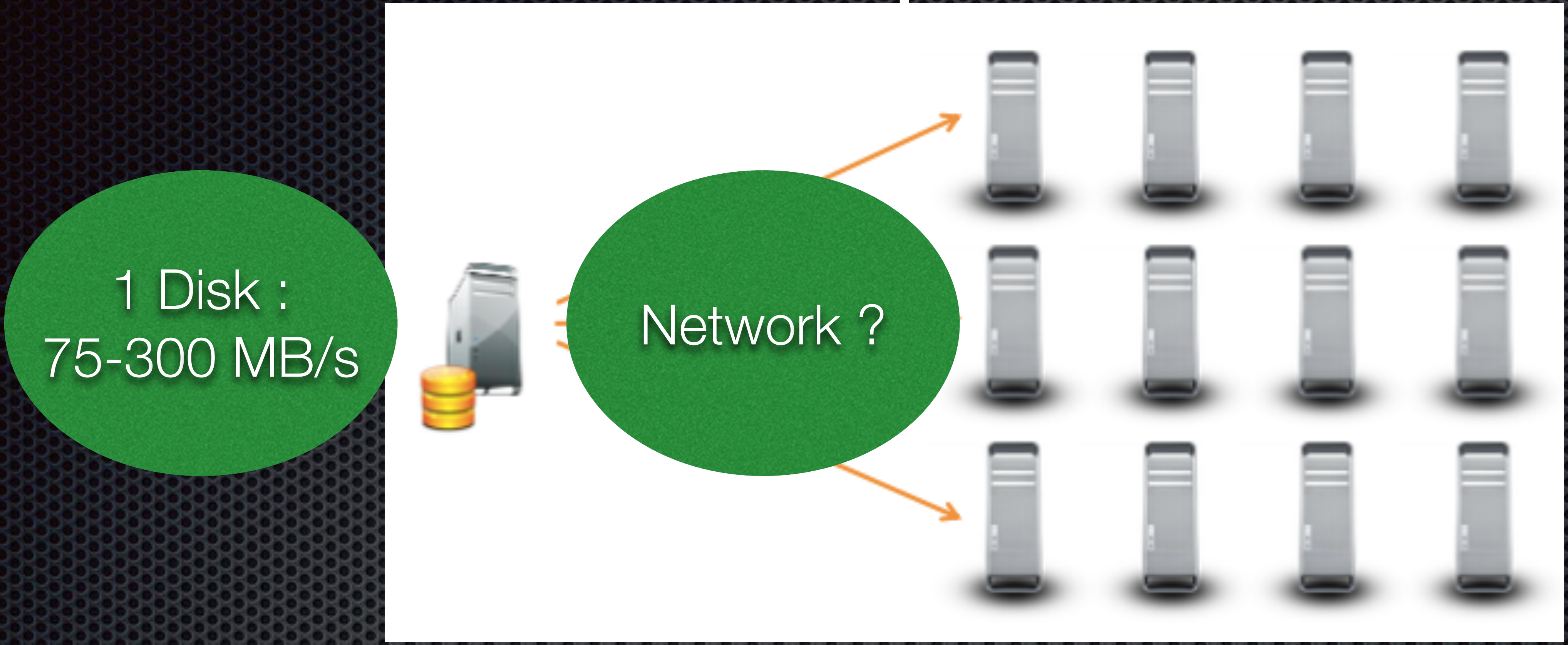
Where is the problem?



1 Disk :
75-300 MB/s

- Traditional Grid/Cluster good for distributed workload
  - One storage (SAN/NAS), multiple machines
  - small data, long process

Where is the problem?



1 Disk :
75-300 MB/s

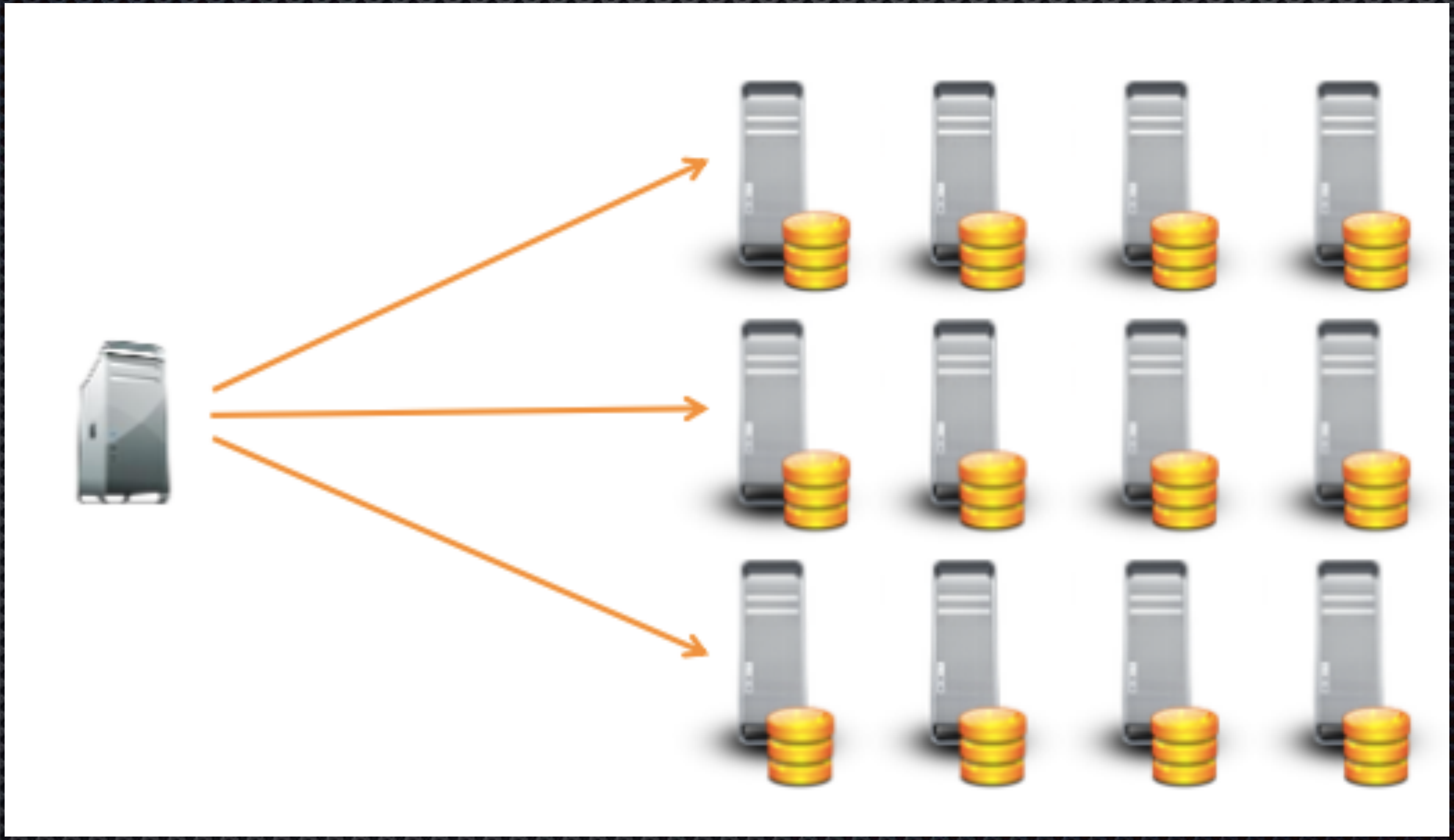
Network ?

- Traditional Grid/Cluster good for distributed workload
  - One storage (SAN/NAS), multiple machines
  - small data, long process

Example

- 200 Gb
  - more than 30 minutes to just read sequentially
  - more than 5 hours to just transfer over fast ethernet

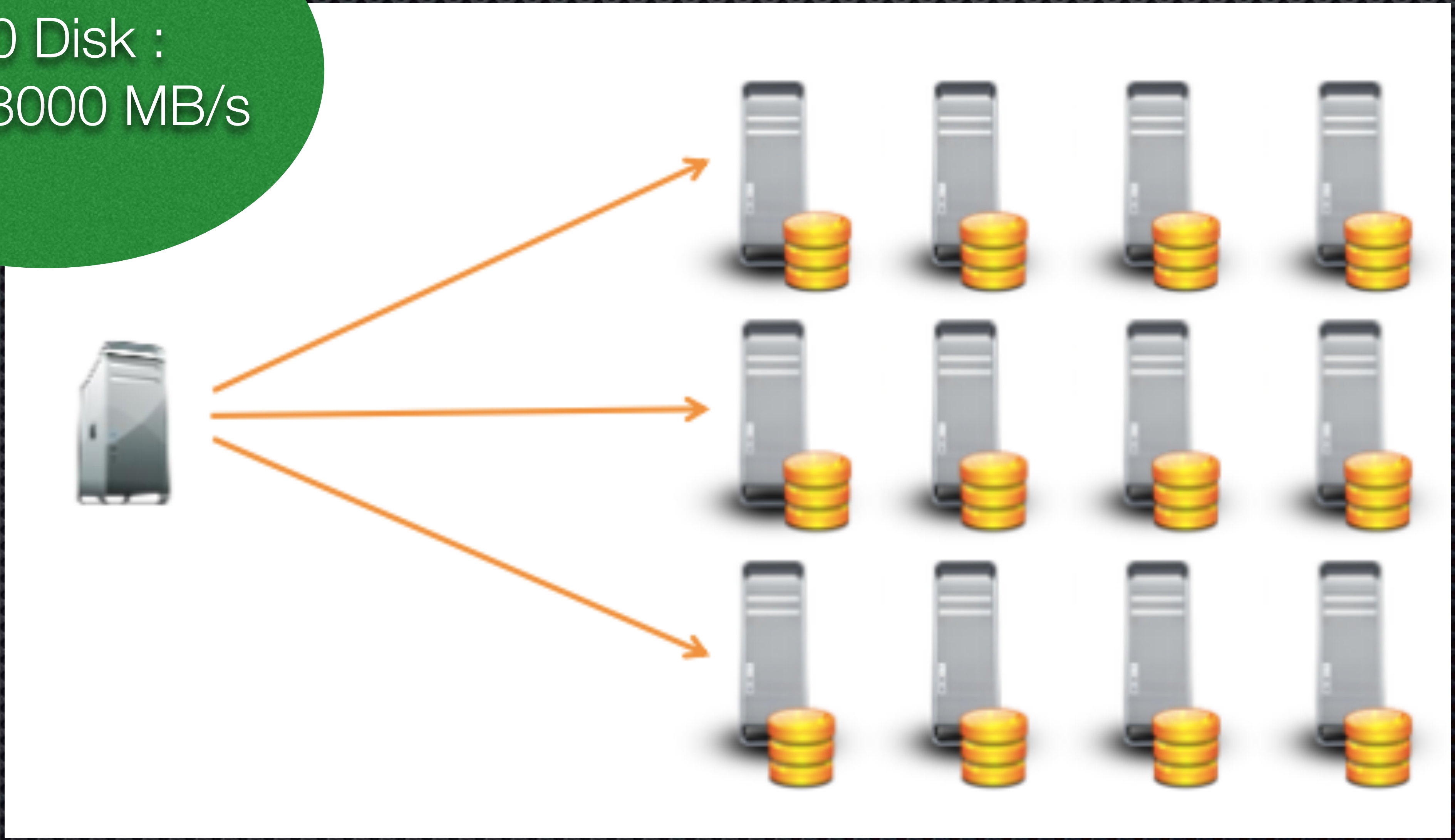
MapReduce?



Let’s distribute the data/storage.

MapReduce?

100 Disk :
7500-3000 MB/s

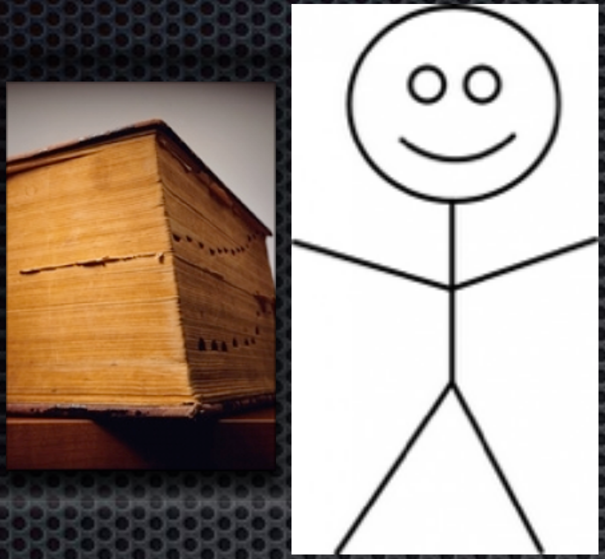


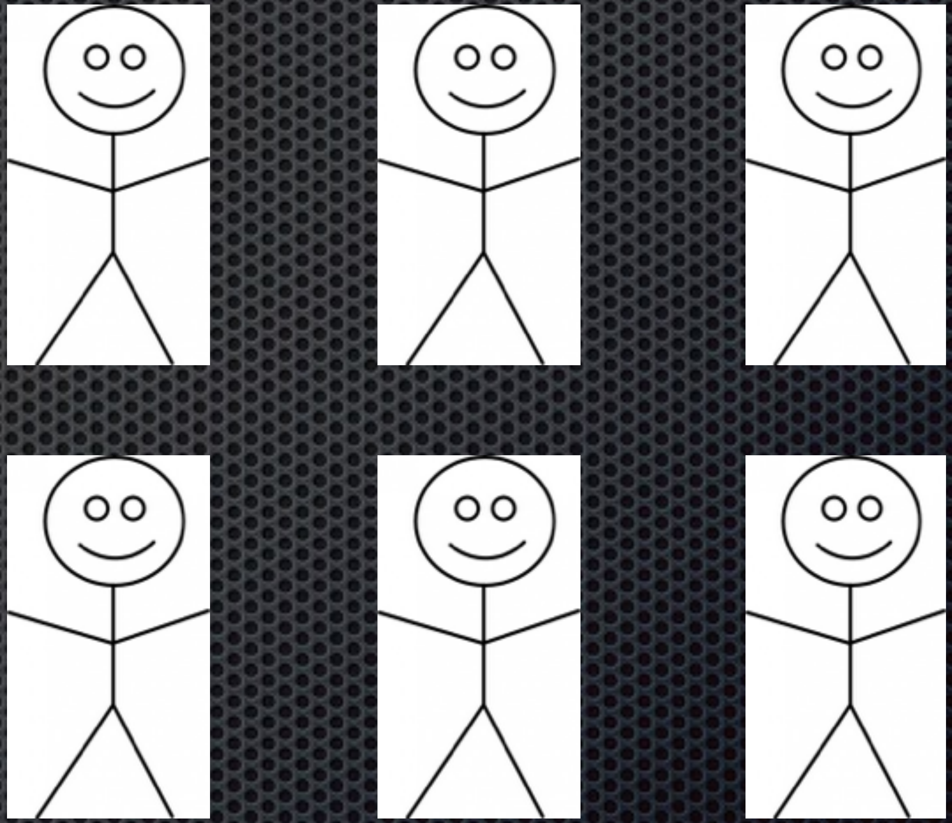
Let’s distribute the data/storage.

Example (Word frequency count)

Find frequency of each word in this big book?

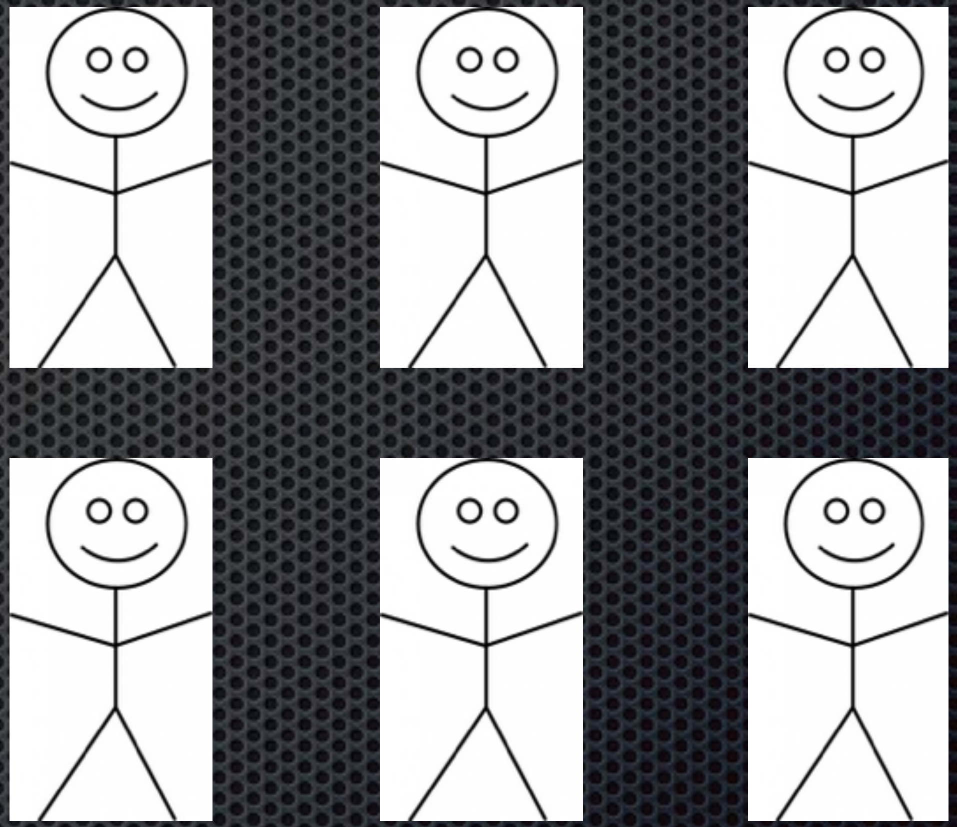
Word freq.: Grid/Cluster





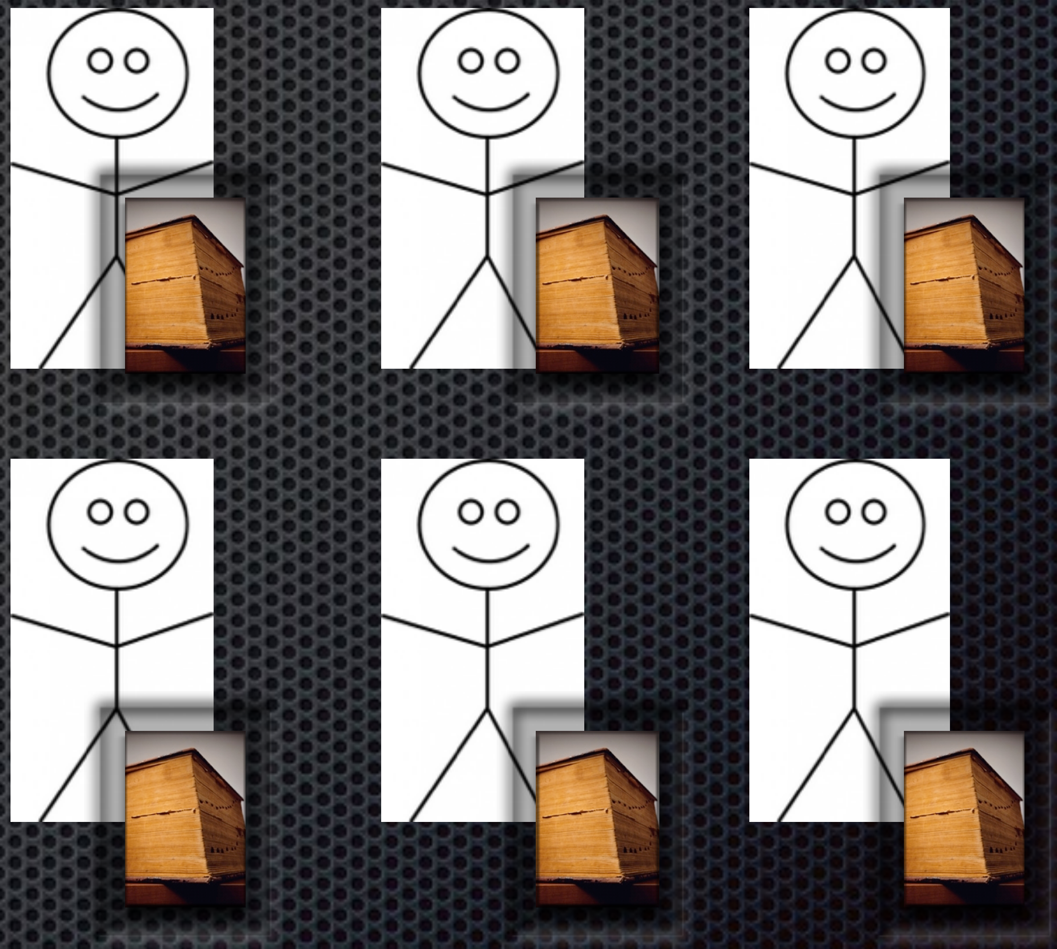
Word freq.: Grid/Cluster

Here is the data.
Please count.



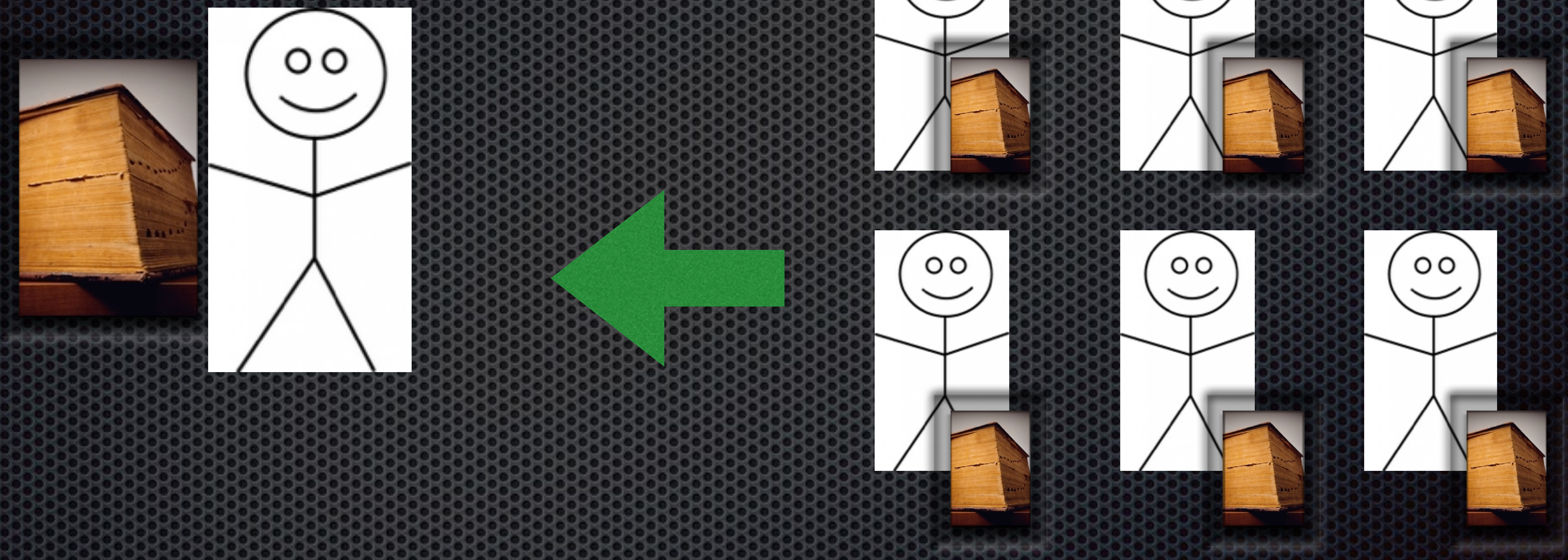
Word freq.: Grid/Cluster

Here is the data.
Please count.

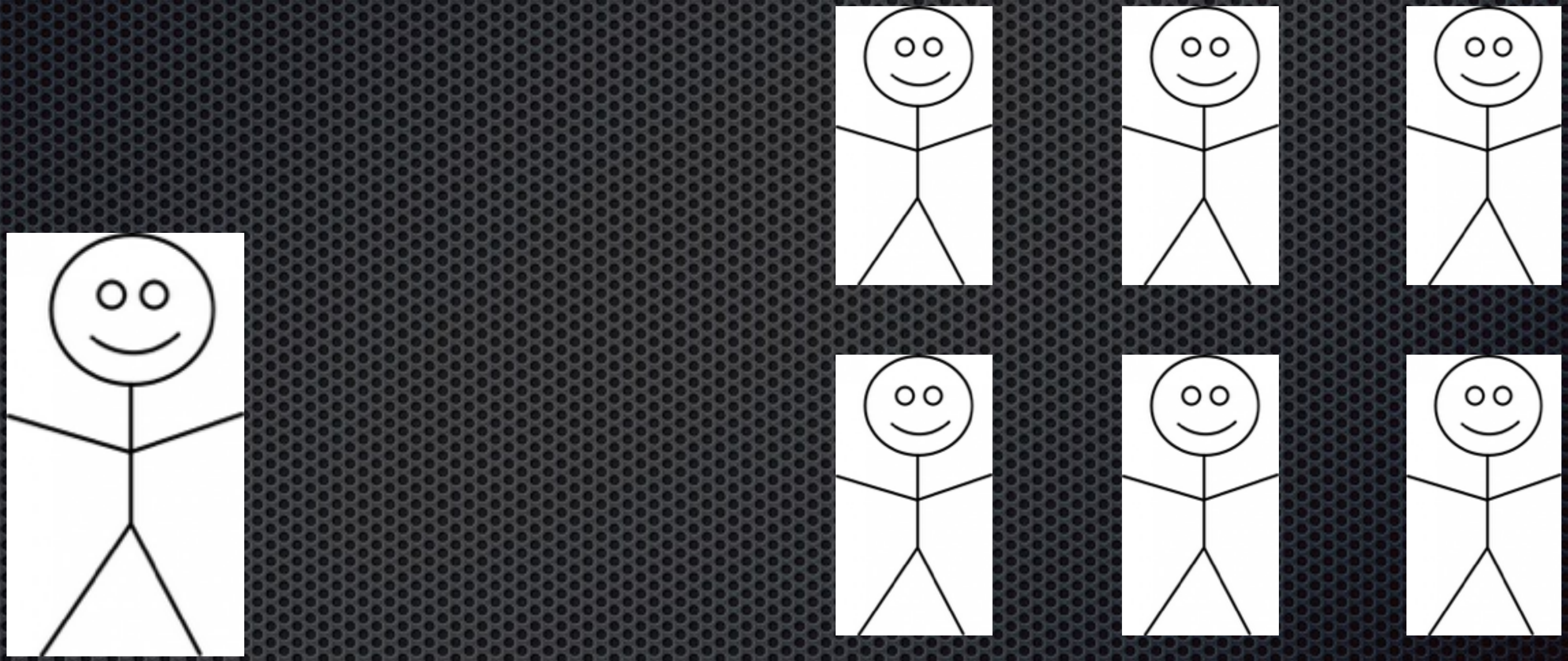


Word freq.: Grid/Cluster

Here is the data.
Please count.

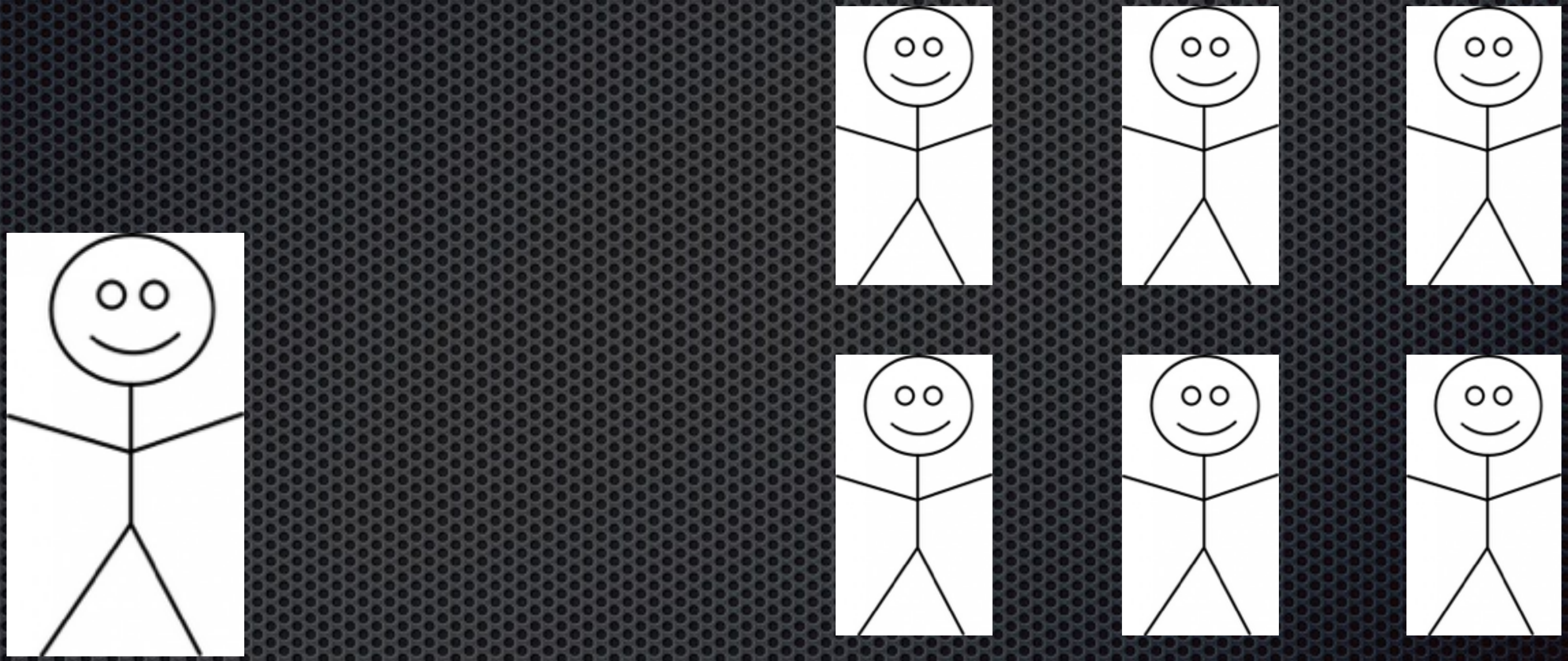


Word freq.: MapReduce



Word freq.: MapReduce

Store a part of data.



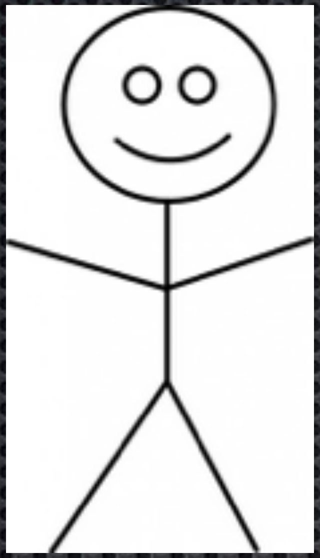
Word freq.: MapReduce

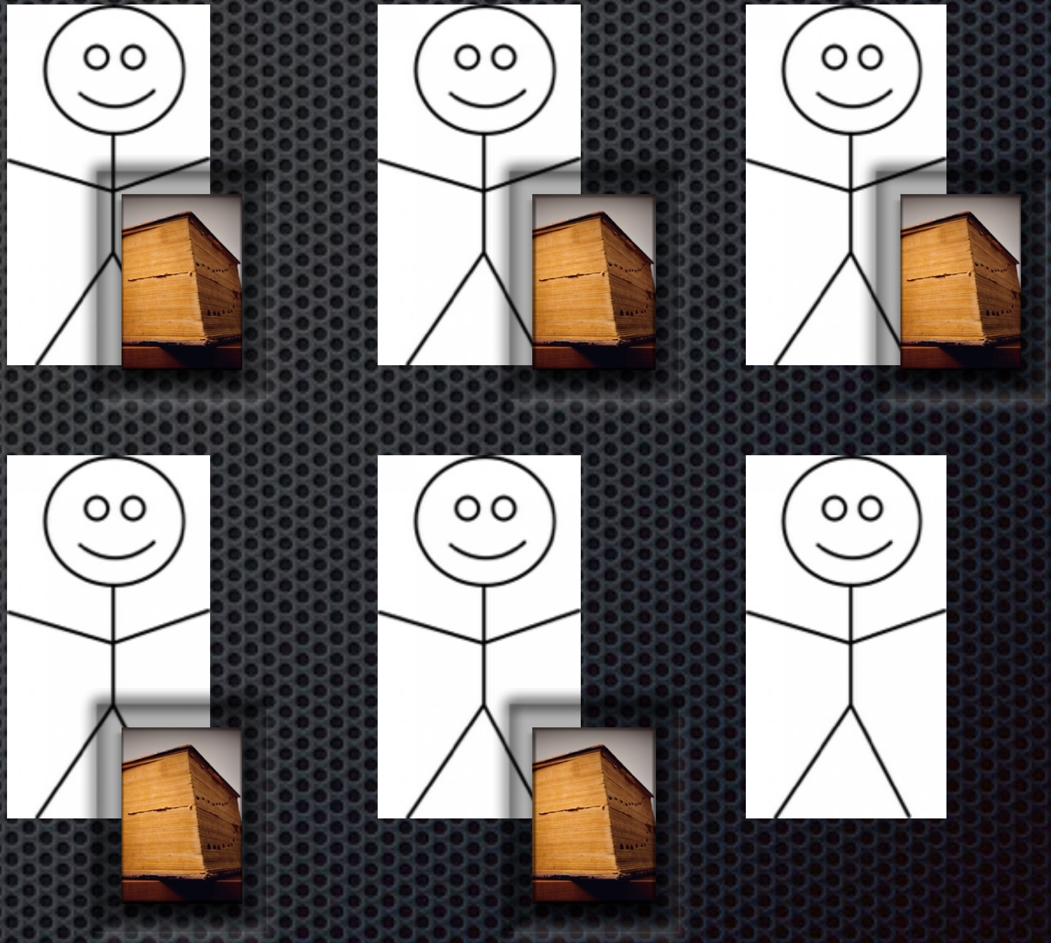
Store a part of data.

Word freq.: MapReduce

Store a part of data.

With your data, please count.

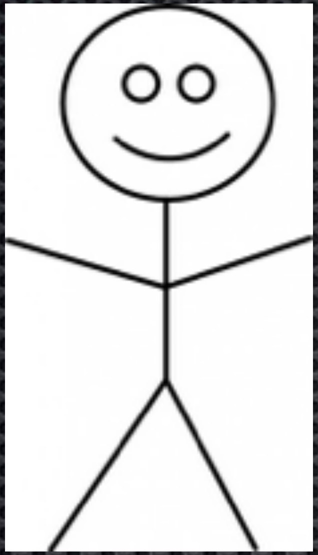




Word freq.: MapReduce

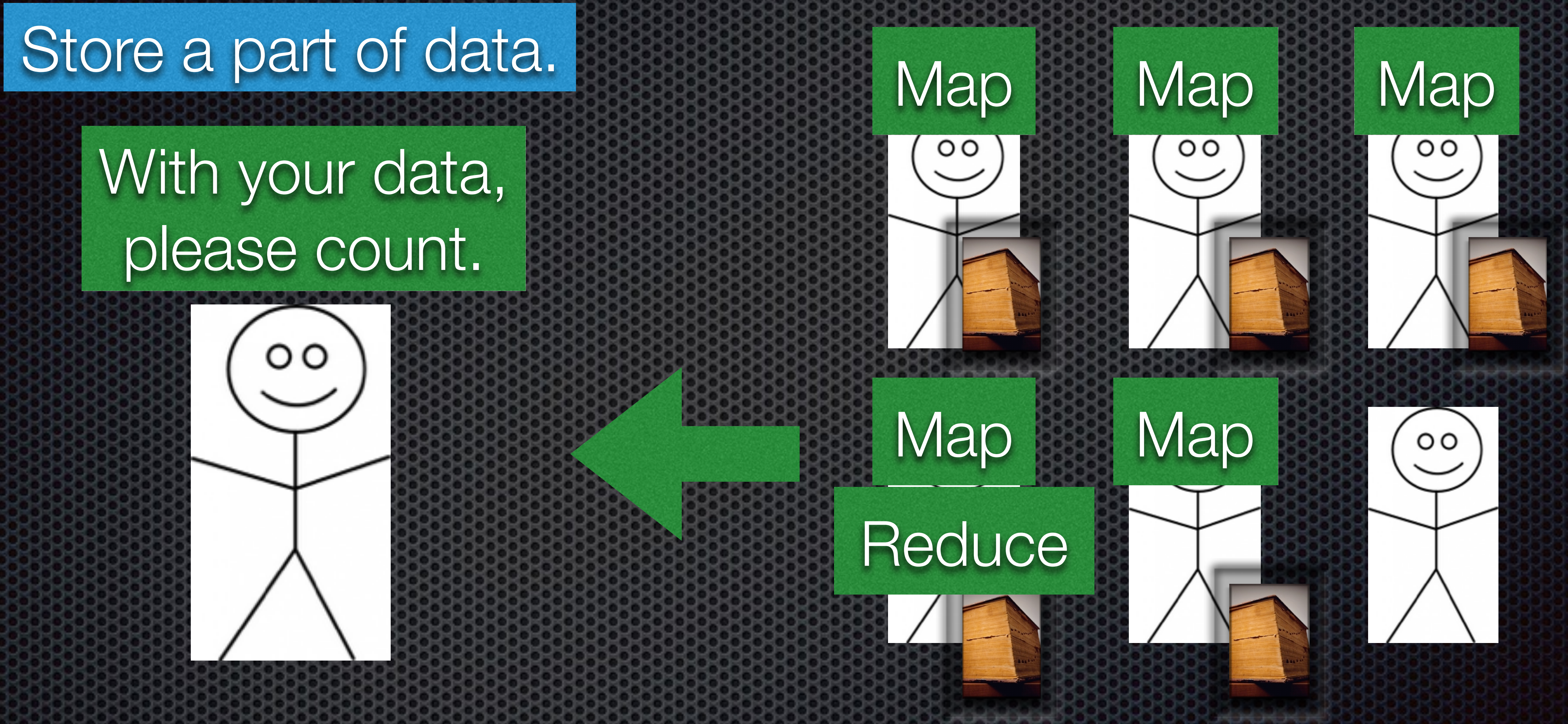
Store a part of data.

With your data, please count.



Map
Map
Map
Map
Map

Word freq.: MapReduce



Store a part of data.

With your data, please count.

Fundamental of MapReduce

- Distributed storage
- Do Map function with local data
  - emit [key, value] pairs
- Pairs with same key feed to same Reduce function
  - emit final value

Fundamental of MapReduce

- Distributed storage
- Do Map function with local data

  Fast
  - emit [key, value] pairs
- Pairs with same key feed to same Reduce function
  - emit final value

Fundamental of MapReduce

- Distributed storage
- Do Map function with local data
  - emit [key, value] pairs
- Pairs with same key feed to same Reduce function
  - emit final value

Reliable?

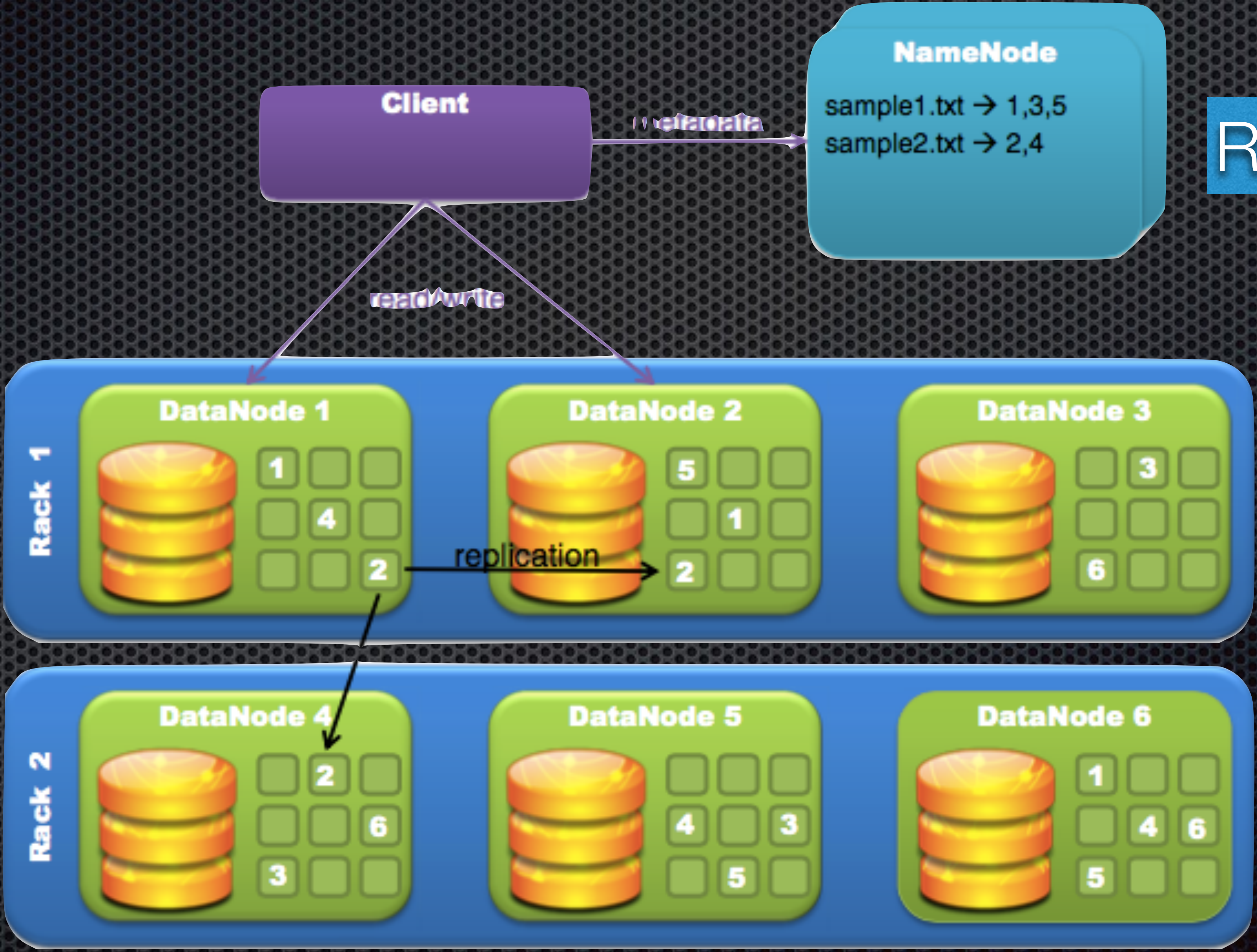
Fast

Disks

- MTBF is 1,200,00 hours
- With 10,000 disks, one will crash every 5 days

Source: google

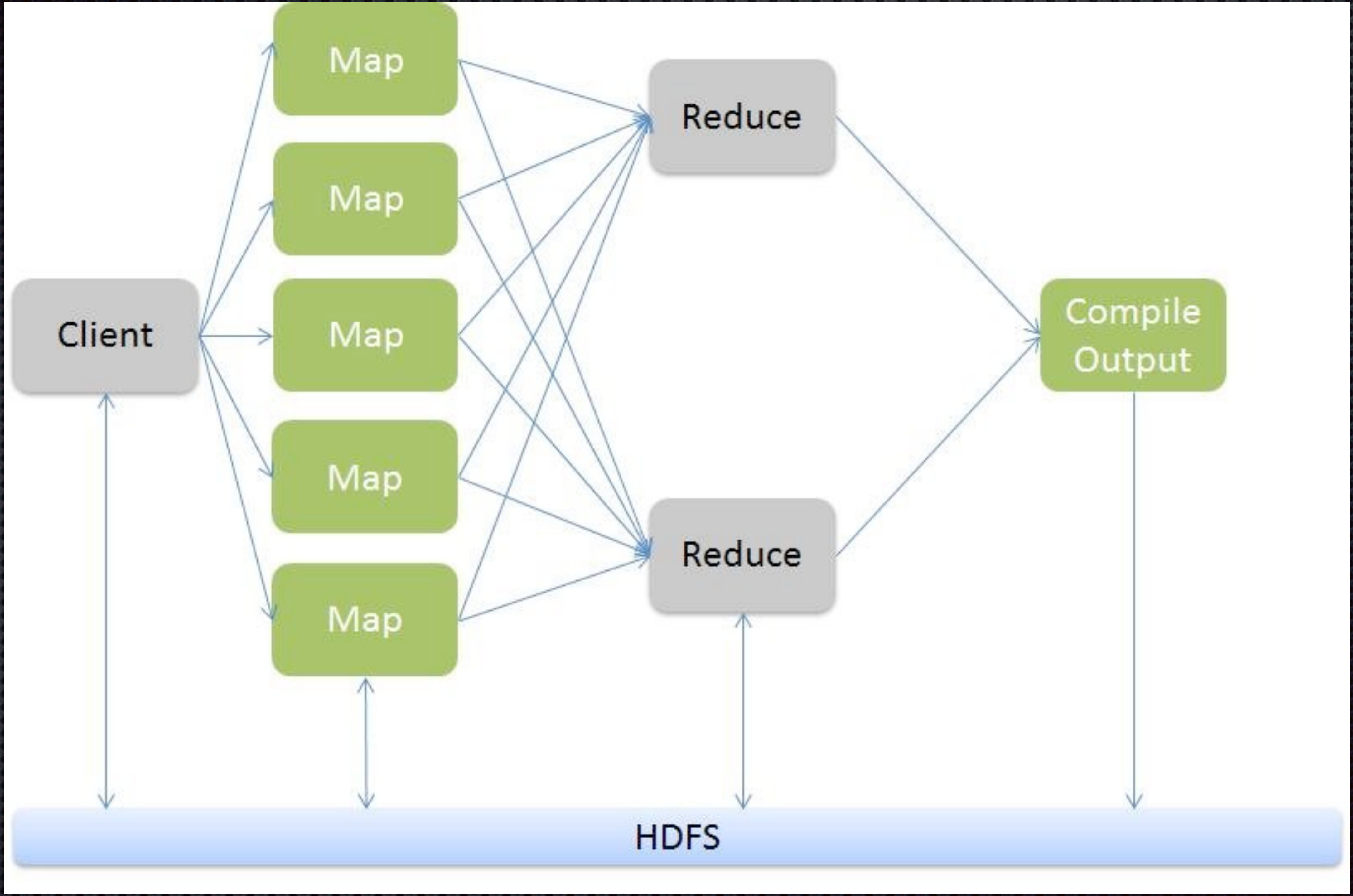
Hadoop HDFS



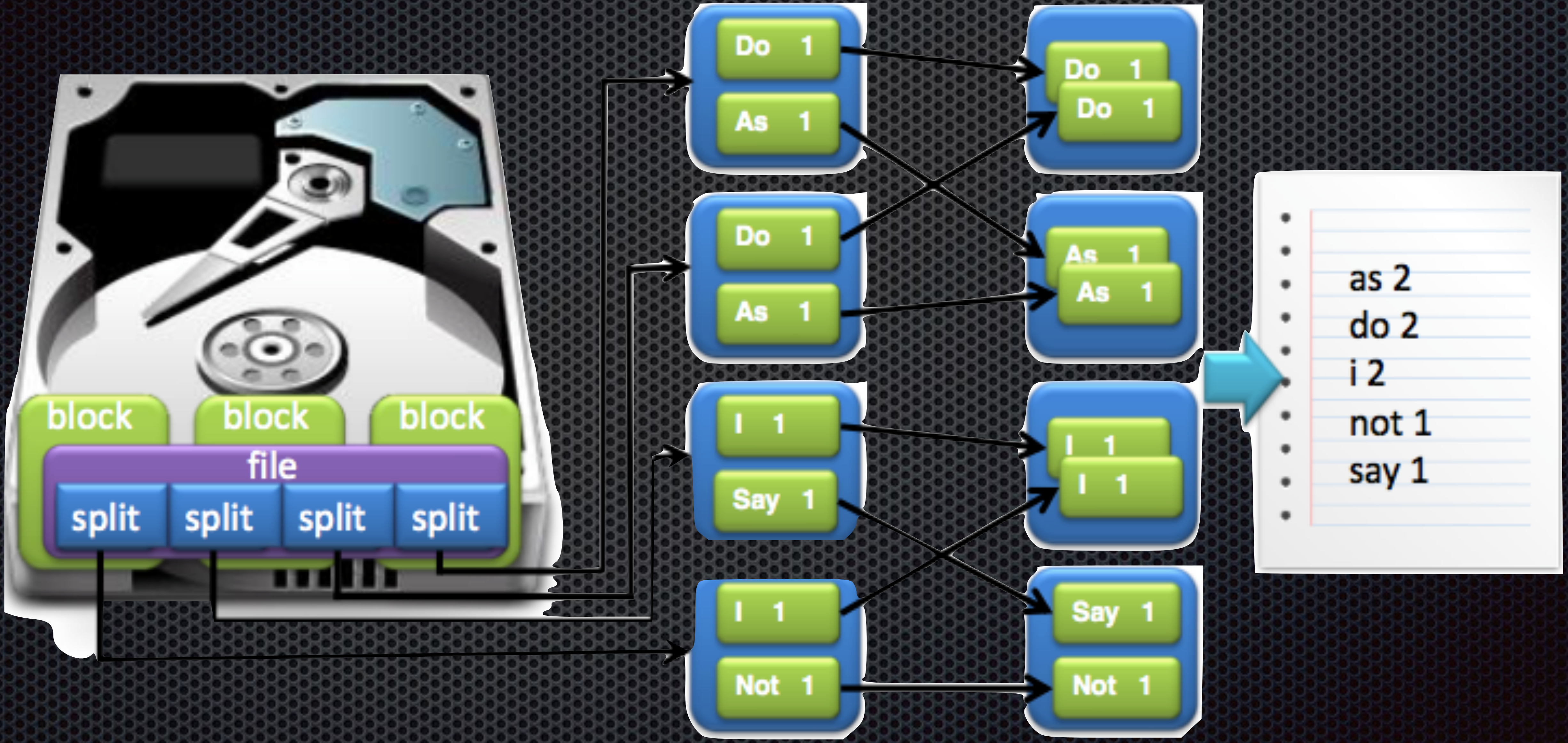
Rackaware

3 copy

How MapReduce work?

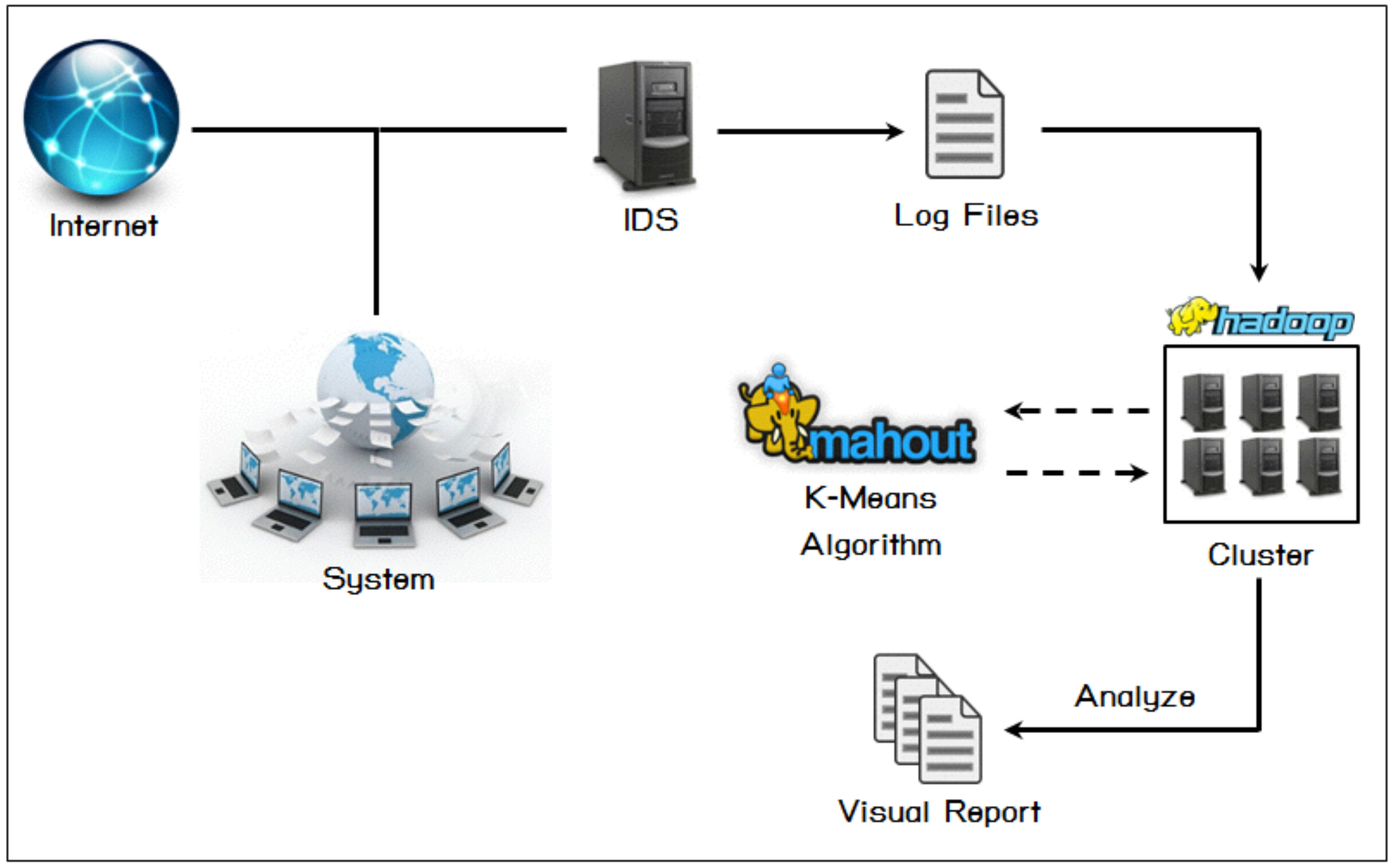


Hadoop Architecture

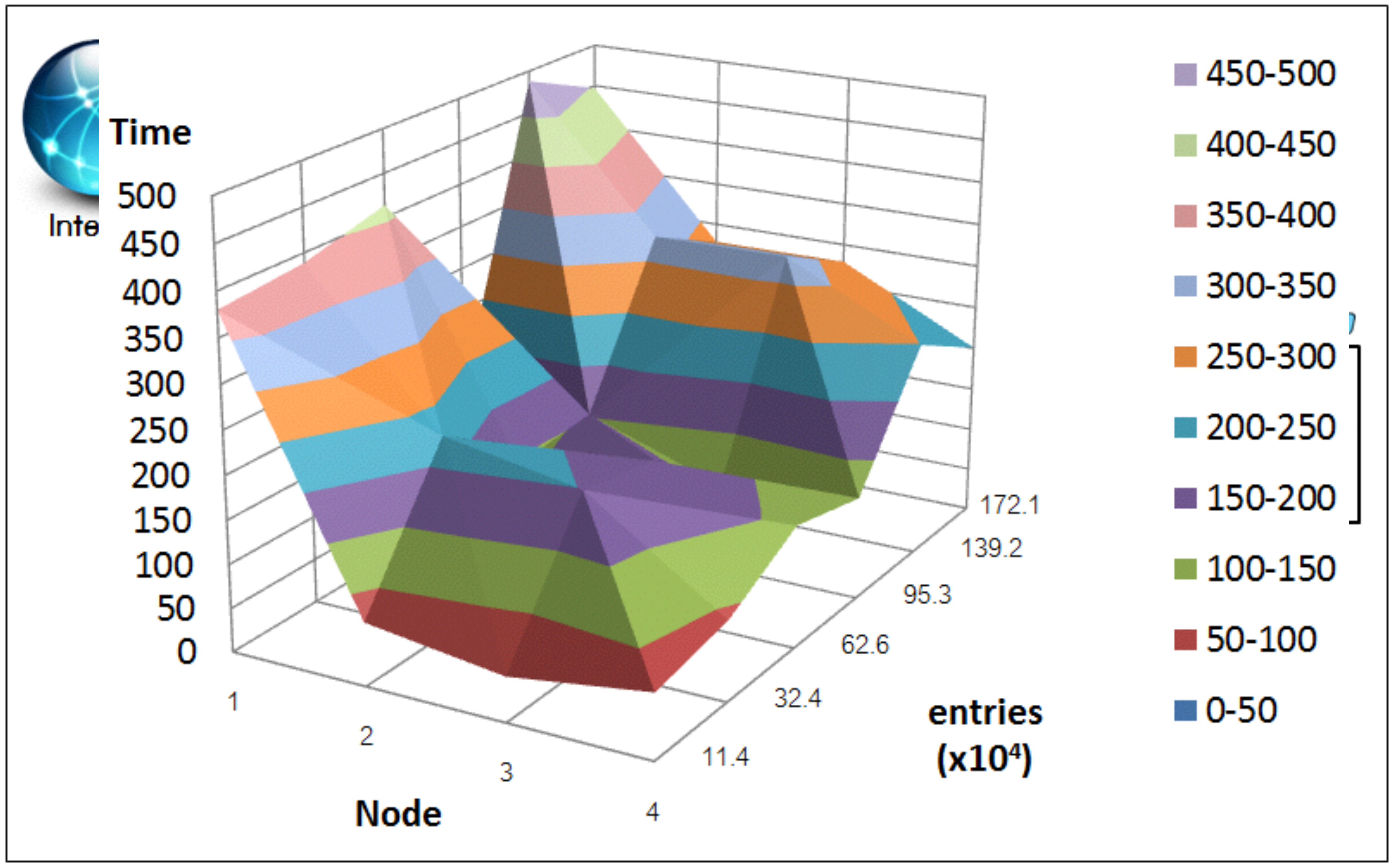


MapReduce in action

Large-Scale log analysis



Large-Scale log analysis



See demo

Sample Data

Personid	Gender	Salary	Name	Grade
A0001	M	1,000	A	A
A0002	F	2,000		B
A0003	M	3,000	C	A

Sample Data

Perso	Gend	Salary	Name	Grade
A000	M	1,000	A	A
A000	F	2,000		B
A000	M	3,000	C	A

Sample Data

Perso	Gend	Salary	Name	Grade
A000	M	1,000	A	A
A000	F	2,000		B
A000	M	3,000	C	A

```
function Map(doc) {
    emit(Grade,1);
    }
```

Sample Data

Perso	Gend	Salary	Name	Grade
A000	M	1,000	A	A
A000	F	2,000		B
A000	M	3,000	C	A

```
function Map(doc) {
    emit(Grade,1);
    }
```

```
function Reduce(key, values)
{
    return sum(values);
}
```

Sample Data

Perso	Gend	Salary	Name	Grade
A000	M	1,000	A	A
A000	F	2,000		B
A000	M	3,000	C	A

Grade	Count
A	2
B	1

```
function Map(doc) {
    emit(Grade,1);
    }
```

```
function Reduce(key, values)
{
    return sum(values);
}
```

"MapReduce is just A Major Step Backwards!!!"

– Dewitt and StoneBraker in January 17, 2008

Let’s debate!

Major Step Backwards!!!

- No schema, type (Garbage)
- No standard access language (e.g. SQL)
- No index, but brute force
- No transaction
- No integrity (e.g. foreign key)

Let’s fight back!

MapReduce is not DBMS

- No schema, type (Garbage)
- No standard access language (e.g. SQL)
- No index, but brute force
- No transaction
- No integrity (e.g. foreign key)

MapReduce is not DBMS

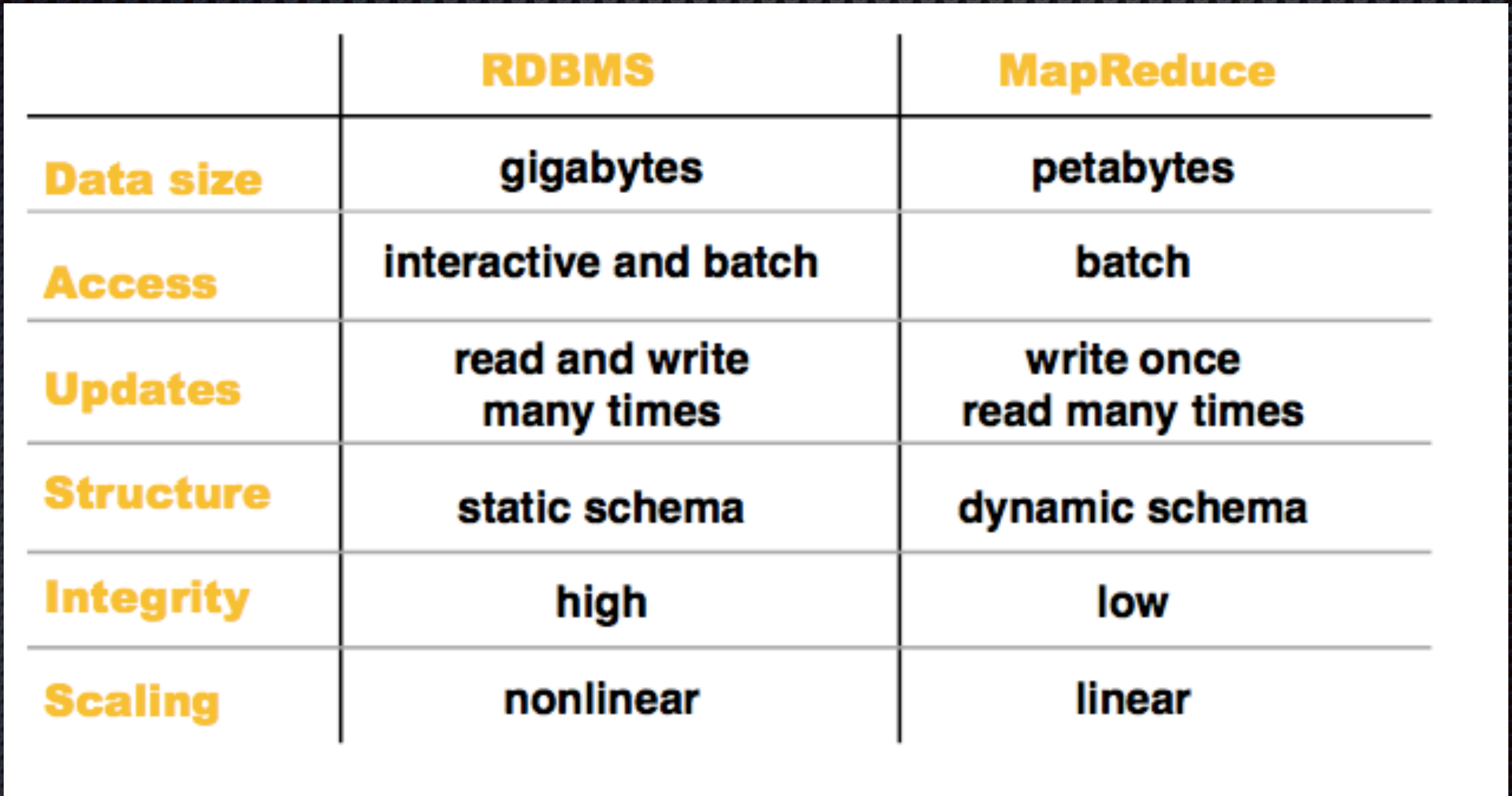
MapReduce is a big forward in.

- Scalability (Scale out)
- Reliable software model for unreliable hardware

What have we learned?

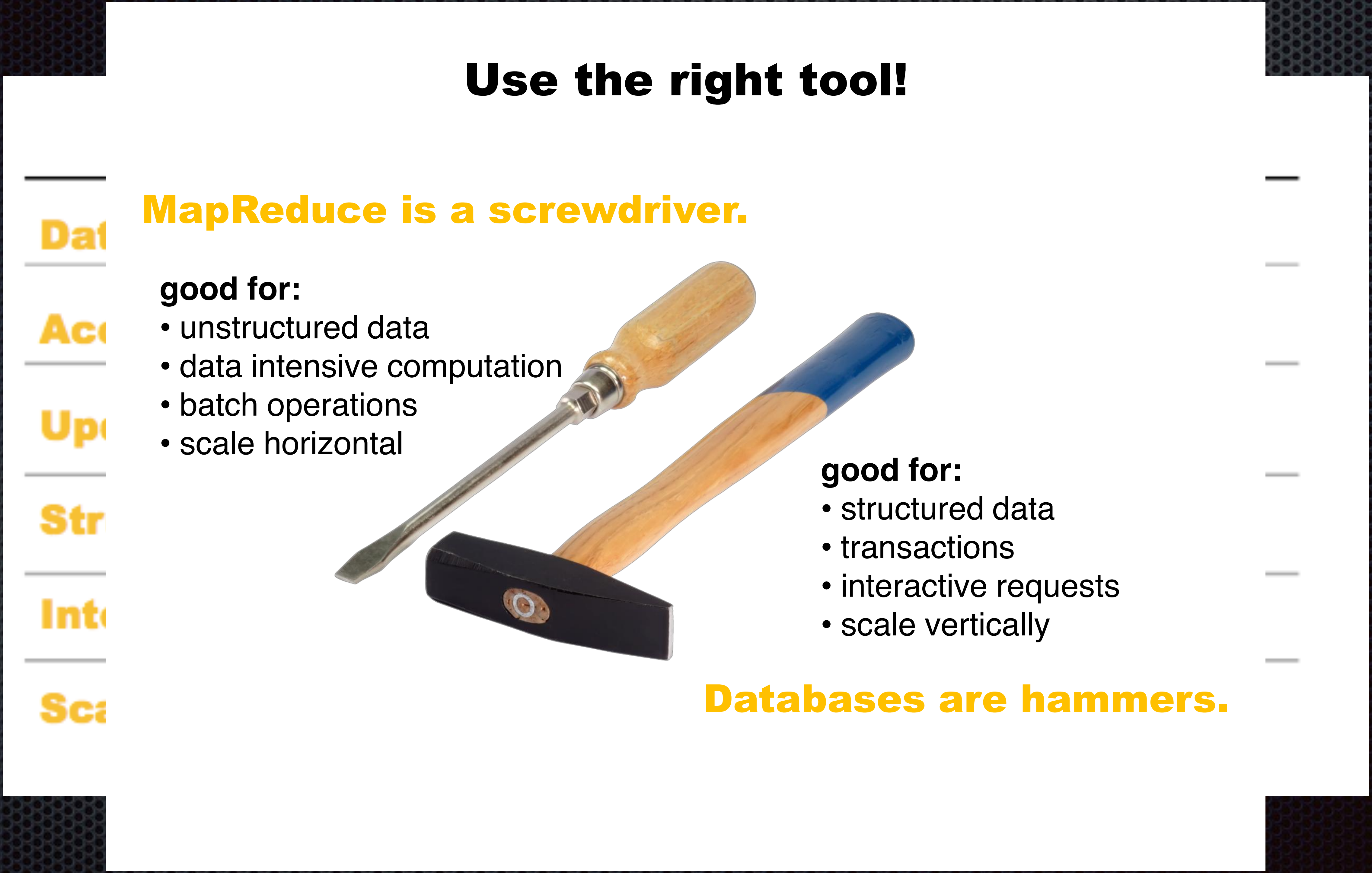
- MapReduce is a software solution for:
  - processing software on unreliable hardware
  - distributing I/O (data as well as workload)
- MapReduce is not DBMS.
- Think in Map function and Reduce function

RDBMS vs. MapReduce



	RDBMS	MapReduce
Data size	gigabytes	petabytes
Access	interactive and batch	batch
Updates	read and write many times	write once read many times
Structure	static schema	dynamic schema
Integrity	high	low
Scaling	nonlinear	linear

RDBMS vs. MapReduce



Dat

Ac

Up

Str

Int

Sca

Use the right tool!

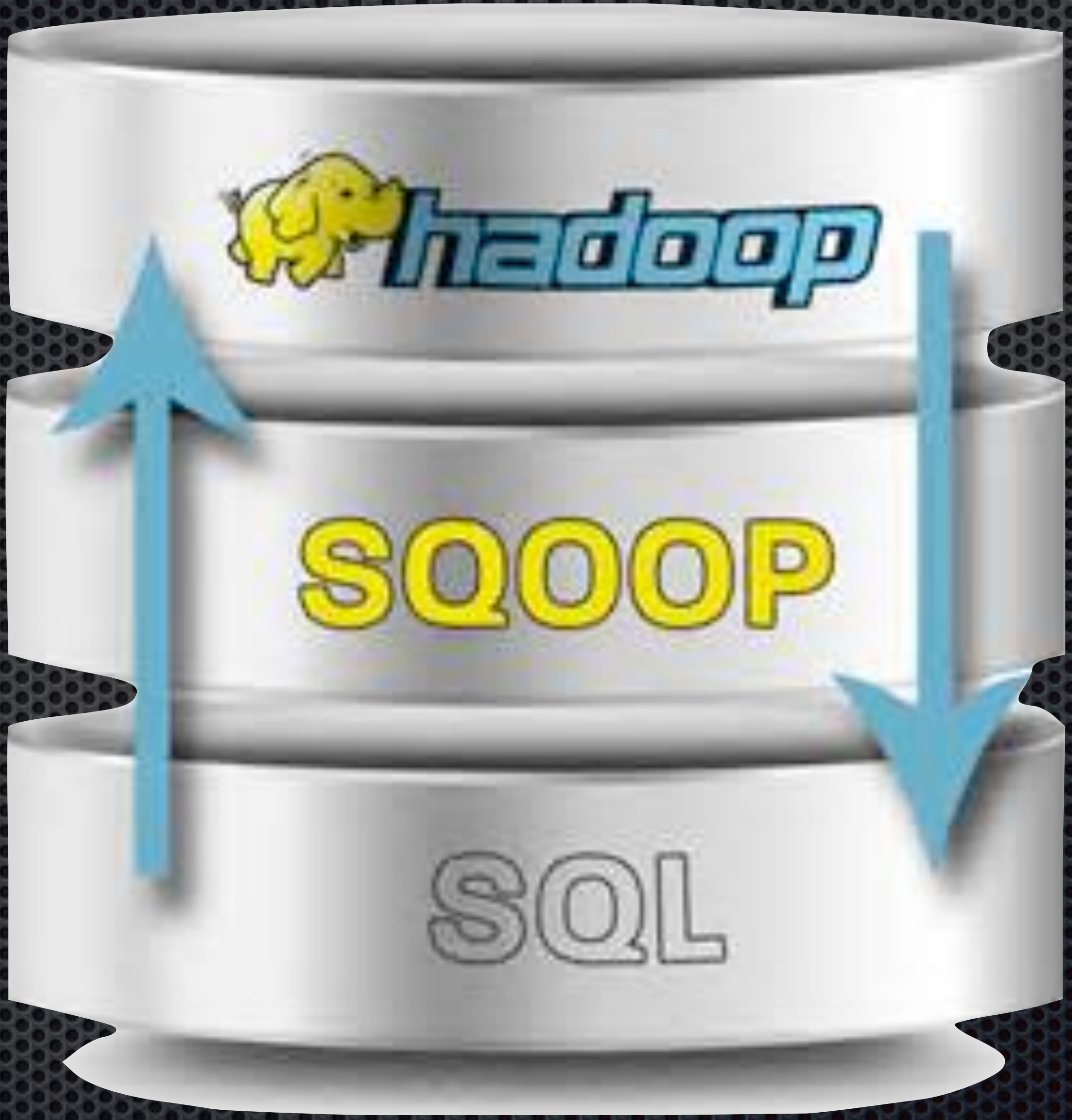
MapReduce is a screwdriver.

good for:
- unstructured data
- data intensive computation
- batch operations
- scale horizontal

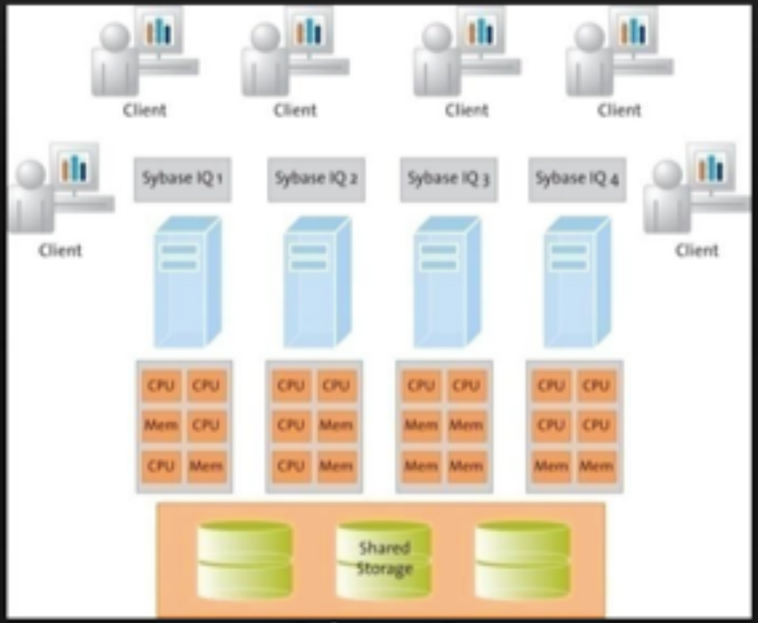
good for:
- structured data
- transactions
- interactive requests
- scale vertically

Databases are hammers.

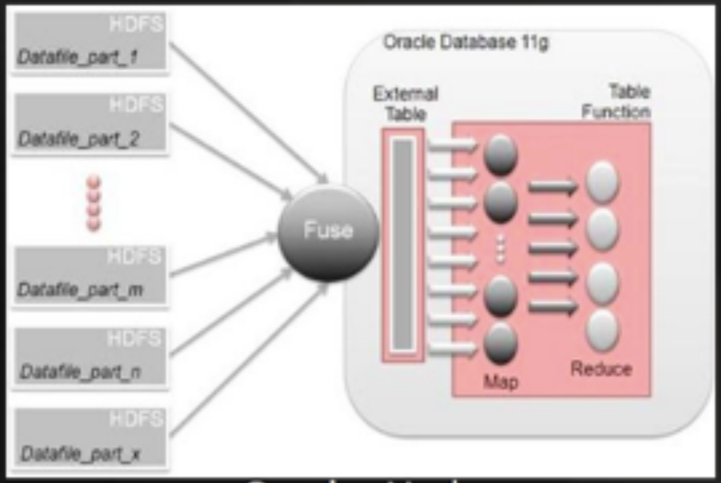
Why don't use both?



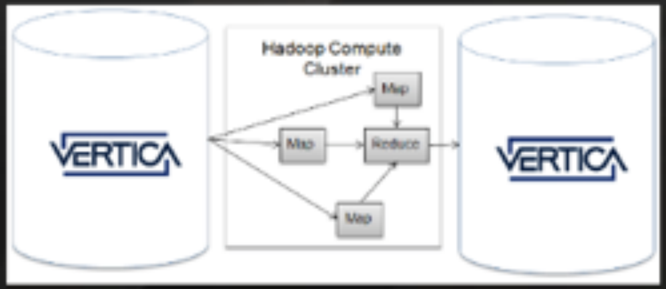
Why don't use both?



Sybase IQ



Oracle+Hadoop

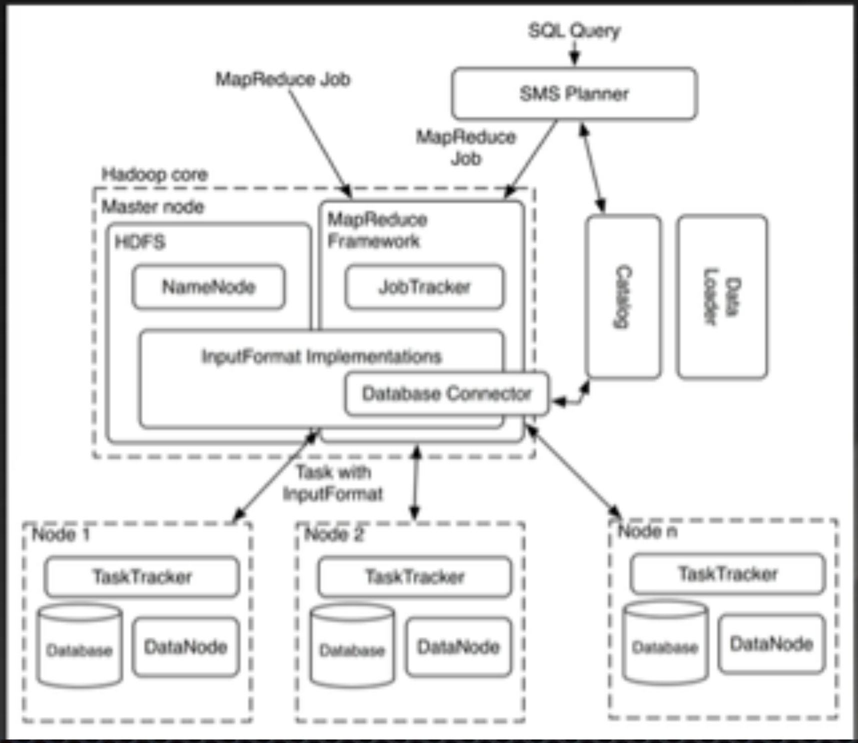


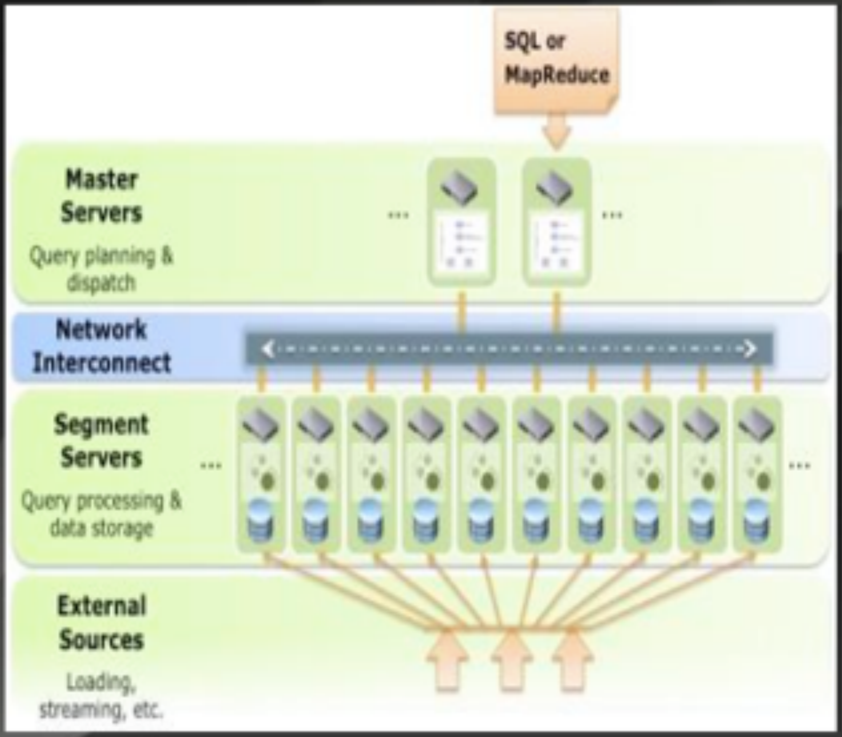
Vertica+Hadoop

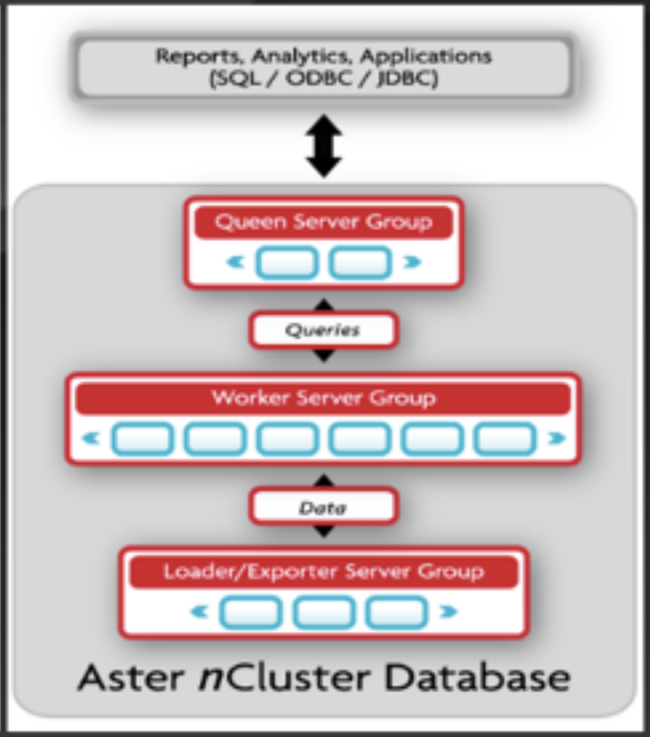
Netezza+MapReduce

Hadoop nodes
AMP

Teradata+MapReduce






Reports, Analytics, Applications (SQL / ODBC / JDBC)
Queen Server Group
Queries
Worker Server Group
Data
Loader/Exporter Server Group
Aster nCluster Database

Hadoop is good for

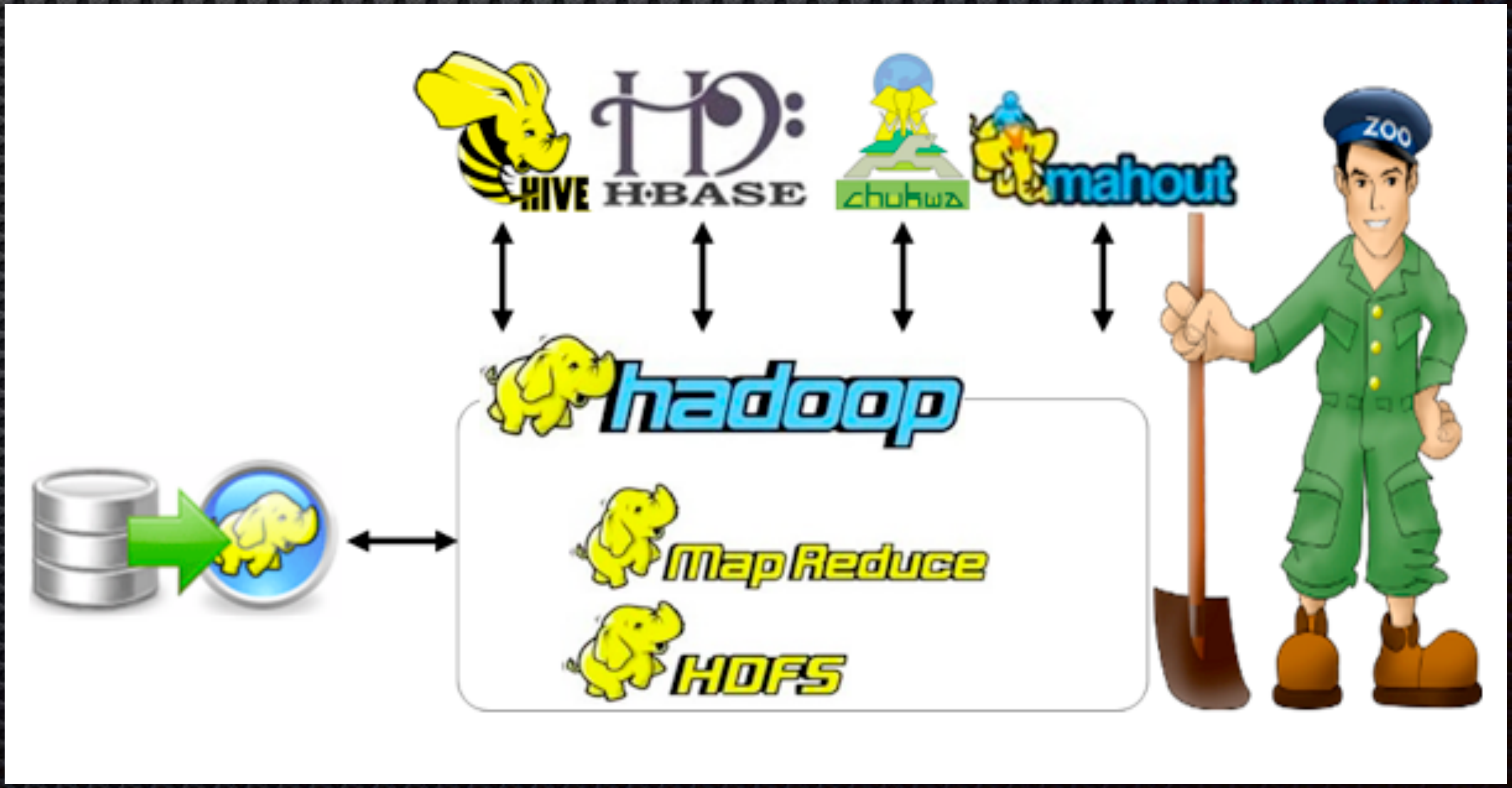
- Large-scale data analysis
- Search engines
- etc…

Hadoop is bad for

- Pi estimation
- Dependency calculation (Recurrent relation)
- DBMS replacement / transaction

Easy ways to MapReduce

- Use high-level analysis tools
  - Hive (SQL style)
  - Gnu R
  - Spark
  - H2O



Thank you
Question?