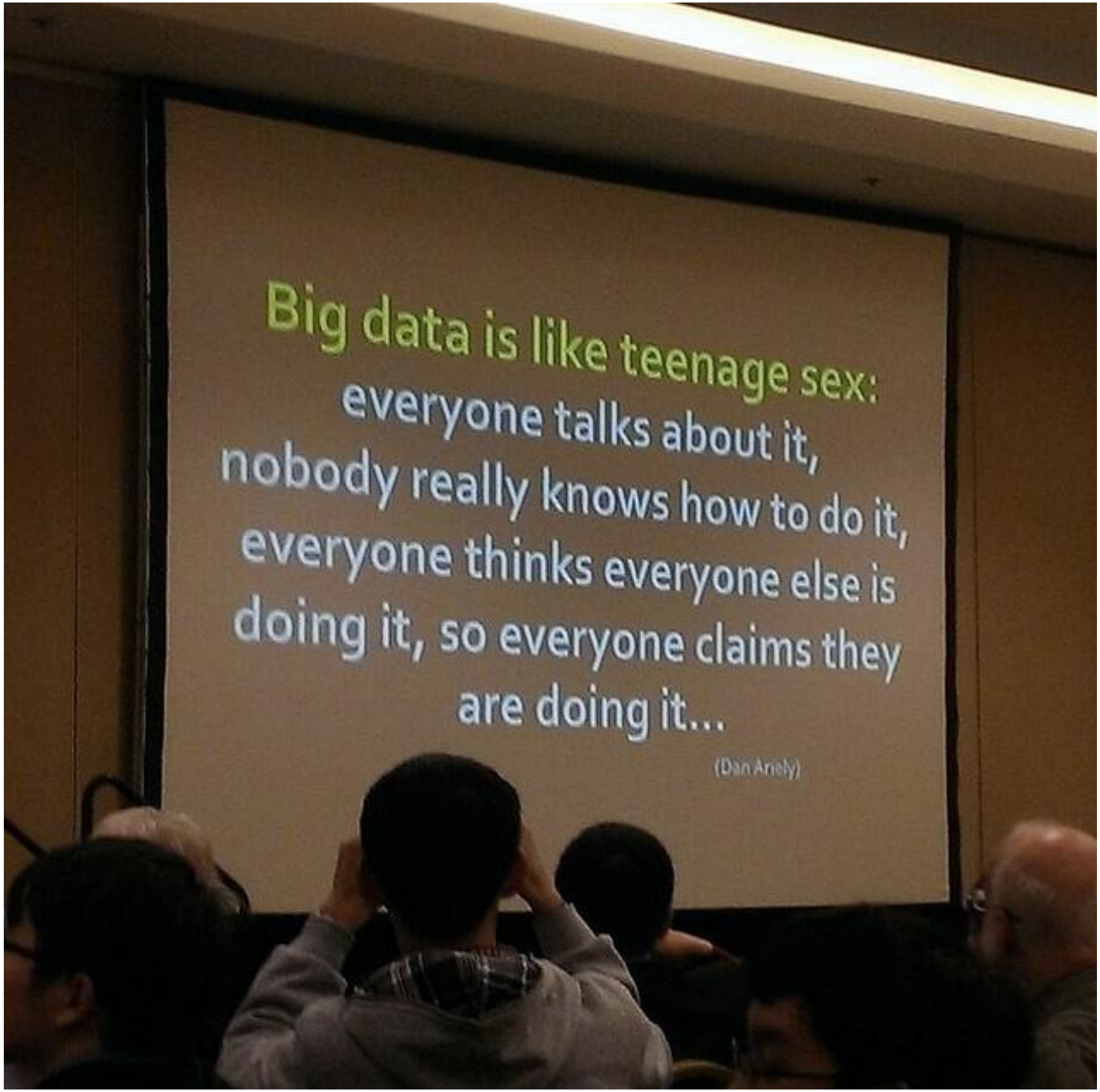
Introduction to Big Data

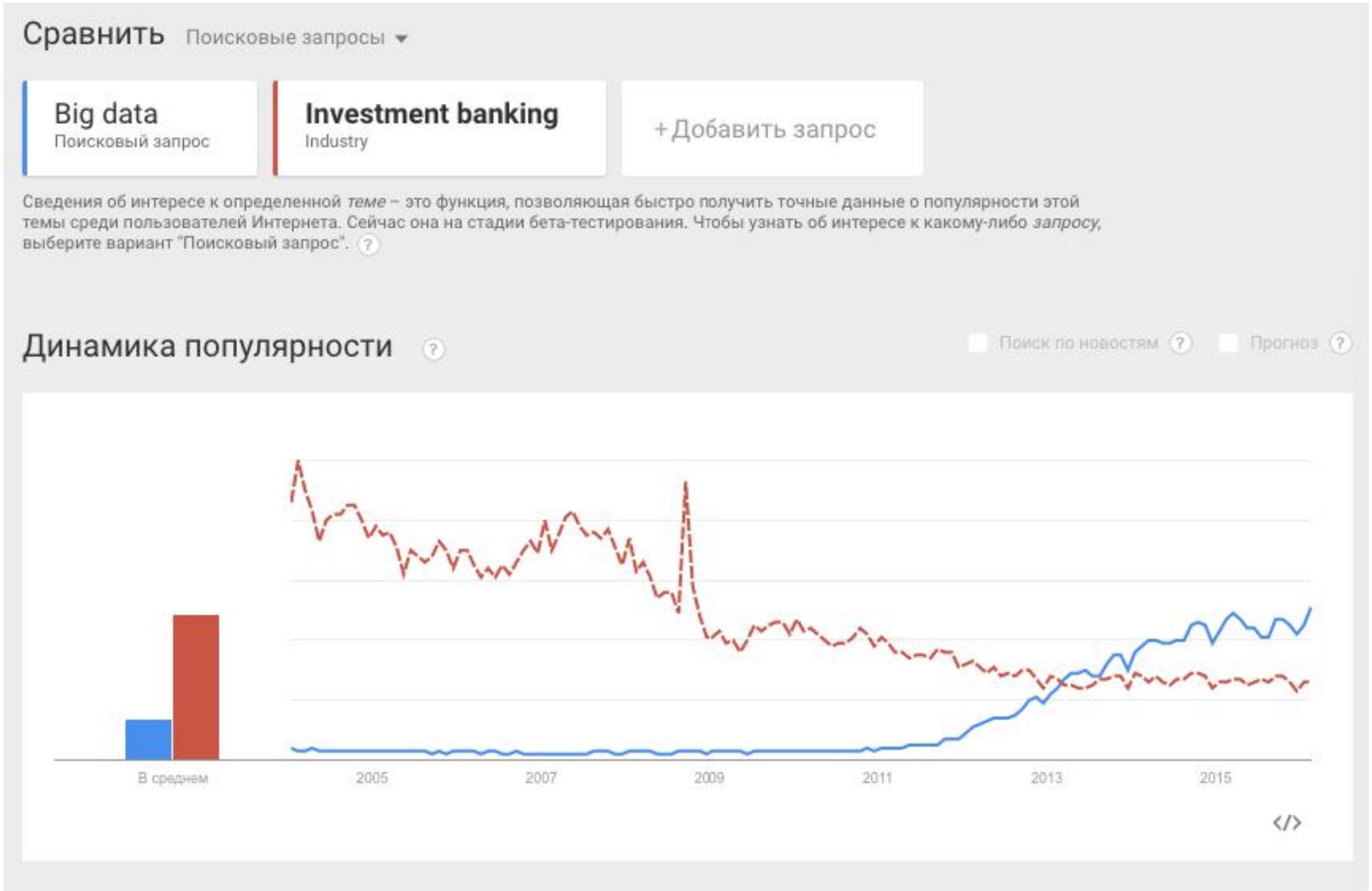
Ilya Ezepov


Big data is like teenage sex:
everyone talks about it,
nobody really knows how to do it,
everyone thinks everyone else is
doing it, so everyone claims they
are doing it...
(Dan Ariely)

Agenda

- Limitation of classical data analysis
- Distributed filesystems
- Data Centres
- MapReduce computational model
- Few words about Big Data world
- Study plan

Popularity



trends.google.com


amazon.com®

item-item collaborative filtering patent:

100x increase in sales

Classical data analysis

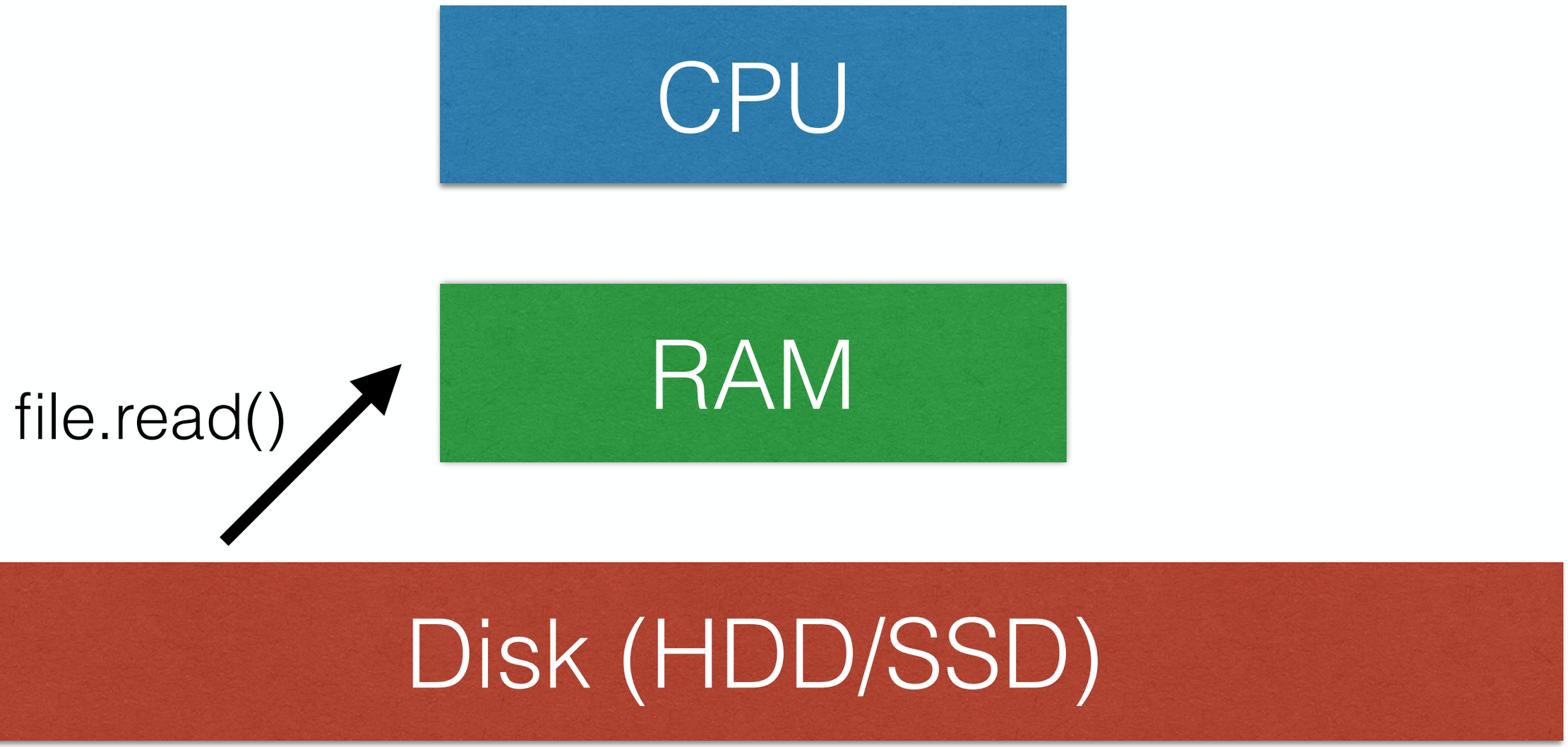
Super simplified computer

CPU

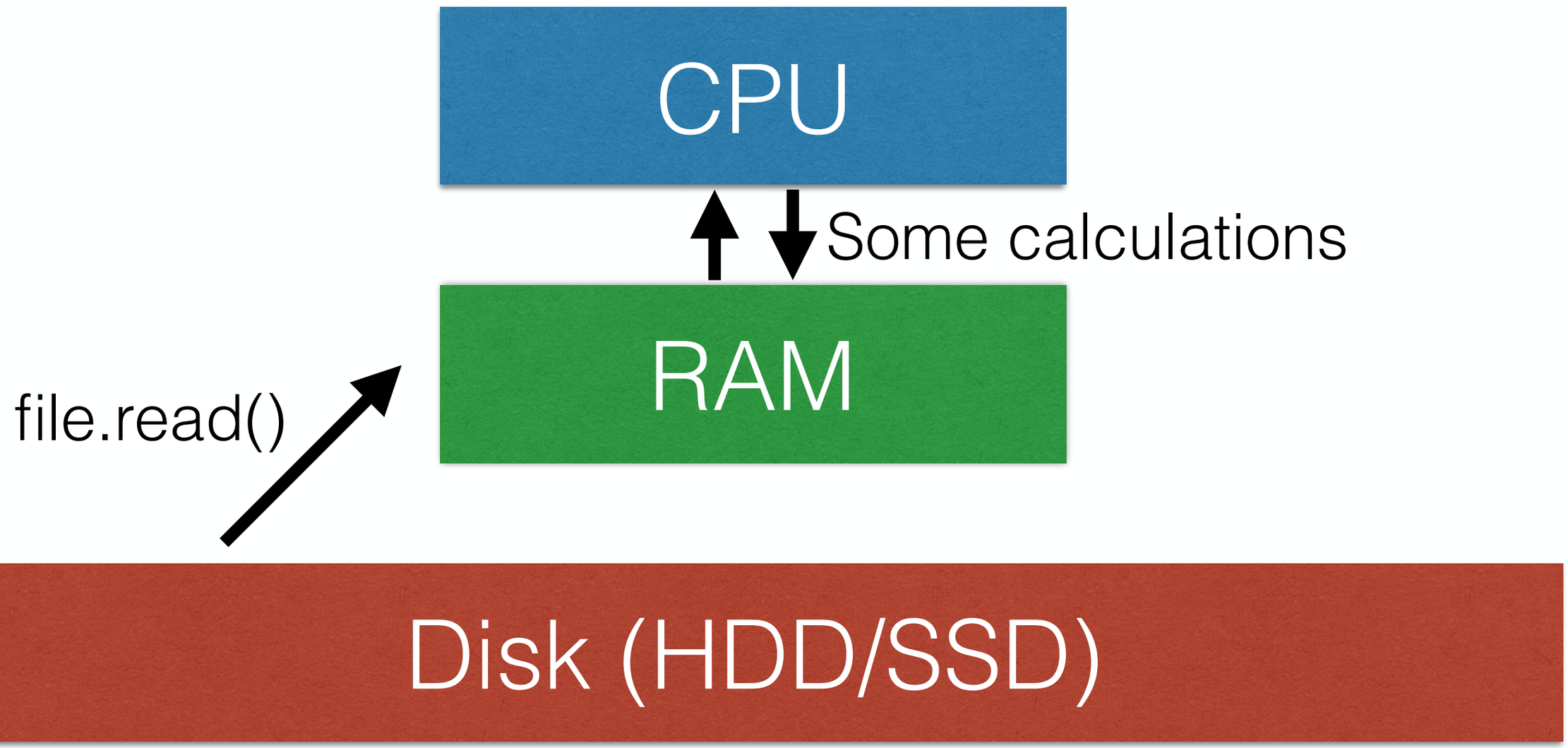
RAM ~8GB

Disk (HDD/SSD) ~1 TB

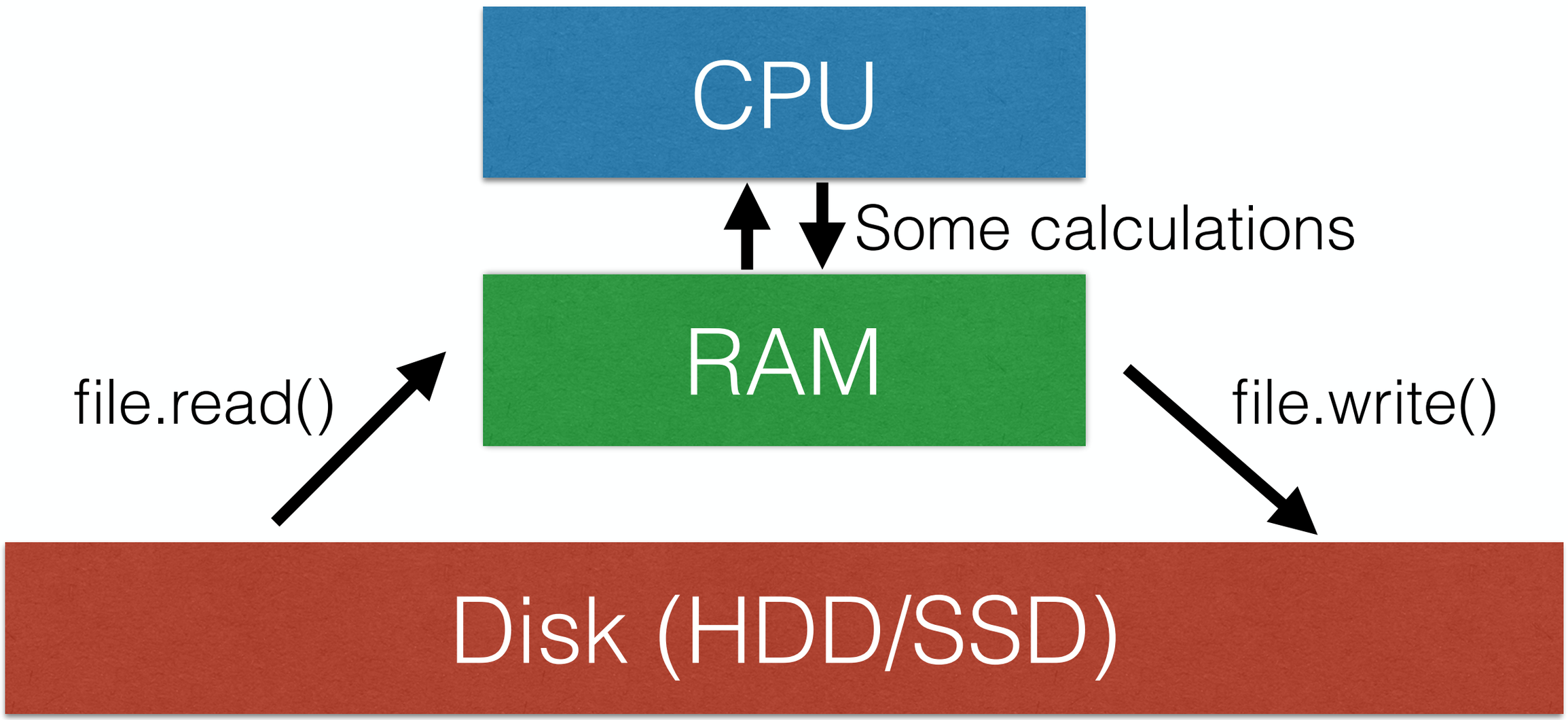
Super simplified computer



Super simplified computer



Super simplified computer



The Internet

- There are 993,059,597 websites online right now

http://www.internetlivestats.com/
http://www.webperformancetoday.com/2012/05/24/average-web-page-size-1-mb/

The Internet

- There are 993,059,597 websites online right now
- 10 pages per site => 10 billion pages

http://www.internetlivestats.com/
http://www.webperformancetoday.com/2012/05/24/average-web-page-size-1-mb/

The Internet

- There are 993,059,597 websites online right now
- 10 pages per site => 10 billion pages
- 1 MB per page

http://www.internetlivestats.com/
http://www.webperformancetoday.com/2012/05/24/average-web-page-size-1-mb/

The Internet

- There are 993,059,597 websites online right now
- 10 pages per site => 10 billion pages
- 1 MB per page
- 10 000 TB ! (~ 10 PB)
- Wikipedia is “only” 10 TB

http://www.internetlivestats.com/
http://www.webperformancetoday.com/2012/05/24/average-web-page-size-1-mb/

Let’s count words!

- Read web page into memory

Let’s count words!

- Read web page into memory
- Count words on the page

Let’s count words!

- Read web page into memory
- Count words on the page
- Add numbers to dictionary {(word, count)}

Let’s count words!

- Read web page into memory
- Count words on the page
- Add numbers to dictionary {(word, count)}
- Repeat!

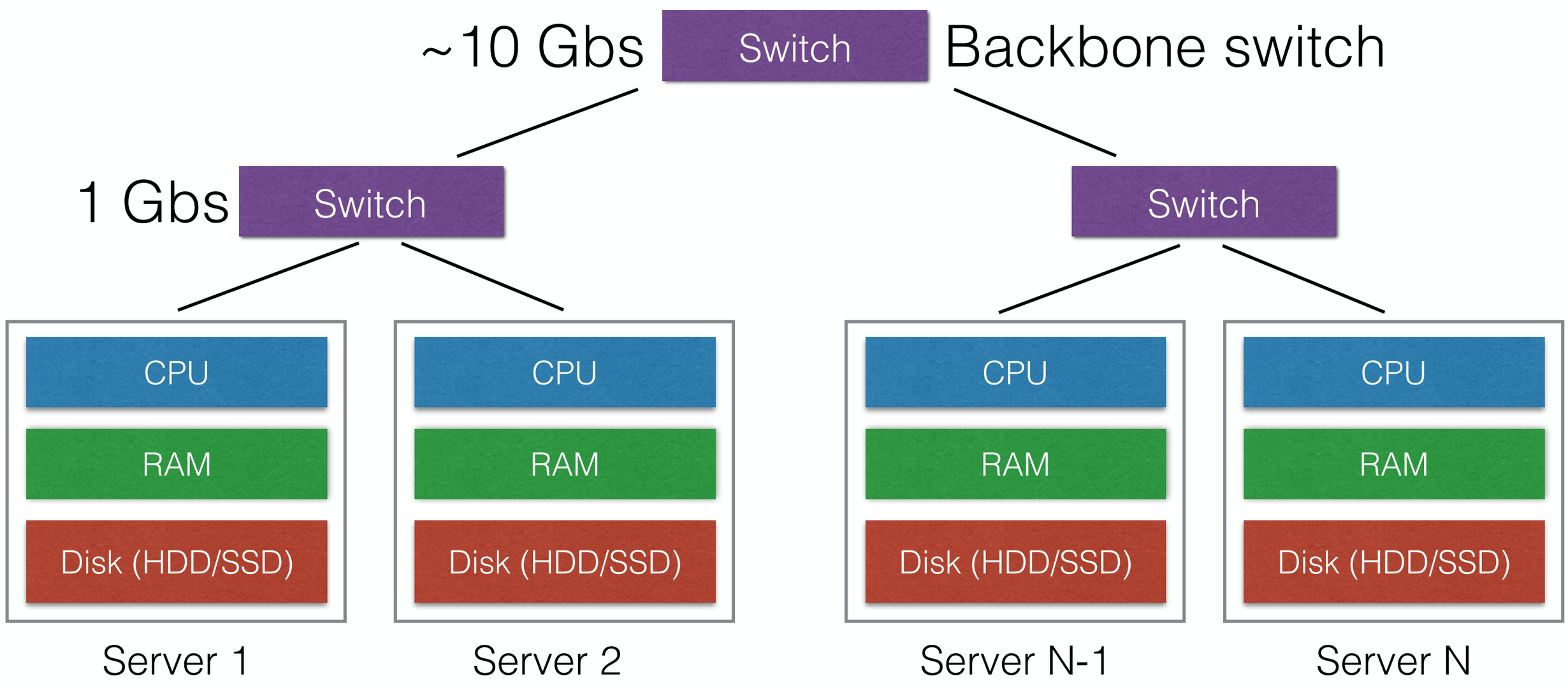
WordCount algorithm

- 10 000 TB to process
- 100 MB/s HDD read speed
- ~ 104857600 s = 1200+ days
- Even with infinite RAM and top processor the run time id more than 3 years

Get technical

Parallelisation

- The only possible way is to use many machines
- Machines are connected to racks (2-50 machines in rack)




google.com/datacenters

Node failures

- Hard-loaded computer fails every 3 years (1000 days)
- 10K servers in data center …
- 10 failures/day

Node failures

- Hard-loaded computer fails every 3 years (1000 days)
- 10K servers in data center …
- 10 failures/day

- How to save the data?
- How to deal with computations?

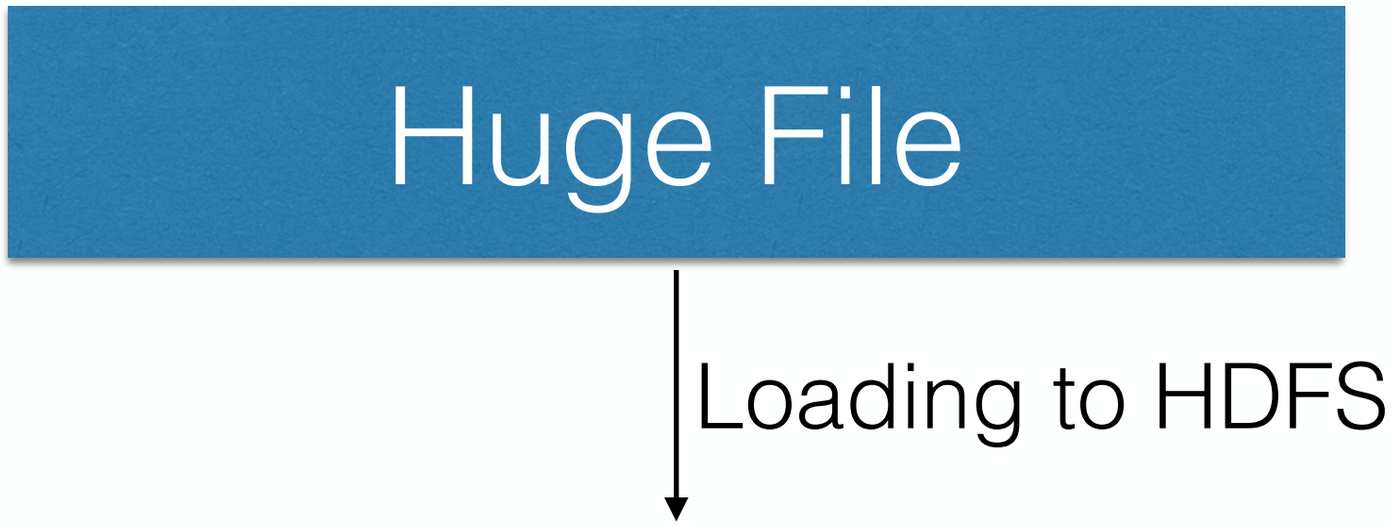
Distributed File System

- It’s not good to store huge files on single server
- Backups are needed

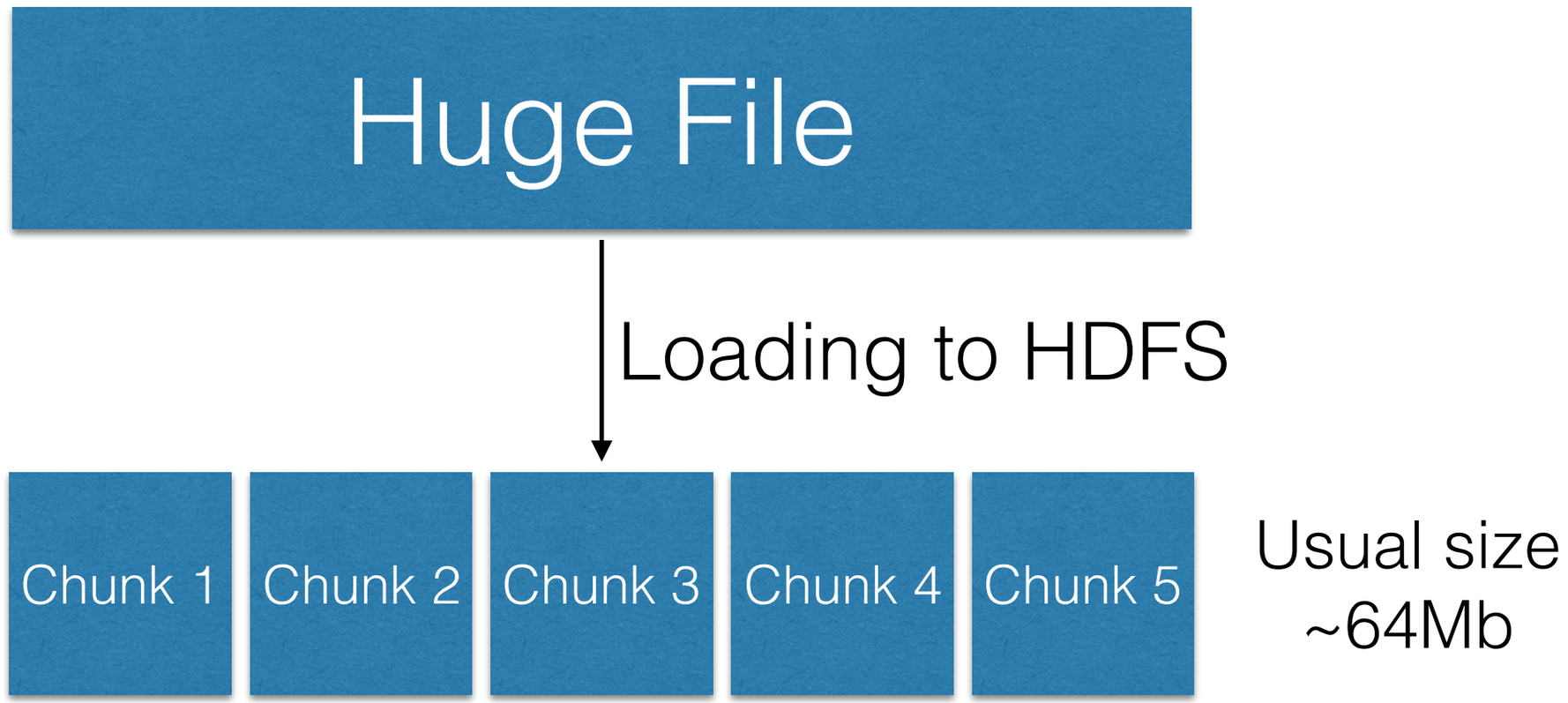
- Most famous realisations:
  - GFS (Google File System), closed
  - HDFS (Hadoop Distributed File System), open-source, part of Hadoop project

- We will talk about second one

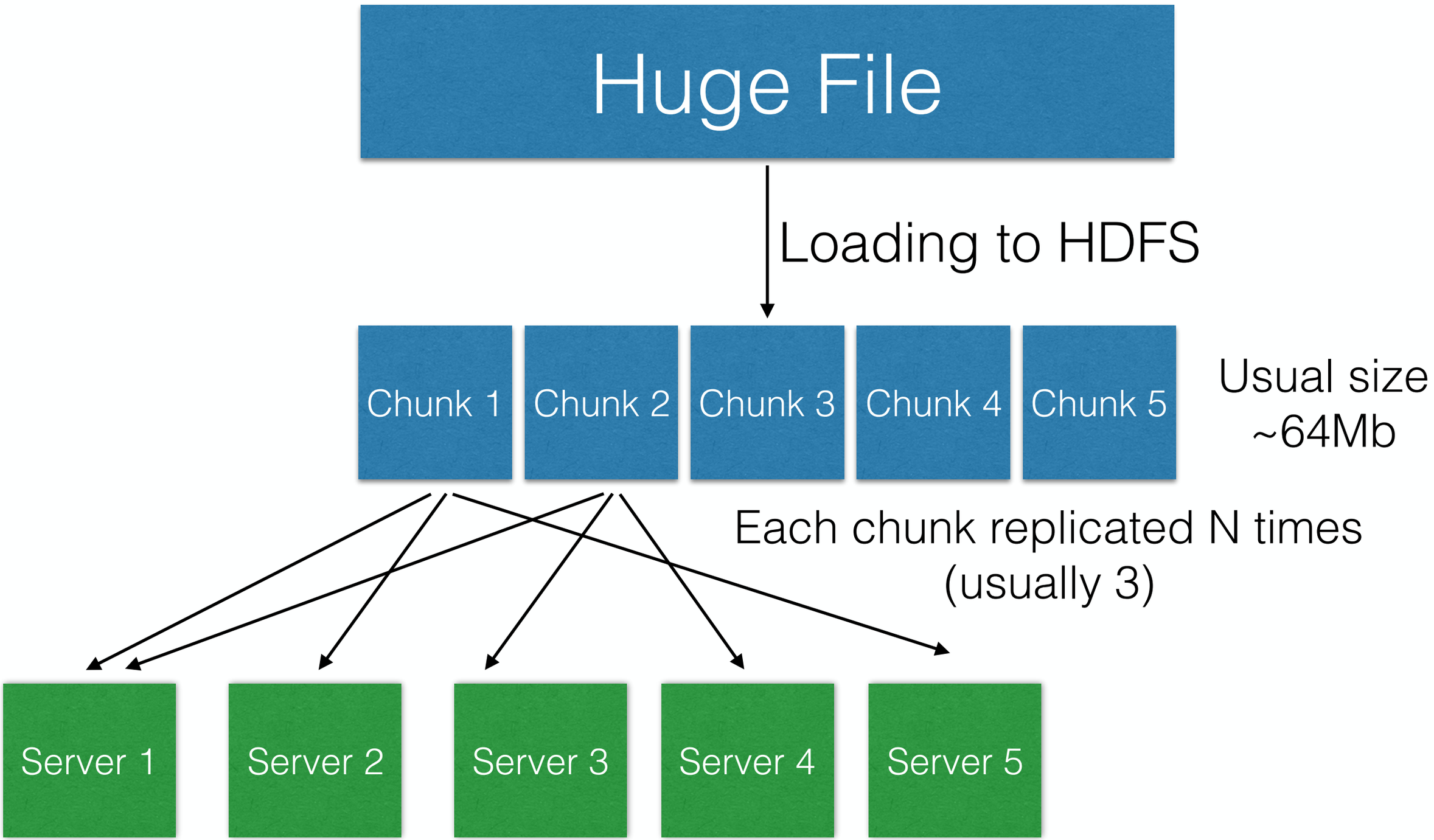
Distributed File System



Distributed File System



Distributed File System



Distributed File System

Huge File

Loading to HDFS

Chunk 1 | Chunk 2 | Chunk 3 | Chunk 4 | Chunk 5

Usual size
~64Mb

Each chunk replicated N times
(usually 3)

Server 1 | Server 2 | Server 3 | Server 4 | Server 5

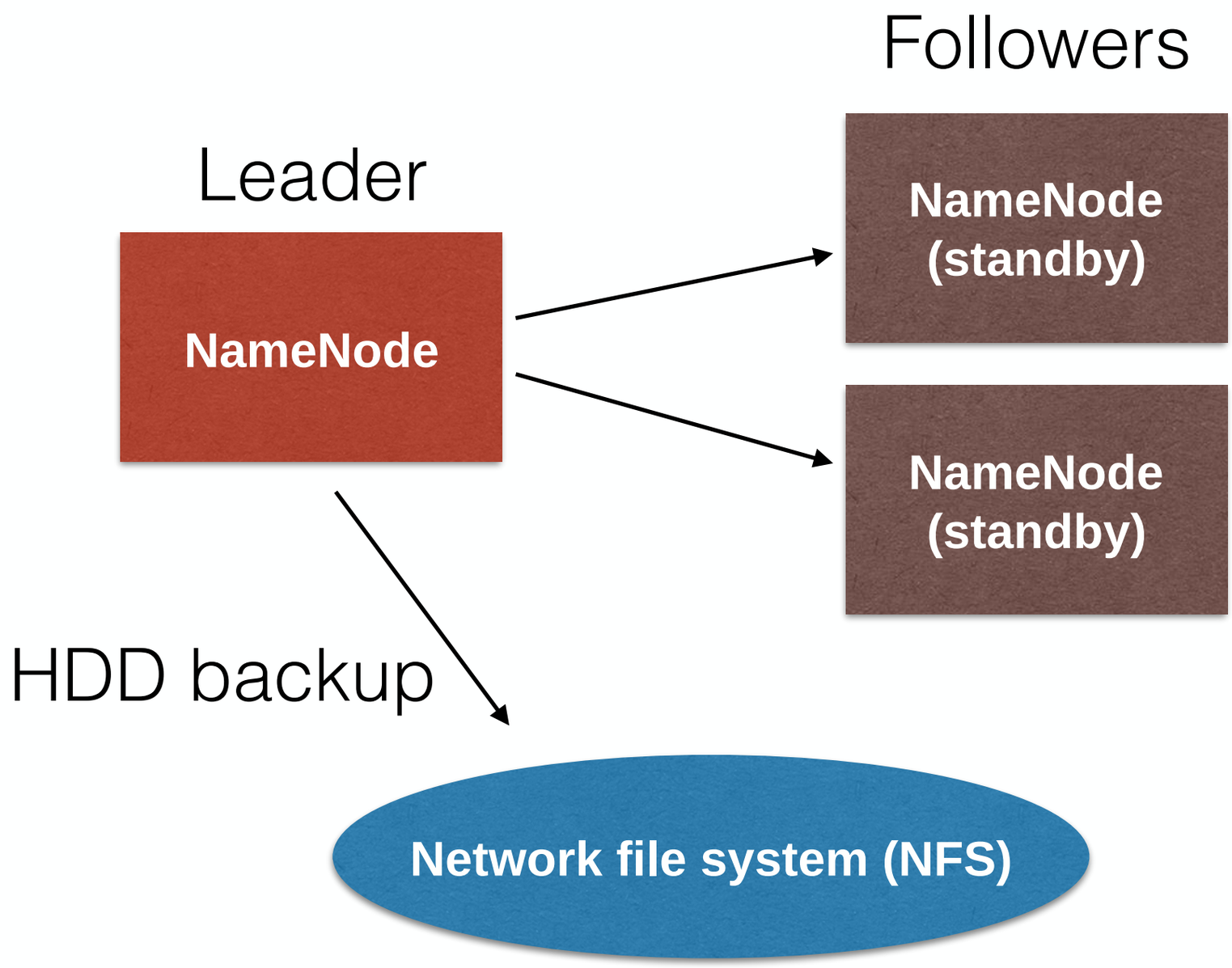
NameNode

Knows where to find chunks

Where are points of failure?

1. Backbone switch (no connection between racks)
2. Rack switch (no connection in the rack)
3. Servers's HDD
4. NameNode network connection
5. NameNode HDD

NameNode


Followers
Leader
NameNode
NameNode (standby)
NameNode (standby)
HDD backup
Network file system (NFS)

Computations

Parallel sorting

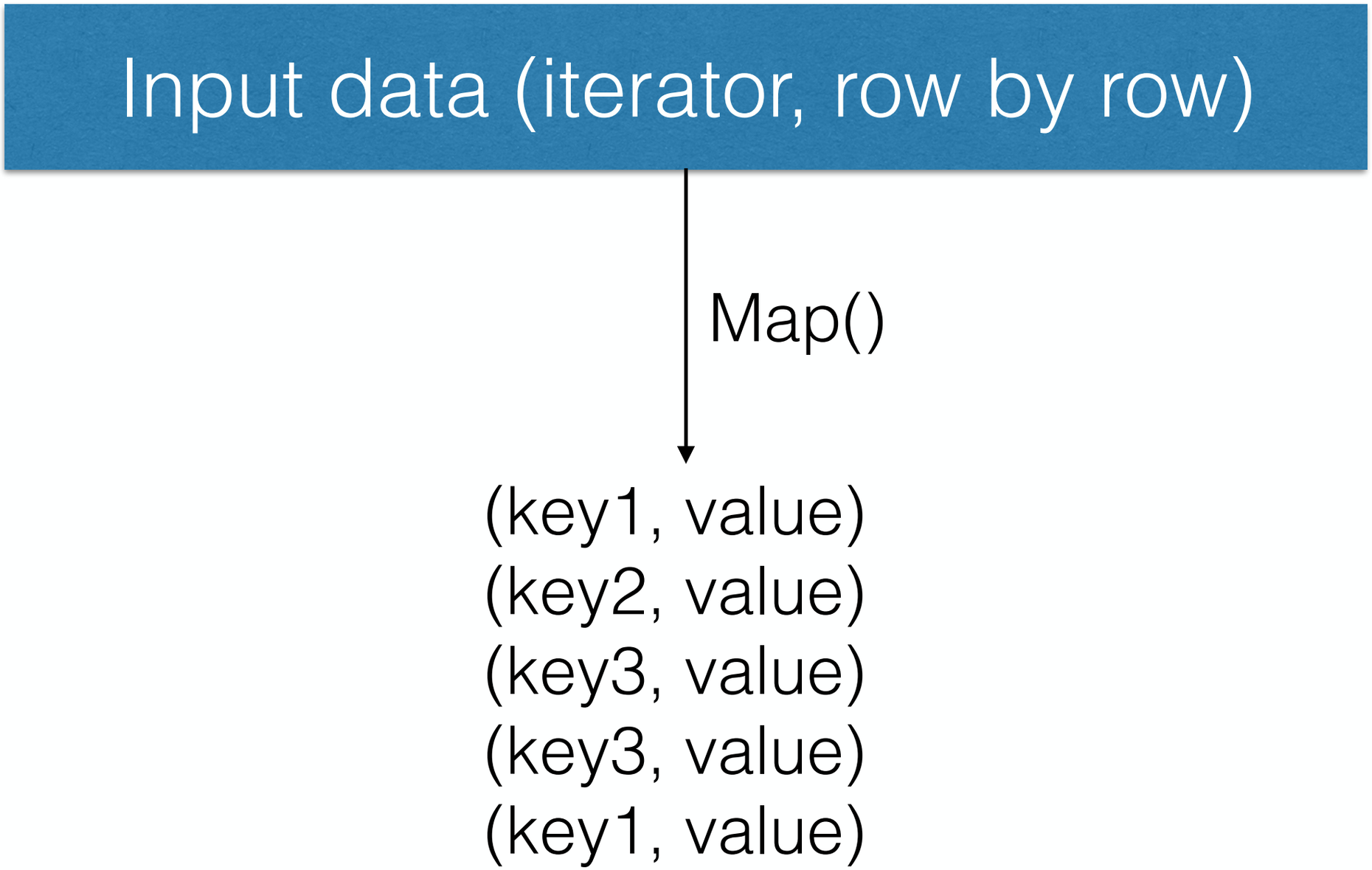
- Various algorithms (e.g. Merge Sort)
- Google results (on 10k machines data center):
  - 2007: 1 PB / 12.13 hours
  - 2008: 1 PB / 6.03 hours
  - 2010: 1 PB / 2.95 hours
  - 2011: 1 PB / 0.55 hours
  - 2012: 50 PB / 23 hours
  - Why did not they go on?

https://cloud.google.com/blog/big-data/2016/02/history-of-massive-scale-sorting-experiments-at-google

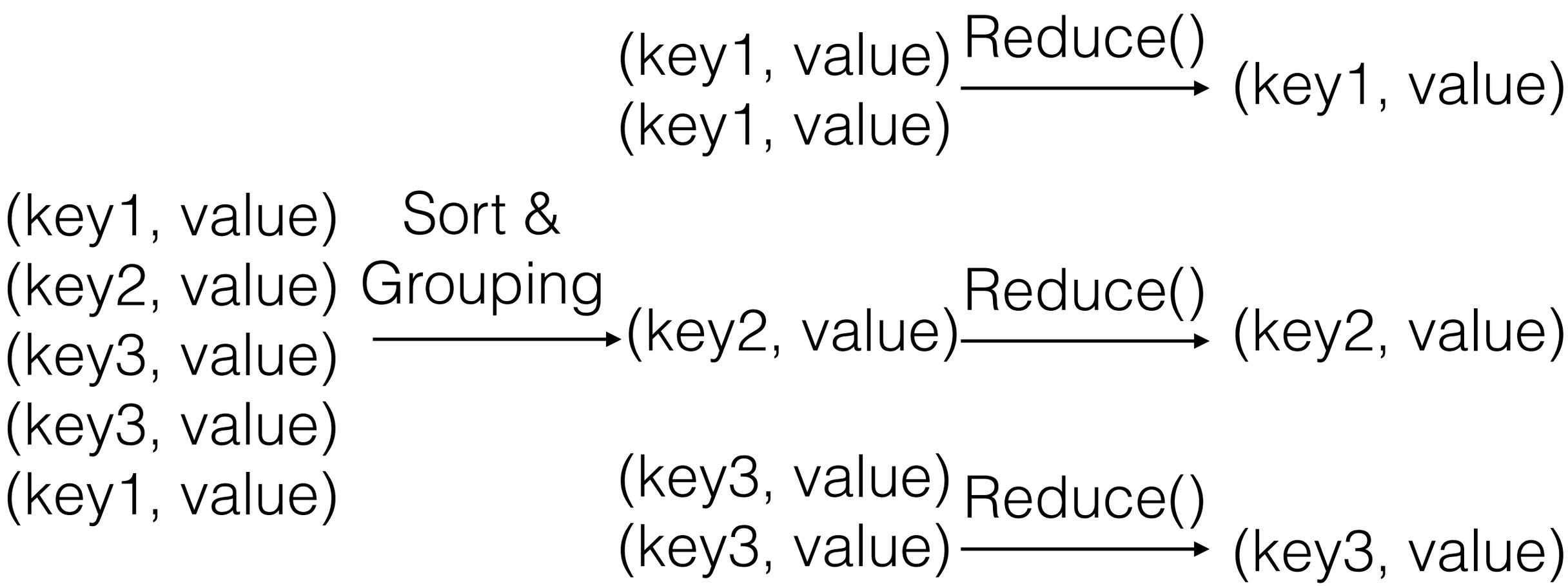
MapReduce

- Largle-scale computational model
- Released by Google (known since 1995)
- Natively parallelised
- Various problems could be solved
- **Must-know on any data scientist interview**

Step one : Map


Input data (iterator, row by row)
Map()
(key1, value)
(key2, value)
(key3, value)
(key3, value)
(key1, value)

Step two : Reduce



Example

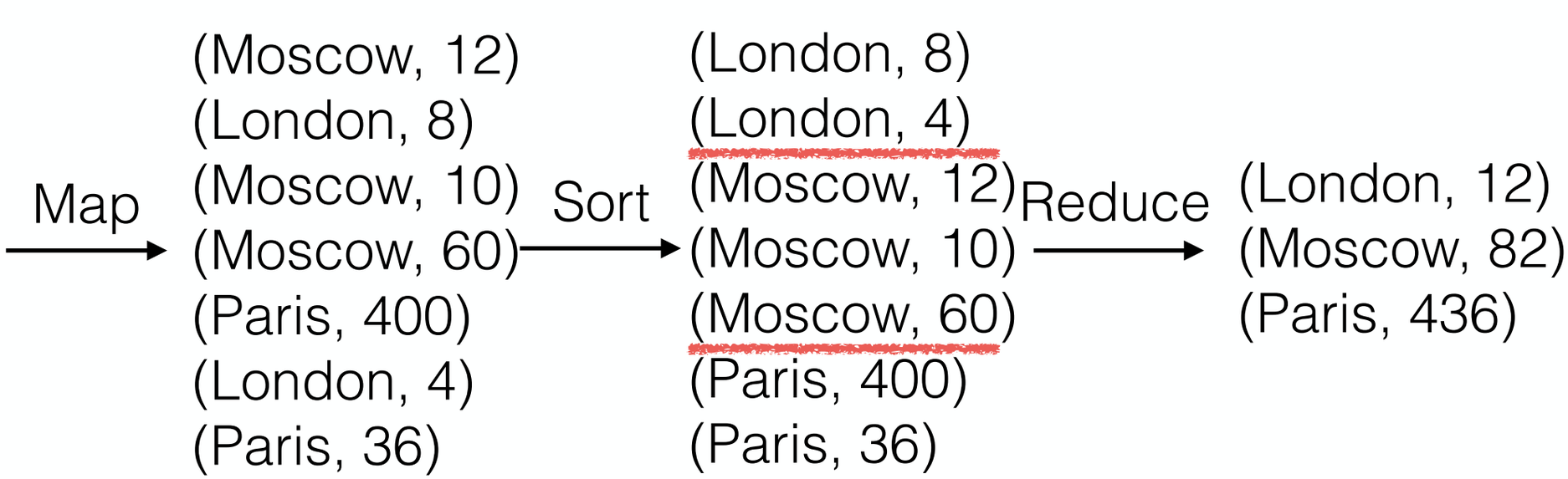
We need to calculate revenue by city of the international shop

Shop	Category	Value	Price	Revenue
Moscow	closes	1	12	12
London	closes	1	8	8
Moscow	music	2	5	10
Moscow	toys	12	5	60
Paris	music	4	100	400
London	closes	1	4	4
Paris	music	6	6	36

Example

Map: Take row and return (city, revenue)
Reduce: Sum all values for the key

Shop	Revenue
Moscow	12
London	8
Moscow	10
Moscow	60
Paris	400
London	4
Paris	36



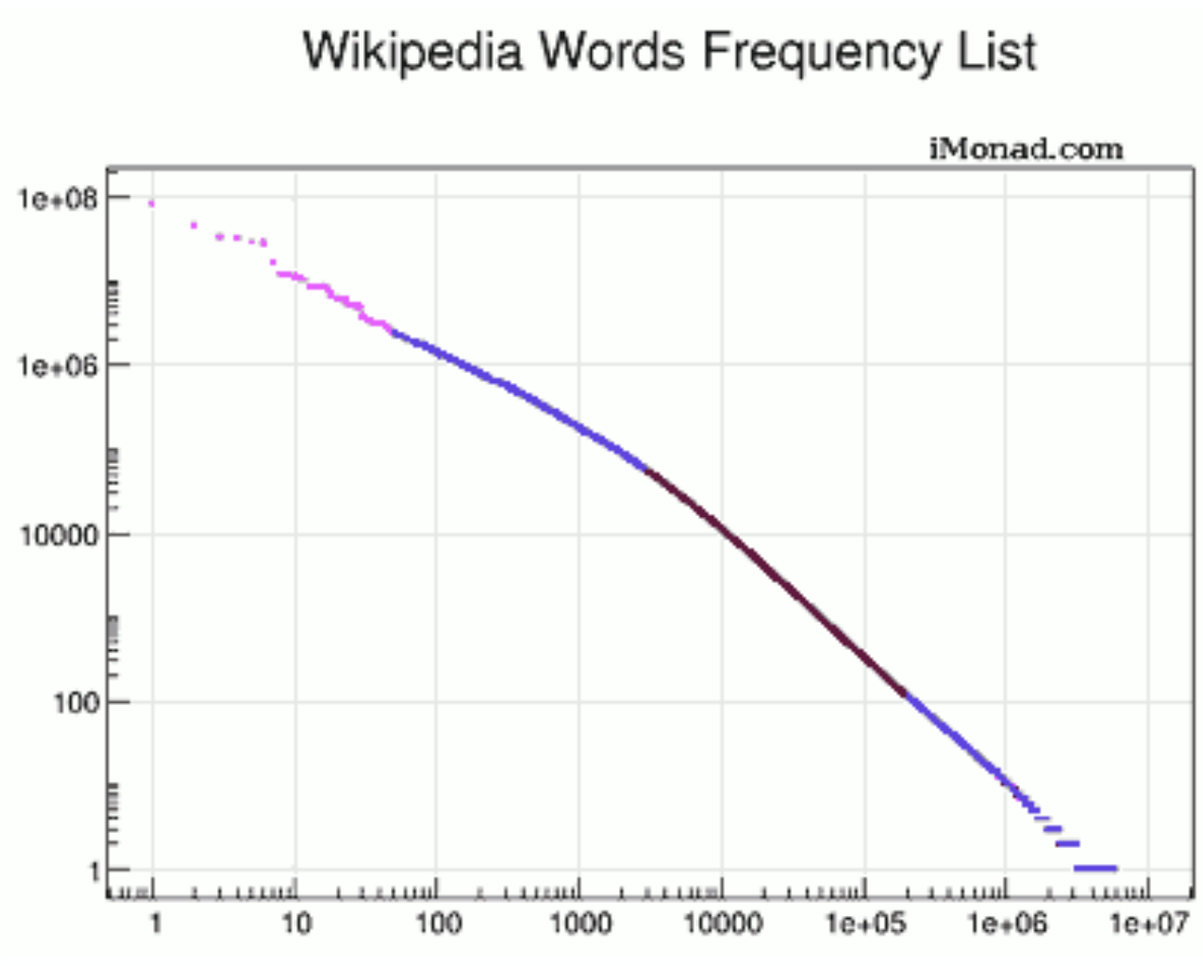
More examples

- Revenue by category
- Revenue by shop and category
- Mean revenue by the shop
- Uniq stores
- Histogram of sales

More about WordCount

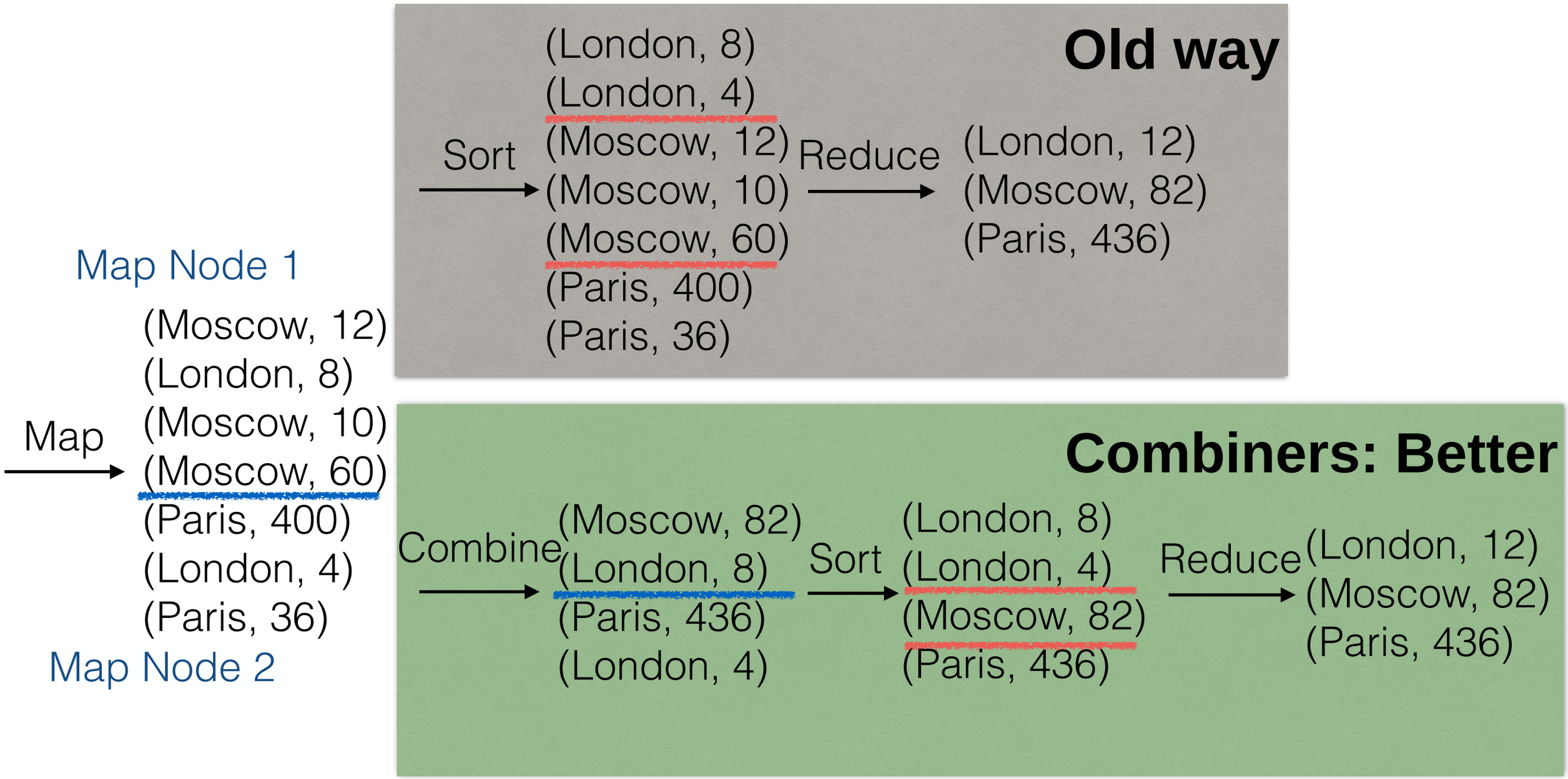
- “Hello World” of MapReduce: Word count
  (on 10000 machines 1200 days become 4 hours)

- The one problem - “monsters”
  some reducers will get ~1e8 (key, value) pairs



Combiners

- Combiners are reduce function (usually) run on the map node after mapping, before sorting



Restrictions on combiners

- Commutative and associative
  - f(a,b) = f(b,a)
  - f(a,f(b,c)) = f(f(a,b),c)

- Sum, prod
- Mean- why? how to solve the problem?
- Median, quantiles - why? what to do?

Environment

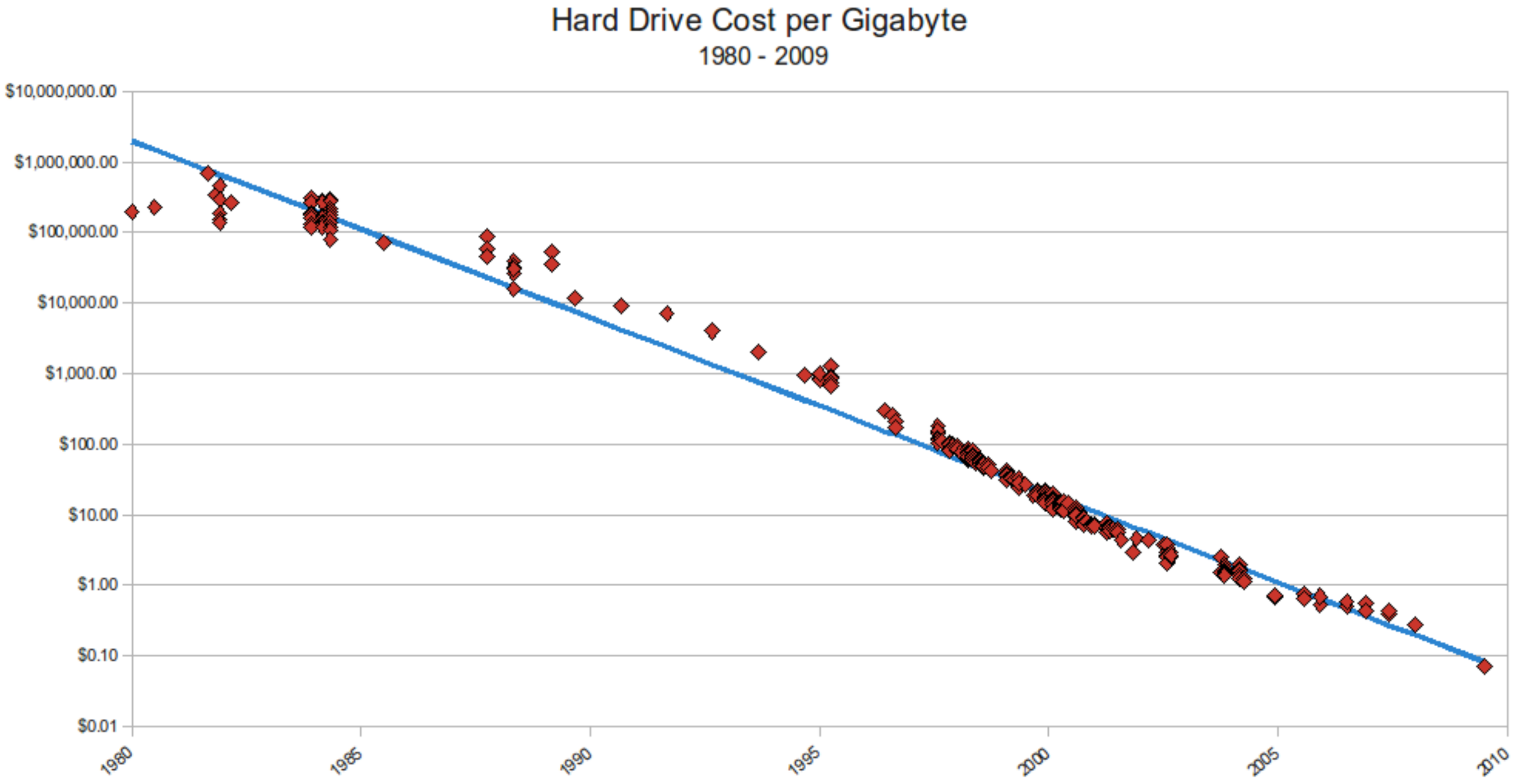
- Partitioning
- Scheduling
- Running processes near the data
- Grouping
- Handling failures
- Managing all inter-machine communications
  (you need just to specify two functions on any language)

What about node failure?

1. Map Node failure during running?
2. Reduce Node failure during running?
3. Master Node?

Few more words about Big Data

Why?


Hard Drive Cost per Gigabyte
1980 - 2009
$10,000,000.00
$1,000,000.00
$100,000.00
$10,000.00
$1,000.00
$100.00
$10.00
$1.00
$0.10
$0.01
1980
1985
1990
1995
2000
2005
2010

3V

1. **Volume**
2. **Variety**

   If something in the data may be wrong, it will
3. **Velocity**

   Yandex Real Time Crypta: 250k RPS, 15 TB/day

   (Wikipedia: 30-70k RPS, Reddit DDoS: 400k)

Correlations

- On the enormous amount of samples even weak correlations become meaningful

Observation: people buy beer with diapers

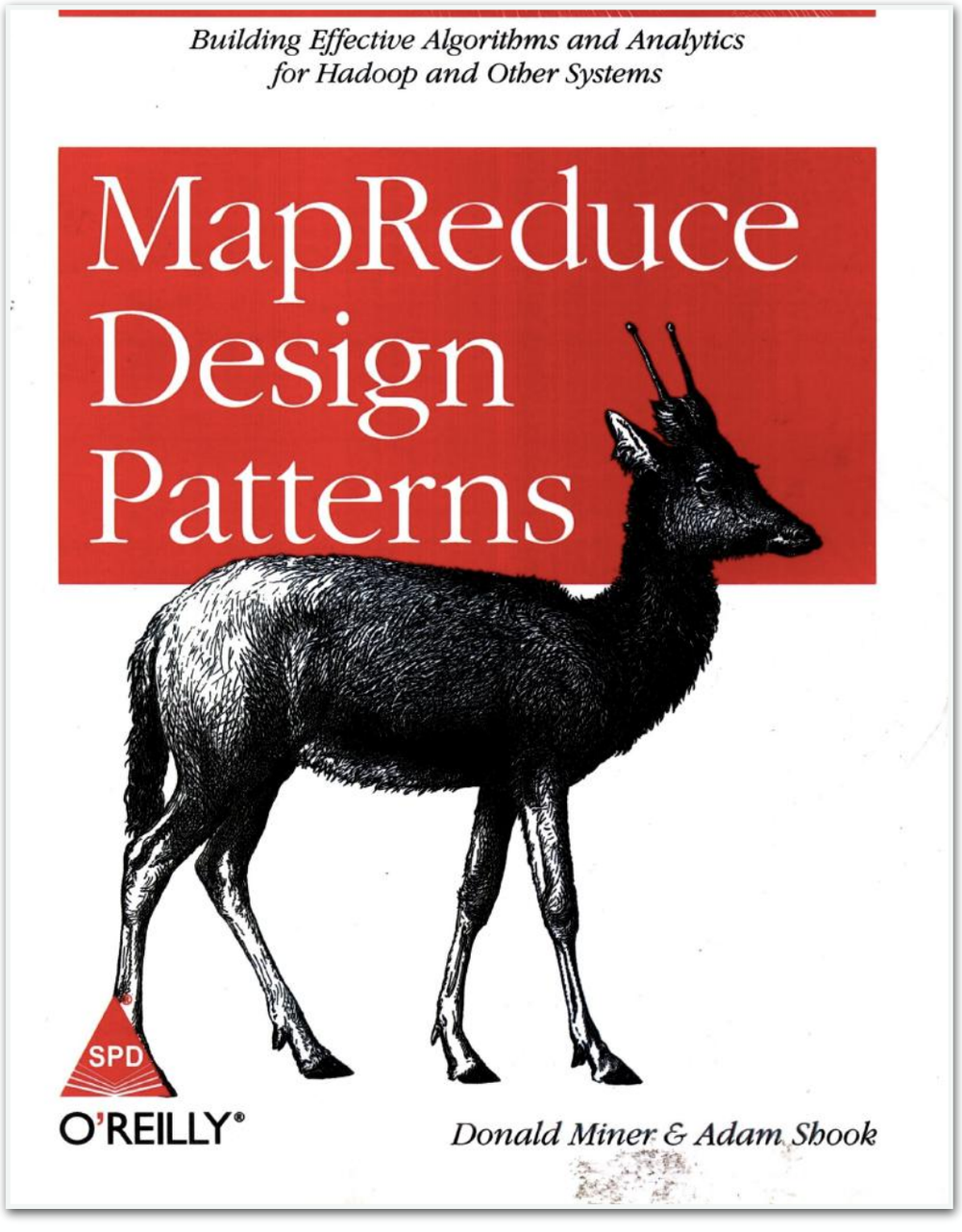
- The classic way: check p-value, use Granger causality test.
- The Big Data way: doesn't matter.
  Let's just make money on this correlation

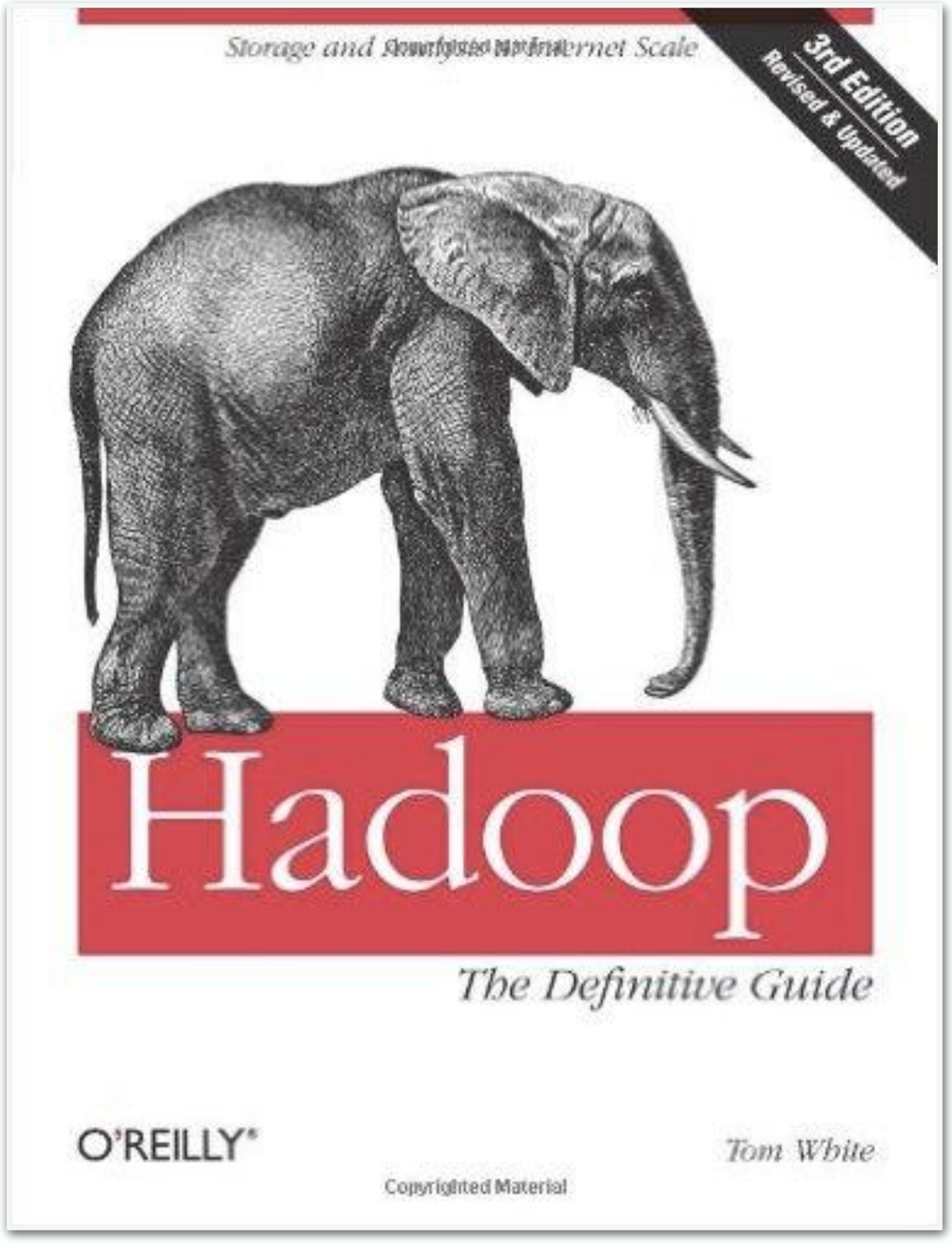
What to do next?

Read a popular book

Russian translation is also good

Read a tech book





Get familiar with Hadoop

- Check out the CDH by Cloudera
- Read about Hive & Pig
- Run local single-node pseudo cluster
- Play with Amazon AWS EMR (Elastic Map Reduce)

  10 machines for 0.15$/hour
- Run your own Hadoop on AWS

Study

- Mining Massive Data Sets @ Coursera, mmds.org
- Introduction to Hadoop and MapReduce @ Udacity
- Big Data Specialisation @ Coursera

“A real data scientist(TM) can implement algorithms, write proofs, setup Hadoop clusters, perform RCA, talk to clients, and doesn’t exist.”

Somewhere on Twitter