

**Additional Topics: Big Data**

# **Lecture #1**

**An overview of “Big Data”**

Joseph Bonneau  
jcb82@cam.ac.uk  
April 27, 2012

# Course outline

- 0 – Google on Building Large Systems (Mar. 14)
  - David Singleton
- 1 – Overview of Big Data (today)
- 2 – Algorithms for Big Data (April 30)
- 3 – Case studies from Big Data startups (May 2)
  - Pete Warden

# Goals after four lectures

- recognise some of the main terminology
- remember that there are many tools available
- realise the potential of Big Data
  - all of you are skilled enough to get involved!

# What is Big Data?

Data sets that grow so large that they become awkward to work with using on-hand database management tools? (Wikipedia)

# What is Big Data?

- distributed file systems
- NoSQL databases
- grid computing, cloud computing
- MapReduce and other new paradigms
- large-scale machine learning

# What is Big Data?



from “Big Data and the Web: Algorithms for Data Intensive Scalable Computing”  
PhD Thesis, Gianmarco De Francisci Morales

# What is Big Data?

- buzzword?
- bubble?
- gold rush?
- revolution?
- funding fad?
  - DARPA XDATA project, March 2012

**Who's profiting?**



# RapLeaf

```
{  
  "age": "21-24",  
  "gender": "Male",  
  "interests": {  
    "Blogging": true,  
    "High-End Brand Buyer": true,  
    "Sports": true,  
  },  
  "education": "Completed Graduate School",  
  "occupation": "Professional",  
  "children": "No",  
  "household_income": "75k-100k",  
  "marital_status": "Single",  
  "home_owner_status": "Rent"  
}
```



- 10 TB of user data
- all computation done using local Hadoop cluster

# Gnip



- “grand central station” for social web streams
- aggregates several TB of new social data **per day**
- all data stored on Amazon S3

# Climate Corporation

The screenshot shows the Climate Corporation website. At the top left is the logo, which consists of a stylized blue and green leaf-like shape next to the text "THE CLIMATE CORPORATION". To the right of the logo are three links: "Look Up Policy", "Contact Us", and "Agent Login". Below these links are two buttons: "FOR GROWERS" and "FOR AGENTS". The main banner features a smiling man in a blue t-shirt and a green baseball cap standing in a field under a cloudy sky. The text "Total Weather Insurance" is prominently displayed, followed by the tagline "Protect Your Profits From Bad Weather". Below this, there are four numbered steps in green circles: 1. Get Your Weather Risk Report (with a bar chart icon), 2. Get Custom Weather Insurance Plan (with a document icon), 3. Weather Happens (with a sun and cloud icon), and 4. Get Paid Automatically (with a dollar sign and envelope icon). On the right side of the banner, there is a green box with the text "Start Here" and "Get your FREE WEATHER RISK ANALYSIS". Below this text are two input fields: "County or Zip Code" and a dropdown menu currently showing "Corn". At the bottom of this green box is an orange button labeled "Get Started". In the bottom right corner of the banner, it says "BRENT B., ILLINOIS TWI 2012 INSURED".

- 14 TB of historical weather data
- all computation done using Amazon EC2
- 30 technical staff, including 12 PhDs
- 10,000 sales agents

# FICO

**FICO**<sup>™</sup>

ProductsServicesIndustriesDiscussionsPartnersCompanySearch


## Make connections that sell

Use analytics to determine what, how and when customers will buy

Products > Decision Management Applications > FICO® Retail Action Manager

### FICO® Retail Action Manager

FICO Retail Action Manager is a marketing decision application that predicts individual customer sales propensities across multiple products over time, maximizing the effectiveness of sales, merchandising and promotional efforts. Retail Action Manager uses Best Next Action™ predictive analytics to make smarter customer predictions, and optimization to recommend the immediate actions that will meet longer-term sales, marketing and merchandising objectives.



#### Request Information

Enter your info below and we'll respond to you directly

First Name

Last Name

Email Address

Title

Select Job Level

Company

- 50+ years of experience doing credit ratings
- transitioning to predictive analytics

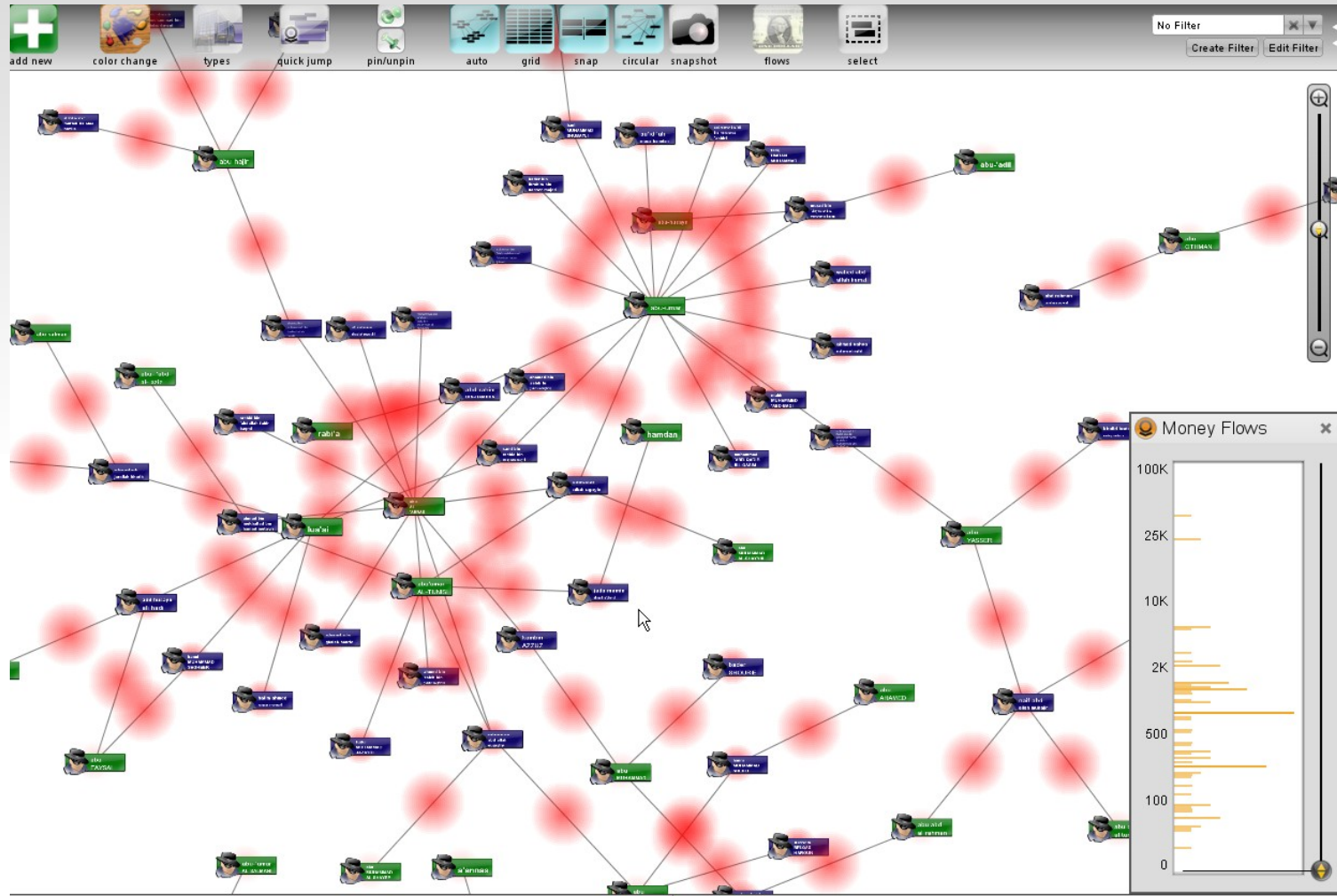


# Cloudera



- employs many of the original Hadoop developers
- now facing many competitors

# Palantir Technologies



- data analysis, exploration tools for government, finance
- 0-\$2 billion in 5 years

# The modern data scientist

- Engineer
  - collect & scrub disparate data sources
  - manage a large computing cluster
- Mathematician
  - machine learning
  - statistics
- Artist
  - visualise data beautifully
  - tell a convincing story

# Sources of Big Data



# Proprietary data: Facebook

- 40+ billion photos
  - 100 PB
- 6 billion messages **per day**
  - 5-10 TB
- 900 million users
  - 1 trillion connections?

# Common Crawl



- covers about 5 million web pages
- 50 TB of data
- available for free, hosted by Amazon

# Twitter “Firehose”



**Clarence Hill**  
@clarencehilljr

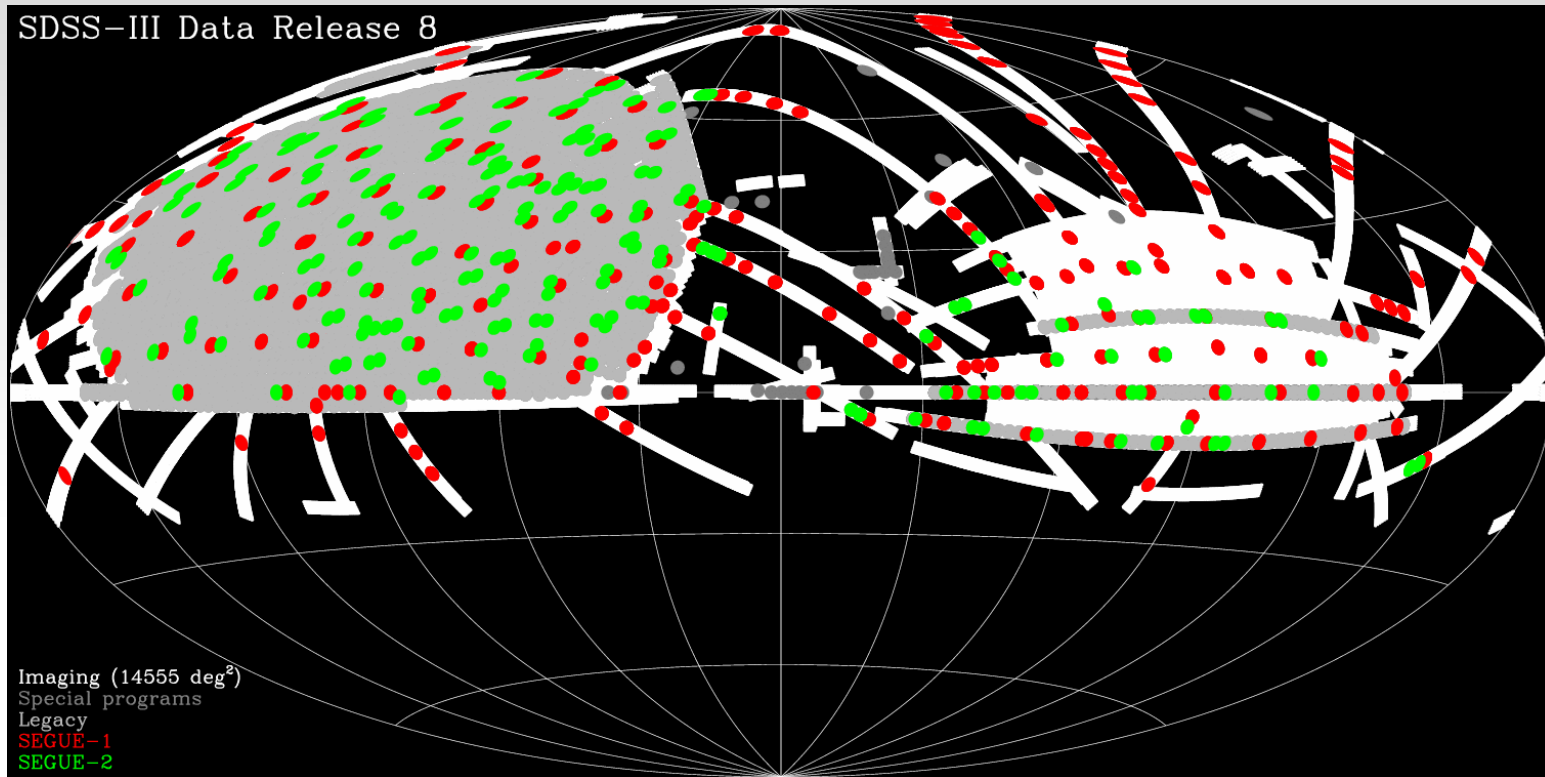


**@PriscoCBS** come on..you doubting the genius of  
parcells..he is never wrong, never made a  
mistake..his football philosophies are dead on

```
{ "in_reply_to_status_id_str":null, "retweet_count":0, "favorited":false, "text":"New iPad vs iPad 2: which should  
you choose? http://t.co/EpygqtlE Redsn0w 0.9.10b6 i4Siri Proxy Server For Spire GEVEY Ultra S WP7  
_83", "in_reply_to_user_id_str":null, "in_reply_to_status_id":null, "created_at":"Mon Mar 19 22:22:32 +0000  
2012", "geo":null, "in_reply_to_user_id":null, "truncated":false, "source":"<a href=\"http://google.com\"  
rel=\"nofollow\">Tech Discovery</a>", "id_str":"181868286009548802", "entities":{"hashtags":[], "urls":  
[ { "indices":  
[45,65], "expanded_url":"http://ow.ly/9A4kC", "url":"http://t.co/EpygqtlE", "display_url":"ow.ly/9A4  
kC" } ], "user_mentions":  
[] }, "contributors":null, "in_reply_to_screen_name":null, "place":null, "retweeted":false, "possibly_sensitive_ed  
itable":true, "possibly_sensitive":false, "coordinates":null, "user":  
{ "profile_text_color":"333333", "profile_image_url_https":"https://si0.twimg.com/profile_images/1817878884/ironman  
_normal.png", "screen_name":"ricegyeat", "default_profile_image":false, "profile_background_image_url":"http://a  
0.twimg.com/images/themes/theme1/bg.png", "favourites_count":0, "created_at":"Thu Feb 09 18:26:24 +0000  
2012", "profile_link_color":"0084B4", "verified":false, "friends_count":0, "url":null, "description":"",  
"profile_background_color":"C0DEED", "id_str":"487764951", "lang":"en", "profile_background_tile":false, "l  
isted_count":0, "contributors_enabled":false, "geo_enabled":false, "profile_sidebar_fill_color":"DDEEF6", "lo  
cation":"", "time_zone":null, "protected":false, "default_profile":true, "following":null, "notifications"  
:null, "profile_sidebar_border_color":"C0DEED", "name":"rice  
gyeat", "is_translator":false, "show_all_inline_media":false, "follow_request_sent":null, "statuses_count":82  
53, "followers_count":109, "profile_image_url":"http://a0.twimg.com/profile_images/1817878884/ironman_normal.png",  
"id":487764951, "profile_use_background_image":true, "profile_background_image_url_https":"https://si0.twimg.com  
/images/themes/theme1/bg.png", "utc_offset":null }, "id":181868286009548802}
```

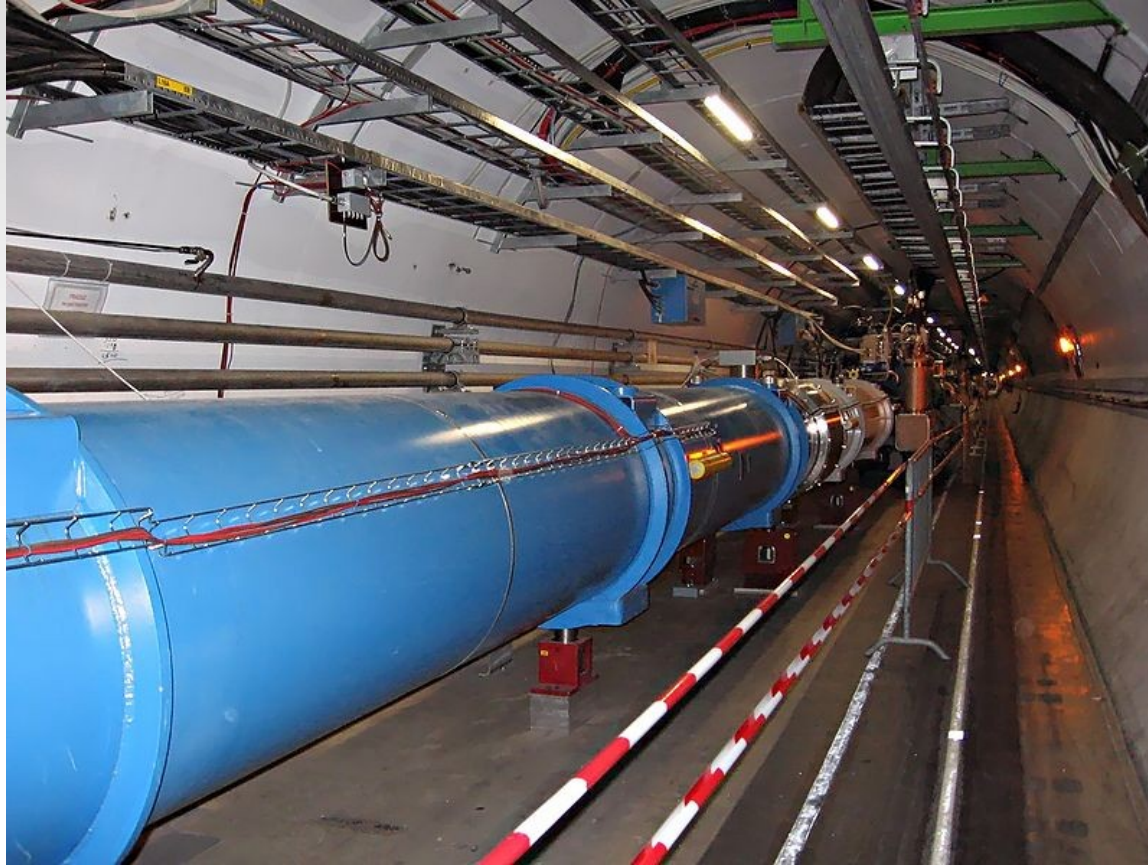
350 M tweets/day x 2-3Kb/tweet  $\approx$  1 TB/day

# Sloan Digital Sky Survey



- 35% of the sky mapped
- 500 million objects classified
- 50 TB of data available

# Large Hadron Collider



- 15 PB of data generated annually
- mostly stored in Oracle databases (SQL)

# Human genome

## Browse a Genome

The Ensembl project produces genome databases for vertebrates and other eukaryotic species, and makes this information freely available online.

Click on a link below to go to the species' home page.

**Popular genomes** ([Log in to customize this list](#))



**Human**

GRCh37



**Mouse**

NCBIM37



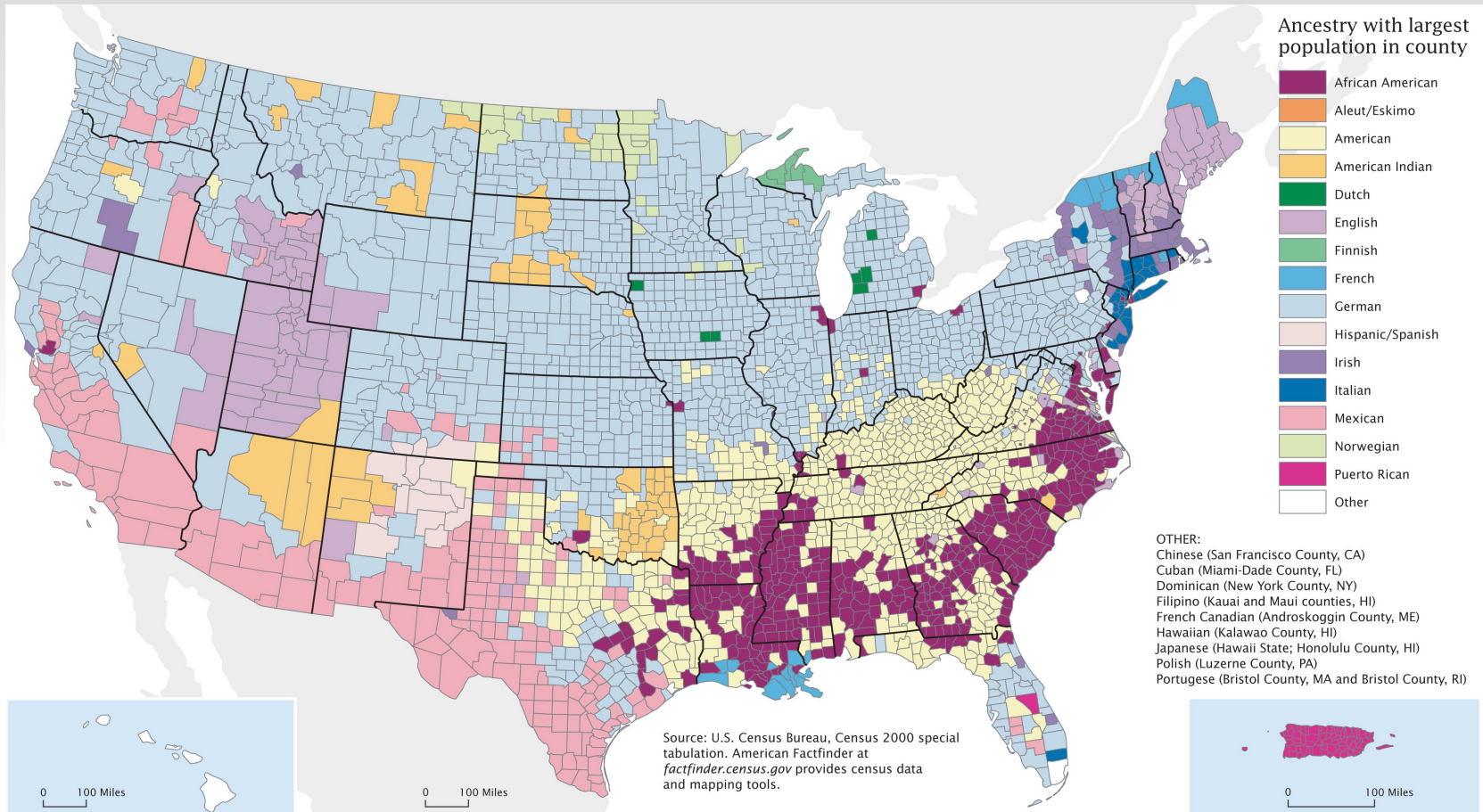
**Zebrafish**

Zv9

- humans and 50 other species
- “only” 250 GB



# US census data (2000)



- detailed demographic data for small localities
- “only” 200 GB

# Roll your own big data



- crawling as a service
- web sites, social profiles, product listings, etc.
- free accounts offer crawls of up to 10k URLs



# Roll your own big data

needle data model sources help feedback logout

## Demo Festivals & Events

From site: [http://njin.state.nj.us/OIT\\_TravelGuide/events.jsp](http://njin.state.nj.us/OIT_TravelGuide/events.jsp)

STATE OF NEW JERSEY  
DIVISION OF TRAVEL & TOURISM

Select a region for detailed information about that area.

SKYLANDS GATEWAY SHORE DELAWARE RIVER ATLANTIC CITY

Great destinations in any direction.

Vacation Packages Free Brochures My Vacation

### Event Details

Escapes: Danger & Survival

Event Site: Location: The Newark Museum

Address: \_Street: 49 Washington St. \_State: NJ \_ZIP Code: 07102

Phone#: Telephone: (973) 596-6550

URL: Link: [www.NewarkMuseum.org](http://www.NewarkMuseum.org)

Description: Description: At the Hazard Ho

start pages

done

guess

Mark as:

Event 1 x

Location 1 x

\_Street 1 x

\_City 1 x

\_State 1 x

\_ZIP Code 1 x

Telephone 1 x

Link 1 x

Data tag buttons automatically created from data model

View of data source (website) with data tags from Needle AI

- Needlebase: graphical tagging of website structure

# Roll your own big data

- Boilermepes: remove “clutter” from web pages
  - Metadata, JavaScript, etc.
- Google Refine: clean up human-entered data
  - fix common typos, spacing, etc.
- NLPToolkit: simplify natural language
  - stem words, replace synonyms
- Lucene: index terms for text search
- Amazon MTurk: human analysis

# Storing big data

# Traditional SQL databases

TABLE instructor

ID	Name
14	David Singleton
27	Joseph Bonneau
52	Pete Warden

TABLE lectures

ID	Title	Lecturer
1	BD at Google	14
2	Overview of BD	27
3	Algorithms for BD	27
4	BD at startups	14

# Traditional SQL databases

TABLE instructor

ID	Name
14	David Singleton
27	Joseph Bonneau
52	Pete Warden

most interesting  
queries require  
computing **joins**

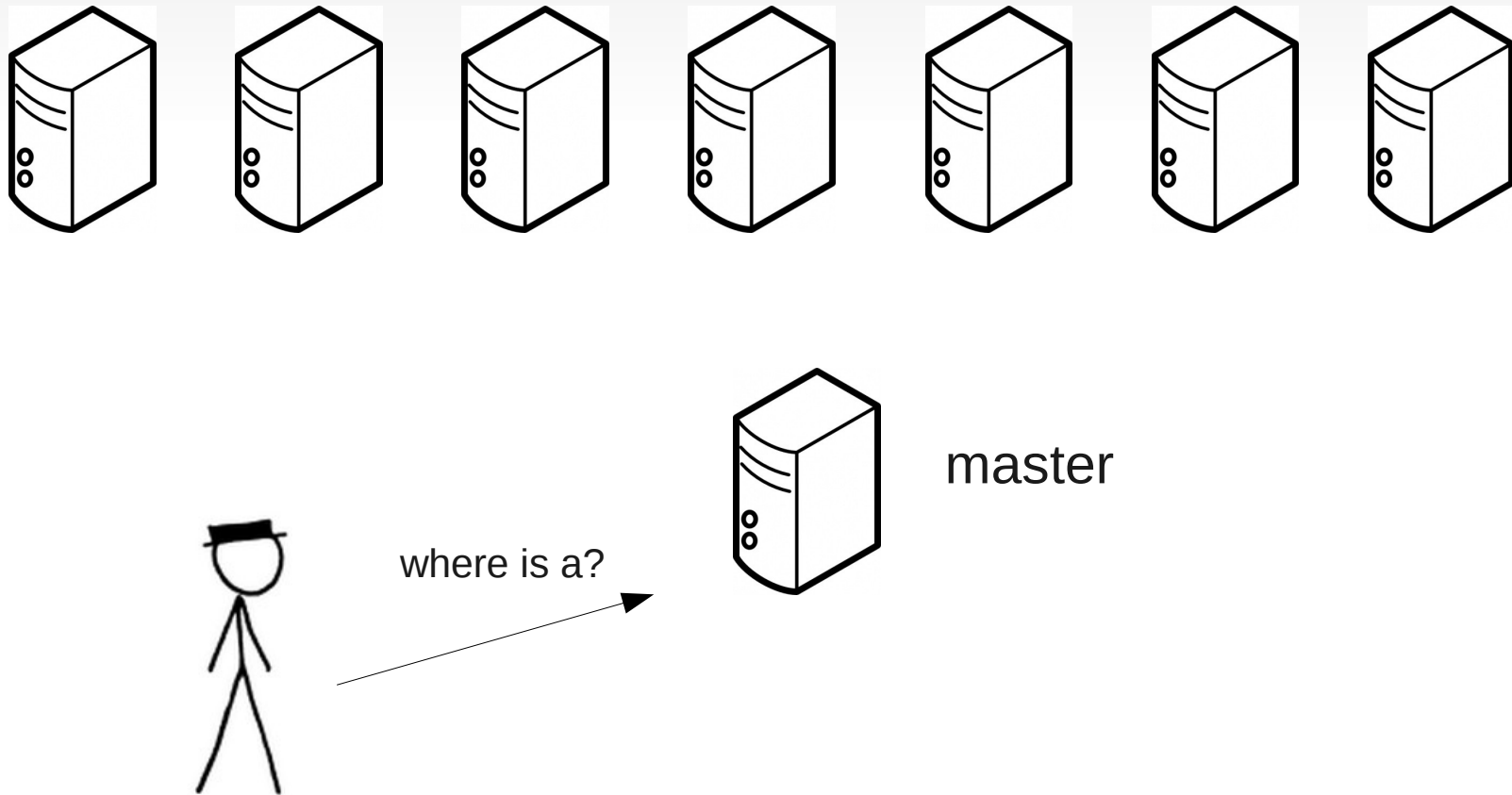
TABLE lectures

ID	Title	Lecturer
1	BD at Google	14
2	Overview of BD	27
3	Algorithms for BD	27
4	BD at startups	14

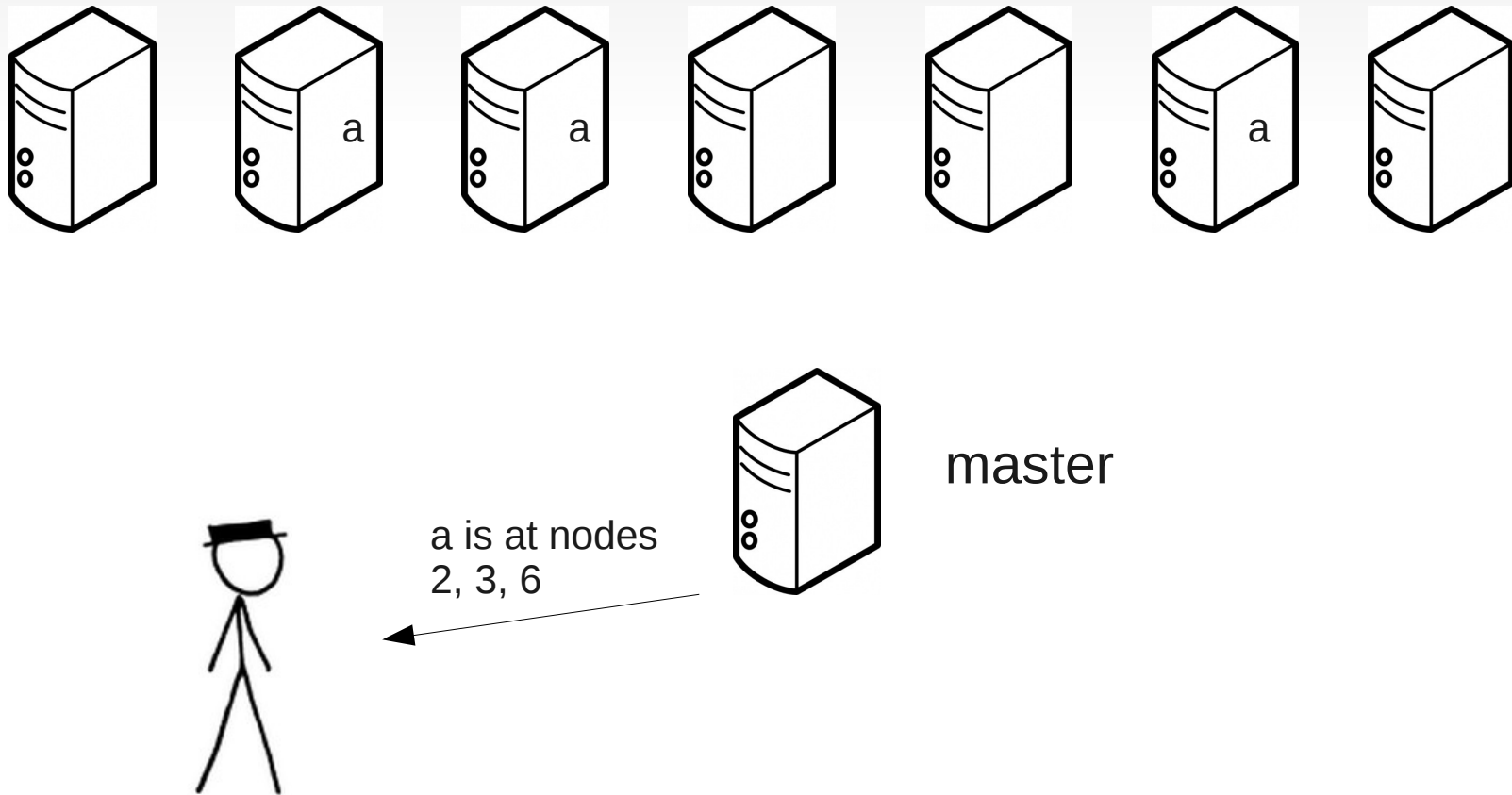
# Data assumptions

Traditional RDBMS (SQL)	NoSQL
integrity is mission-critical	OK as long as most data is correct
data format consistent, well-defined	data format unknown or inconsistent
data is of long-term value	data will be replaced
data updates are frequent	write-once, ready multiple
predictable, linear growth	unpredictable growth (exponential?)
non-programmers writing queries	only programmers writing queries
regular backup	replication
access through master server	sharding

# Replication and sharding

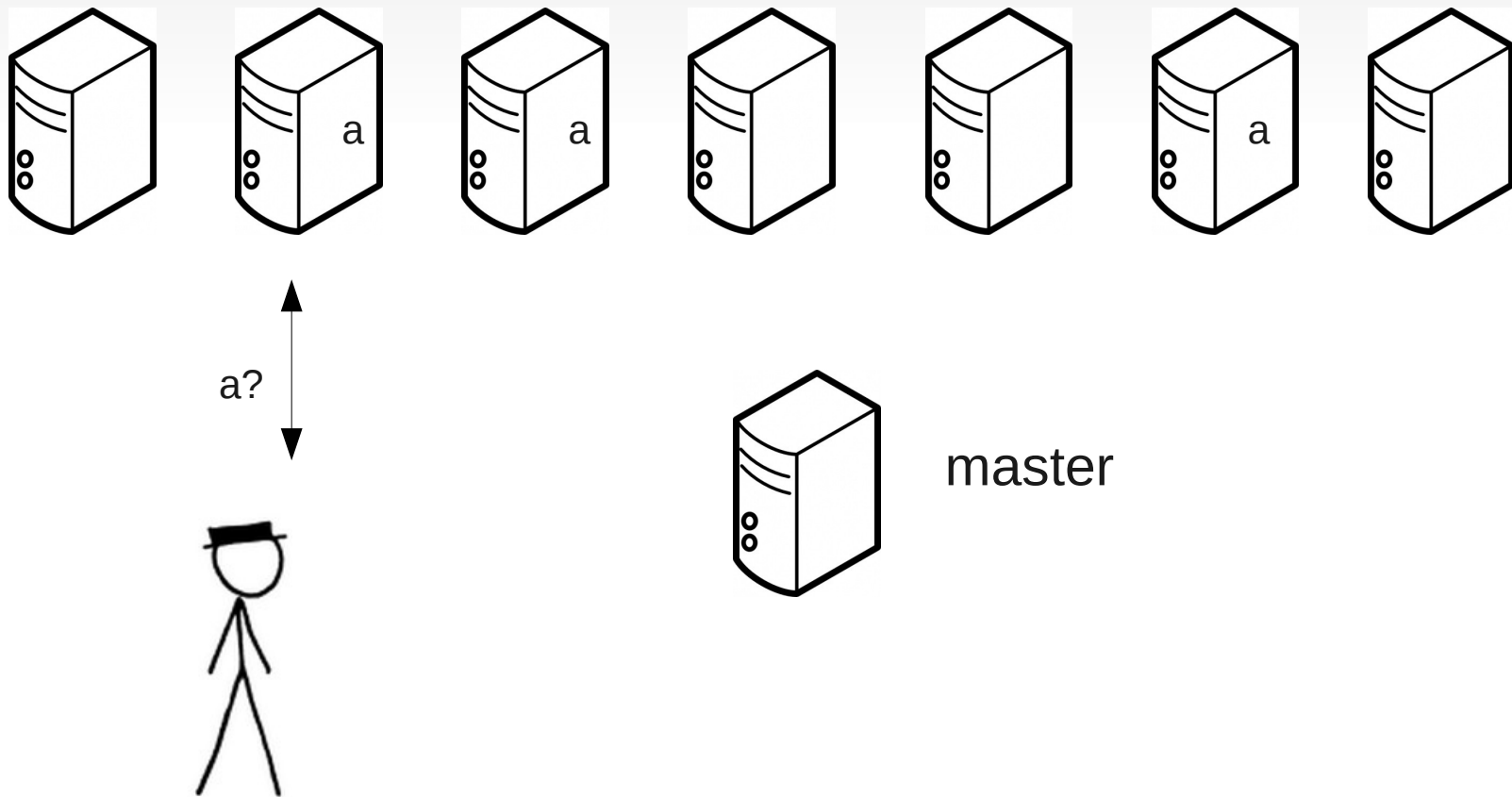


# Replication and sharding





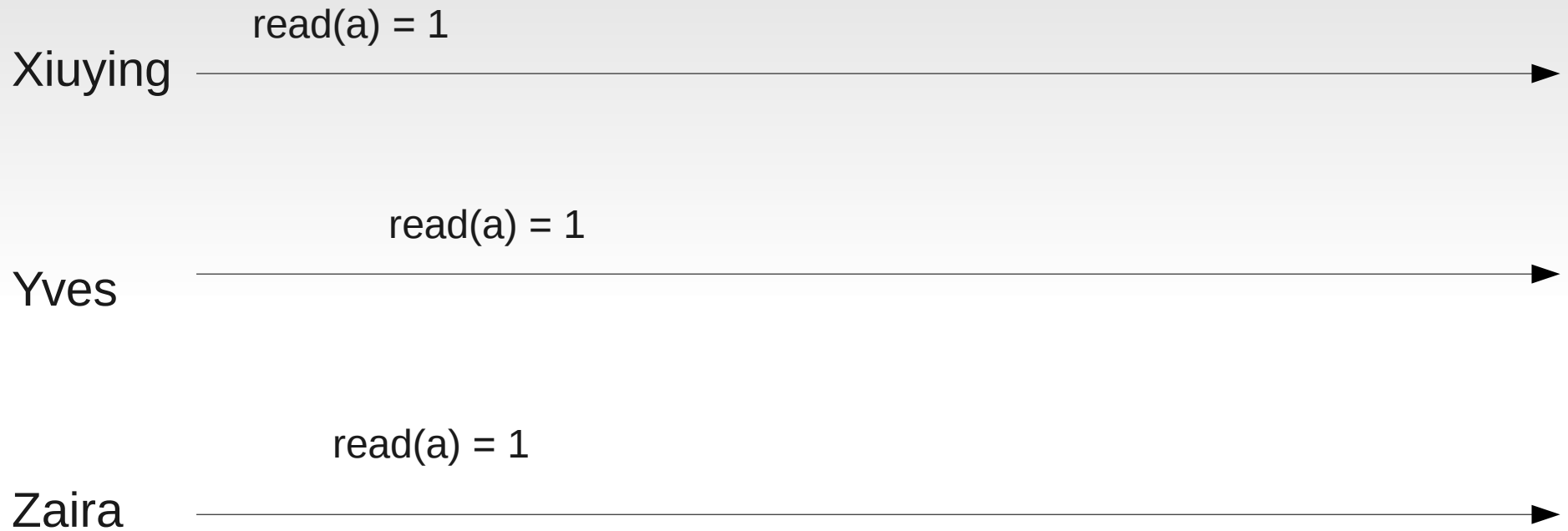
# Replication and sharding



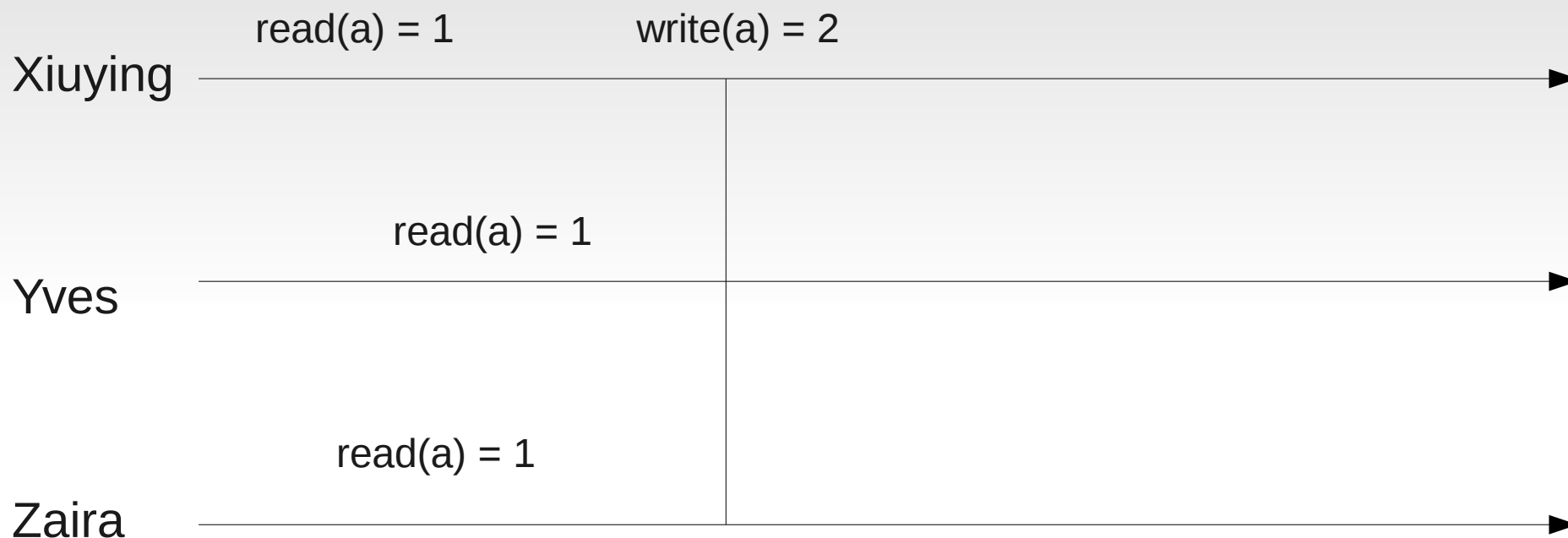
# Integrity guarantees

Traditional RDBMS (SQL)	NoSQL
<b>A</b> vailable	<b>B</b> asically
<b>C</b> onsistent	<b>A</b> vailable
<b>I</b> solated transactions	<b>S</b> oft state
<b>D</b> urable writes	<b>E</b> ventually consistent

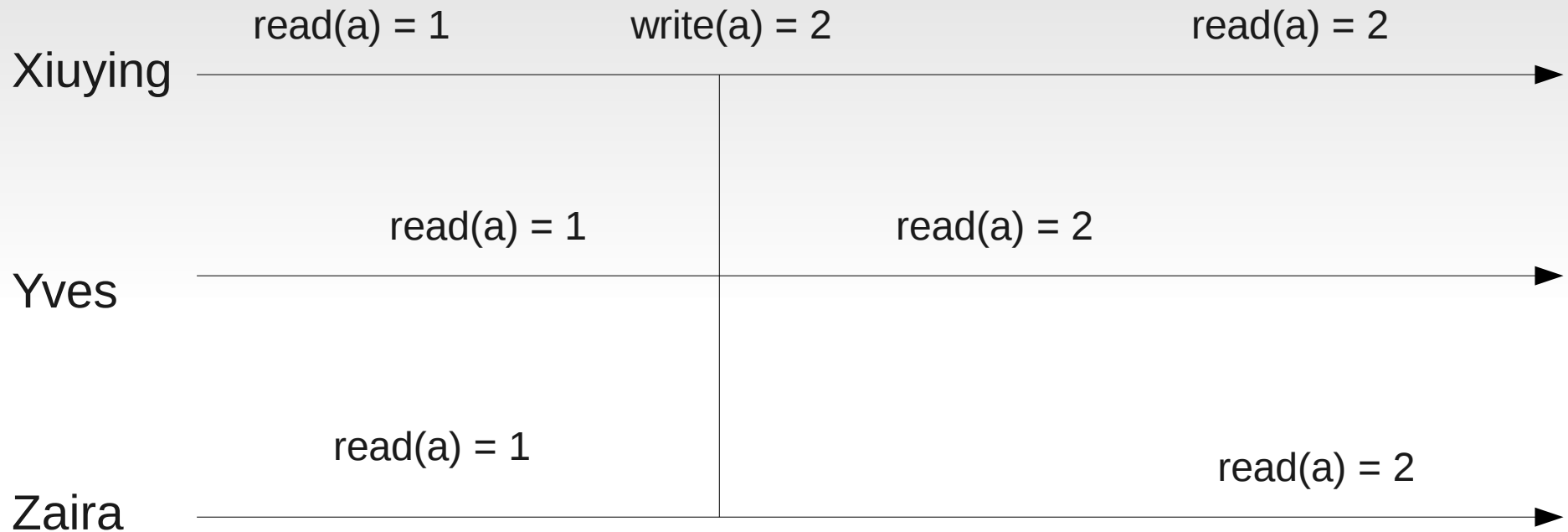
# Strong consistency



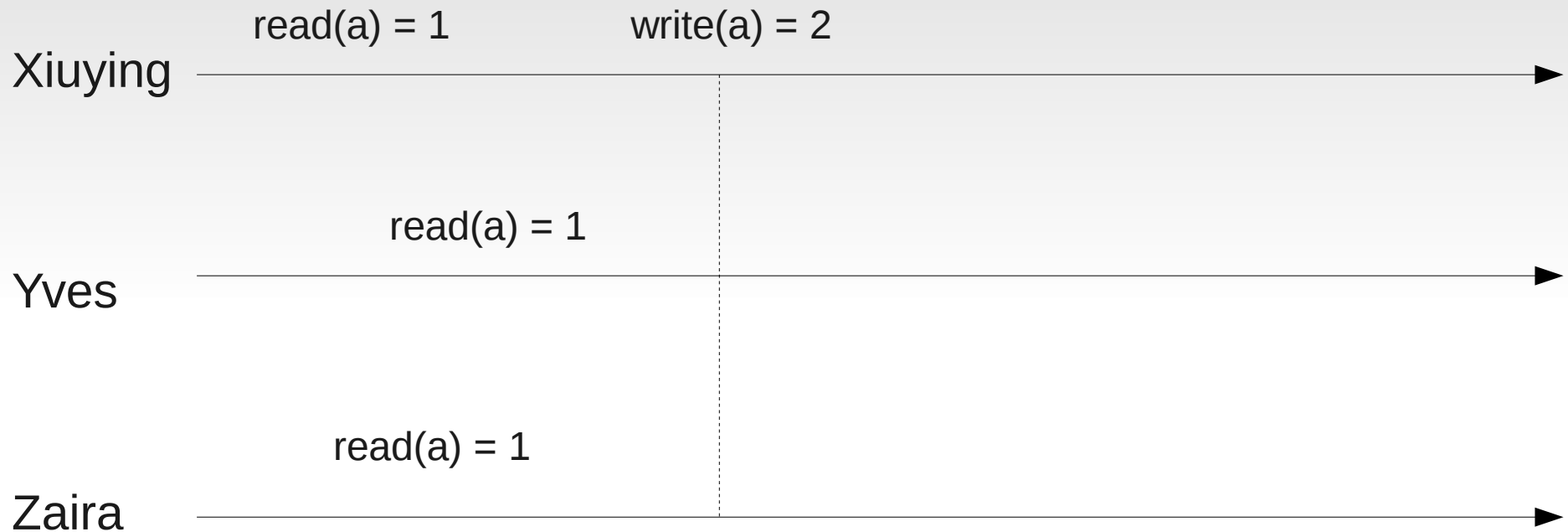
# Strong consistency



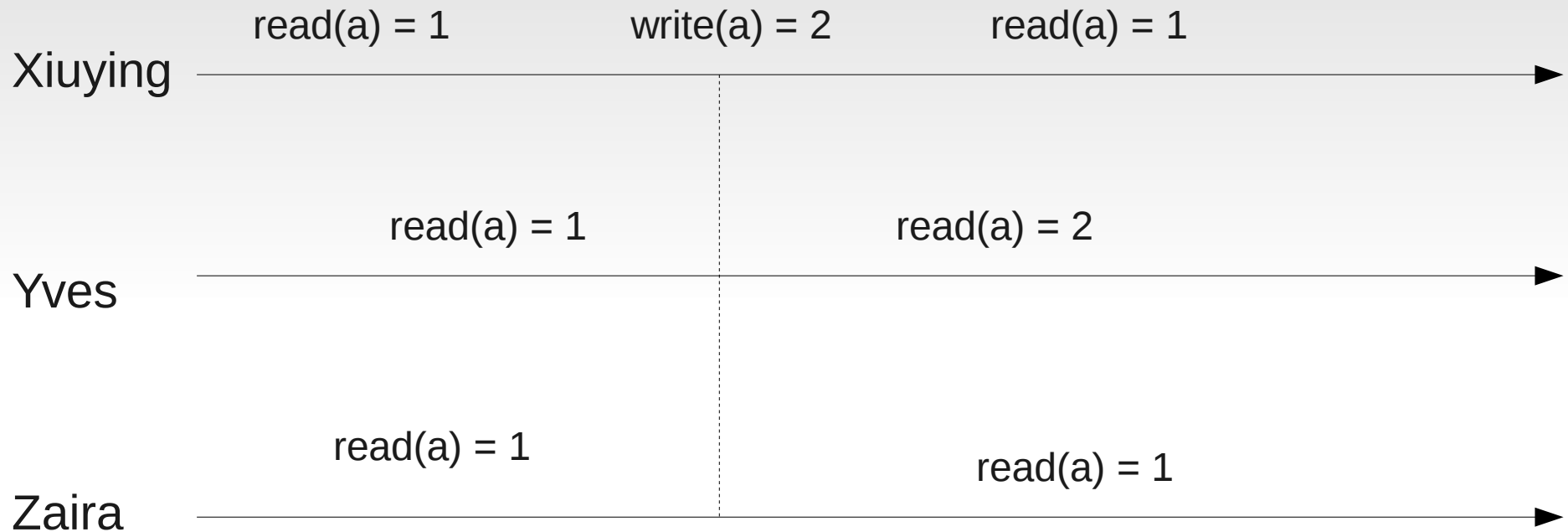
# Strong consistency



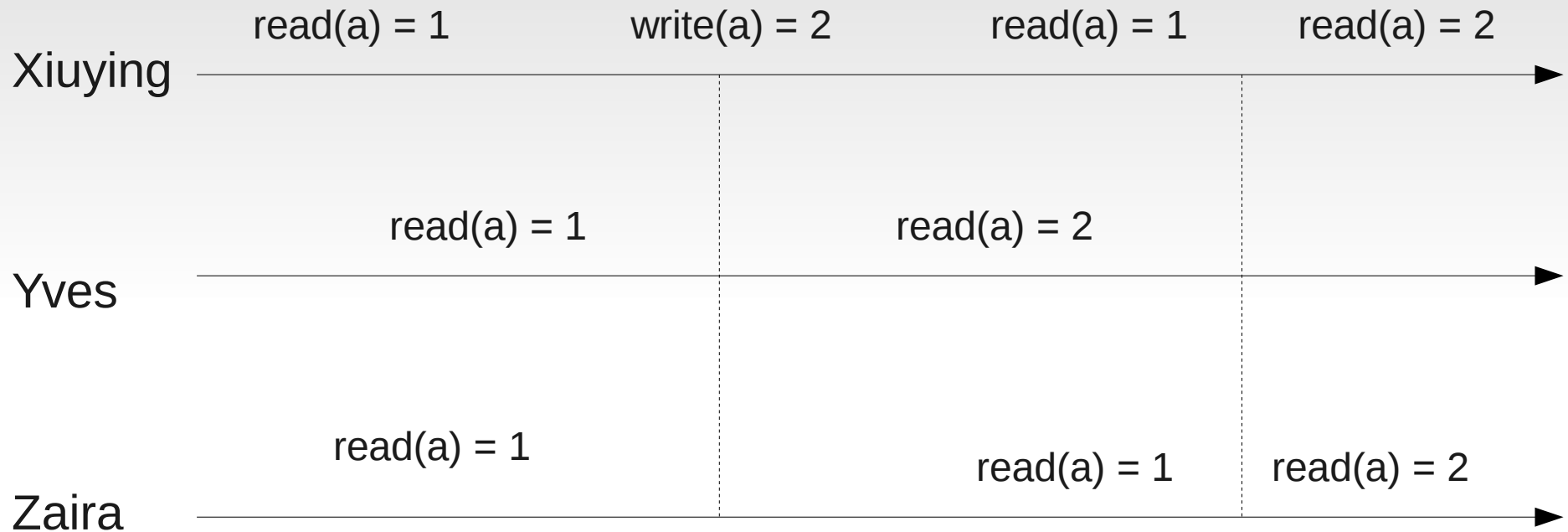
# Eventual consistency



# Eventual consistency



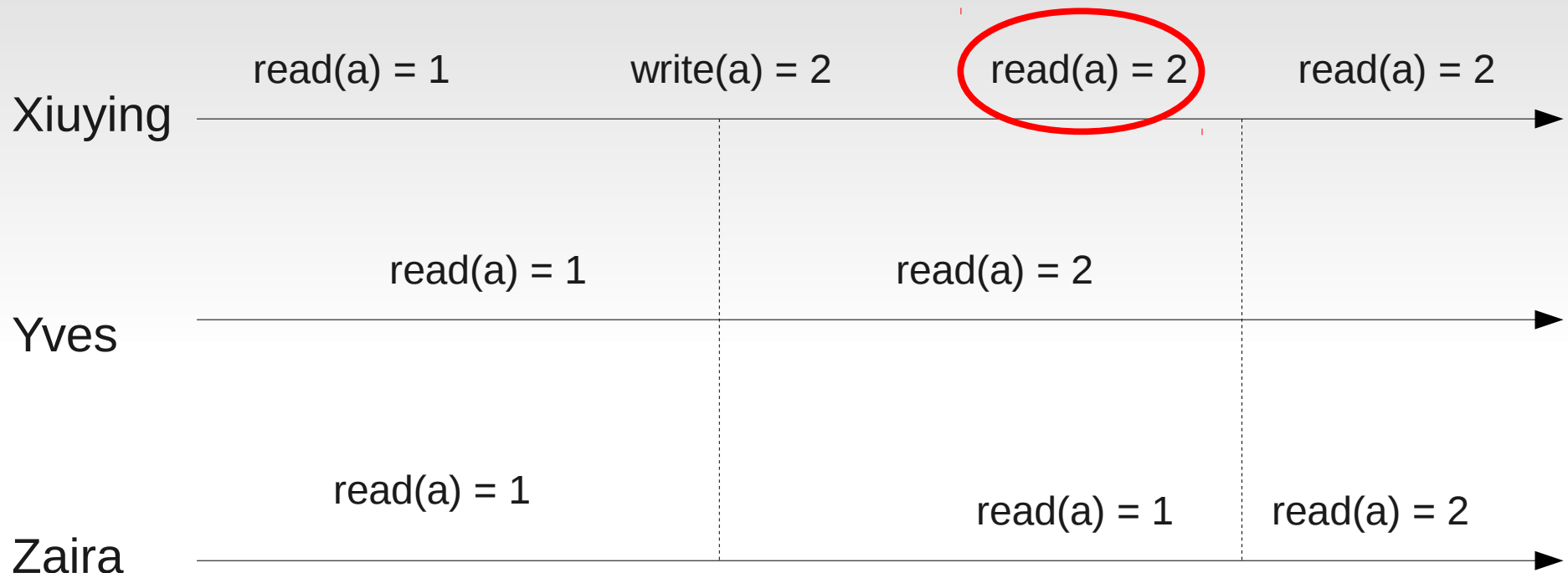
# Eventual consistency



inconsistent window

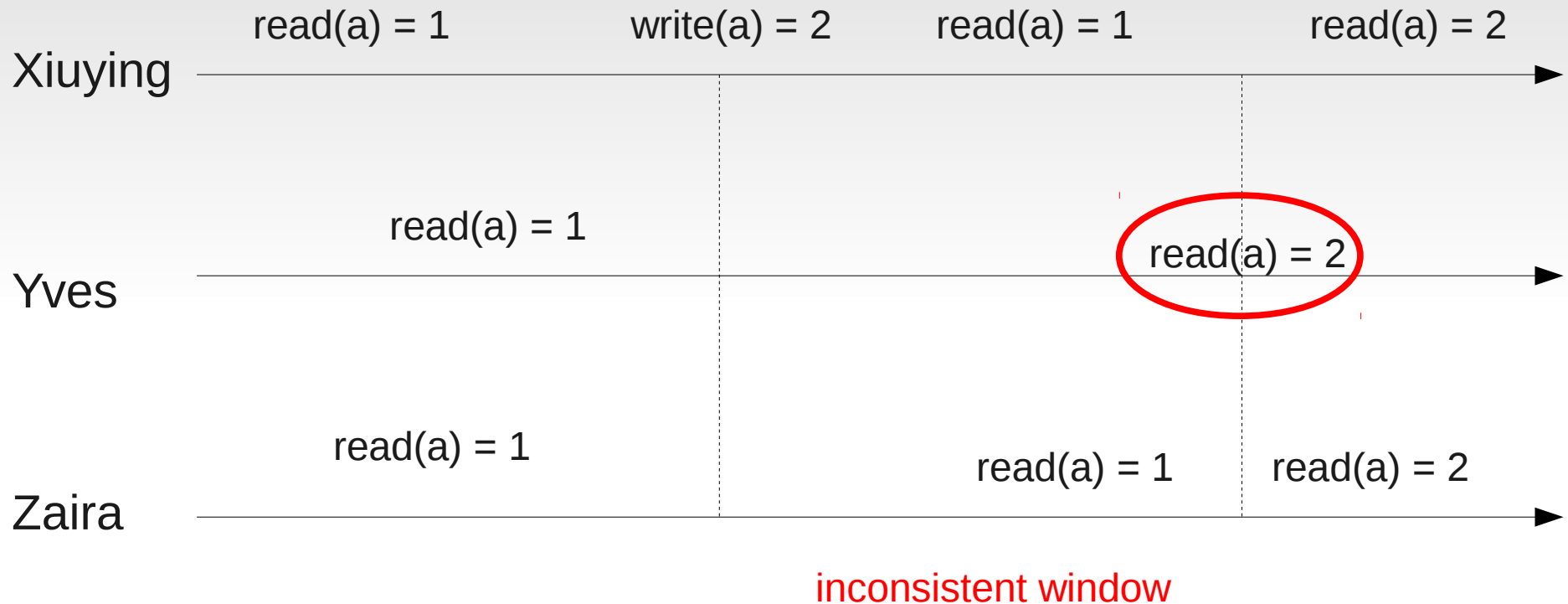


# Read-own writes consistency



inconsistent window

# Monotonic read consistency



# CAP theorem (Brewer)

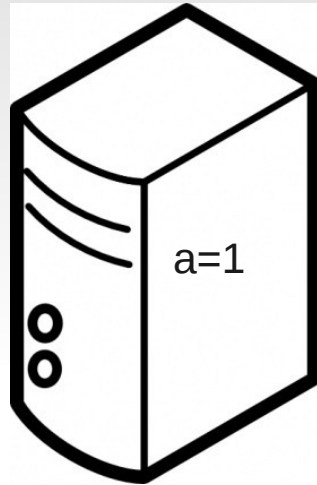
- Consistency
- Availability
- Partitioning

*Pick any two...*

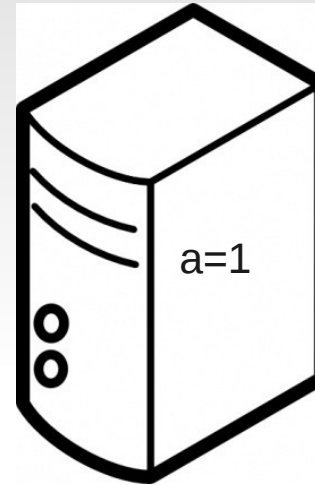
# CAP theorem - informal proof

Ahmed

partition 1



partition 2

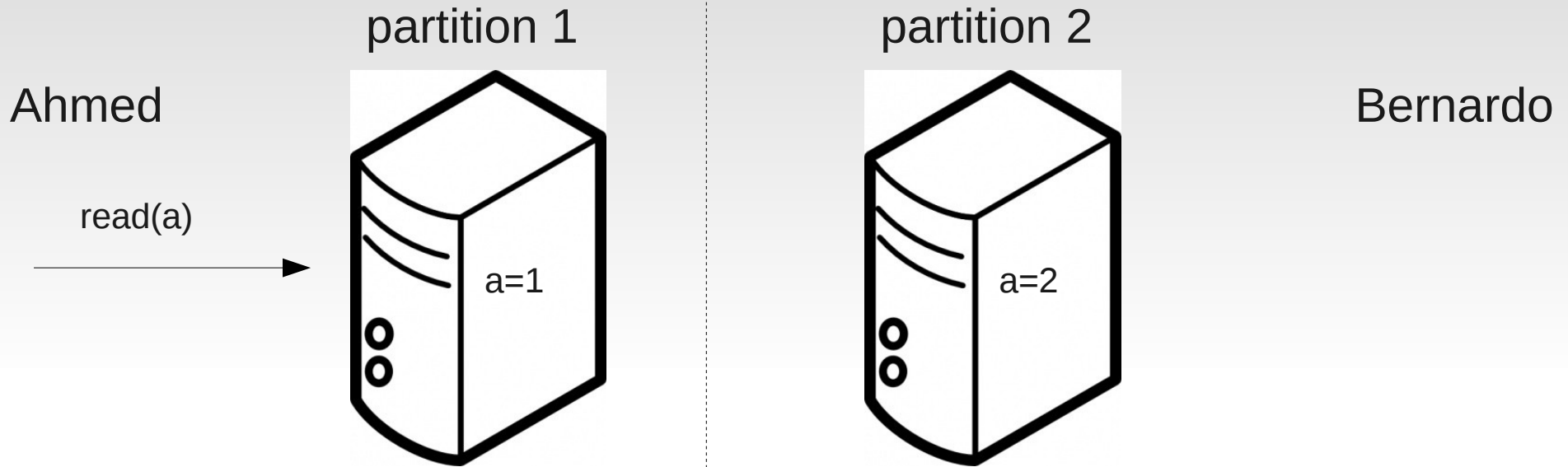


Bernardo

write(a, 2)

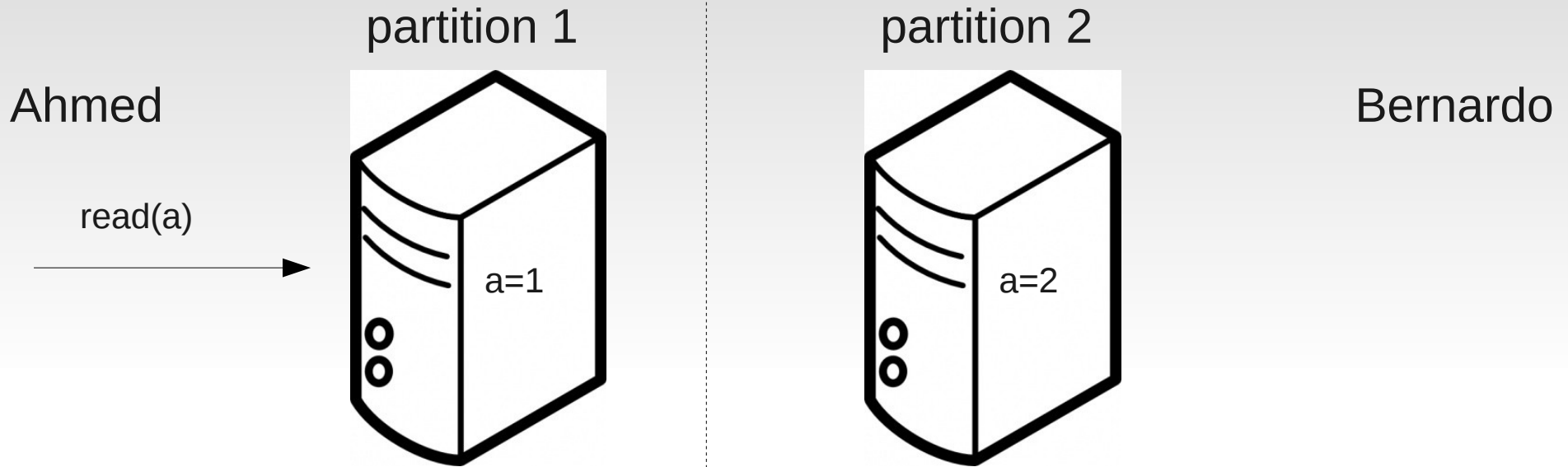


# CAP theorem - informal proof



How should P1 respond to Ahmed?

# CAP theorem - informal proof



How should P1 respond to Ahmed?

- respond that  $a = 1$  (forgo **consistency**)
- delay (forgo **availability**)
- be in constant contact with P2 (forgo **partitioning**)

# Approaches to NoSQL

- key/value
- column-oriented
- document-oriented
- graph-oriented

# Key/value stores

- simplest possible interface:
  - `put('foo', 'bar')`
  - `get('foo')` will return 'bar'
- dates to at least the 1970s (**dbm**)
- **Memcached/MemcacheDB** as a caching layer
- **BerkeleyDB** (now maintained by Oracle)
- others: Tokyo Cabinet, Voldemort, Redis, Scalaris, Google LevelDB



# Column-oriented

RowKey	TimeStamp	ColumnFamily contents	ColumnFamily anchor
com.cnn.www	t1	contents.html = ...	anchor:cnnsi.com = "CNN"
com.cnn.www	t0	contents.html = ...	anchor:cnnsi.com = "News"
...	...	...	...
uk.ac.cam.www	t1	contents.html = ...	anchor:cl.cam.ac.uk = "Home"
uk.ac.cam.cl.www	t1	contents.html = ...	anchor:cl.cam.ac.uk/jcb82 = "My Lab" anchor:cam.ac.uk = "Computer Lab"

Hadoop HBase



# Column-oriented

- maintain unique keys per row
- much more complicated multi-valued columns
  - richer querying possible
- Google's BigTable was a pioneer
  - Bigtable: A Distributed Storage System for Structured Data, Chang et al.
- others: Apache Cassandra, Hadoop HBase, Amazon DynamoDB, Hypertable

# Document-oriented

- like a key-value store, but stores **documents**
- can be *serialised* objects or binary files
  - can query attributes of serialised objects

# XML

```
<person id="11">  
  <name>Joseph Bonneau</name>  
  <email>jcb82@cam</name>  
</person>
```

- can specify separate DTD schema
- can display using XSLT

# JSON

```
{  
  "id": 11,  
  "name": "Joseph Bonneau",  
  "email": "jcb82@cam"  
}
```

- human readable
- native parsing in JavaScript
- BSON-binary version developed for MongoDB

# Protocol buffers

```
message Person {  
  required int32 id = 1;  
  required string name = 2;  
  optional string email = 3;  
}
```

- compiles to binary format-not human-readable
- developed by Google – many similar approaches
  - Apache Thrift

# Document-oriented: MongoDB

```
{
  "_id": ObjectId("4efa8d2b7d284dad101e4bc9"),
  "name": "Joseph Bonneau",
  "role": "PhD student",
  "office number": 17
},
{
  "_id": ObjectId("4efa8d2b7d284dad101e4bc7"),
  "name": "Ross Anderson",
  "role": "Professor",
  "subject": "Security Engineering"
}
```

```
db.find({"name" : "Joseph Bonneau"});
db.find({"name" : { $regex : '.*Anderson$' } } );
db.find({"office number" : { $gt: 16 } } );
```

# Document-oriented

- MongoDB
  - uses BSON (Binary JSON)
  - open-source, commercially developed
- Apache CouchDB
  - uses vanilla JSON
  - open-source, community developed



# Graph-oriented

- developed specifically for large graphs
  - “index-free” lookup of neighbors
- OrientDB is perhaps the best known
  - less practical use than other approaches

# Summary of NoSQL

## Pros:

- scalable and fast
- flexible
- can be easier for experienced programmers

## Cons:

- fewer consistency/concurrency guarantees
- weaker set of queries supported
- less mature

# Big Data stacks

# Big Data stacks

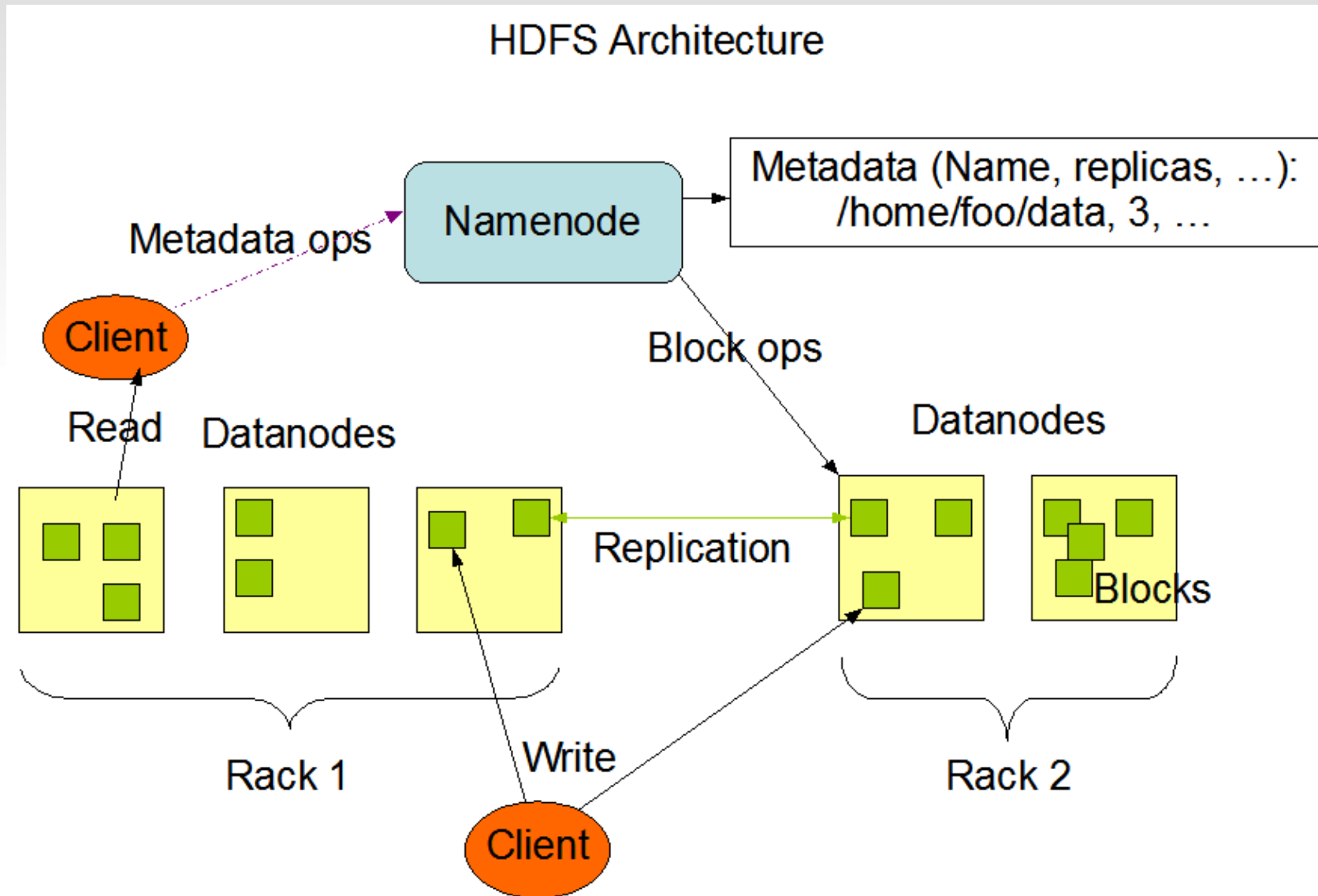
- Google
  - proprietary but influential
- Apache Hadoop
  - totally open-source
- Amazon Web Services (AWS)
  - mixed-source, including some Hadoop

# Hadoop

- founded in 2004 by a Yahoo! employee
- spun into open-source Apache project
- general-purpose framework for Big Data
  - MapReduce implementation
  - supporting tools (distributed storage, concurrency)
- used by everybody...
  - Yahoo!, Facebook, Amazon, Microsoft, Apple

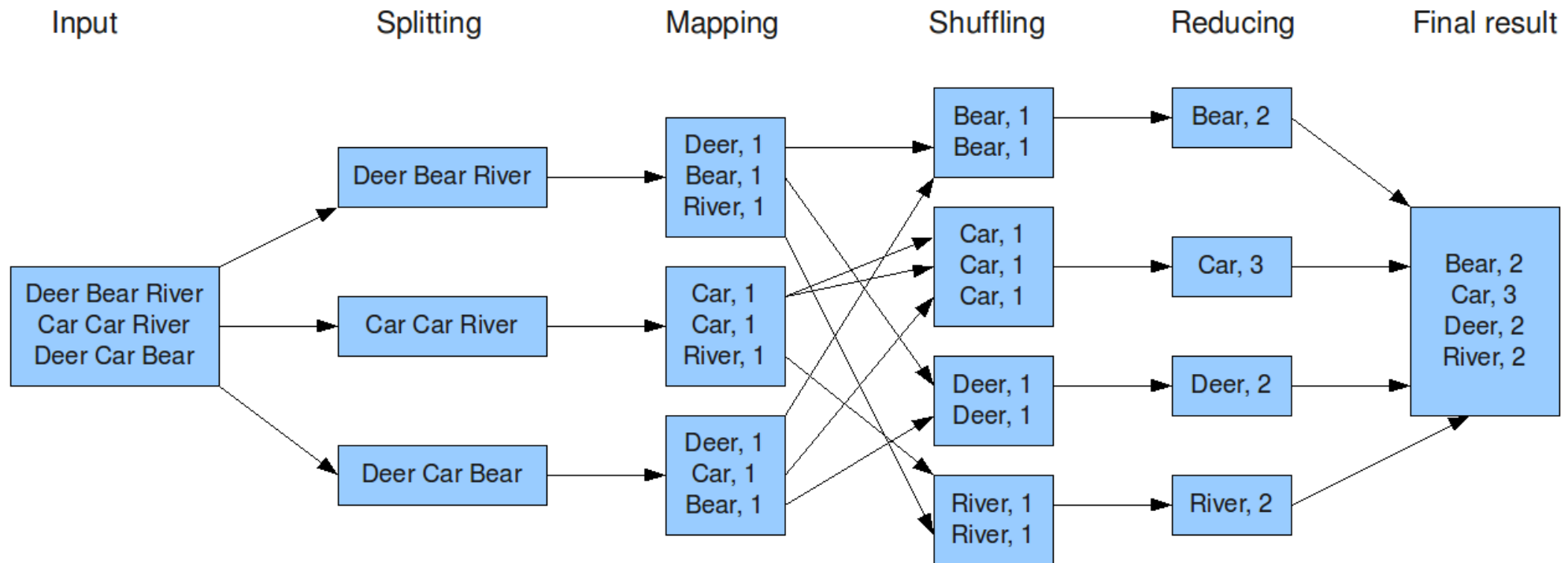


# Hadoop Distributed File Sys.



# Review: MapReduce

The overall MapReduce word count process



# Hadoop MapReduce example

```
public class MapClass extends MapReduceBase implements
Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();
    public void map(LongWritable key, Text value,
                    OutputCollector<Text, IntWritable> output,
                    Reporter reporter) throws IOException {
        String line = value.toString();
        StringTokenizer itr = new StringTokenizer(line);
        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            output.collect(word, one);
        }
    }
}
```



# Hadoop MapReduce example

```
public class Reduce extends MapReduceBase implements
Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text key, Iterator<IntWritable> values,
                        OutputCollector<Text, IntWritable>
output,
                        Reporter reporter) throws IOException {
        int sum = 0;
        while (values.hasNext()) {
            sum += values.next().get();
        }
        output.collect(key, new IntWritable(sum));
    }
}
```

# Hadoop MapReduce example

```
public class WordCount {  
    public static void main(String[] args) throws Exception {  
        JobConf conf = new JobConf(WordCount.class);  
        conf.setJobName("wordcount");  
  
        conf.setOutputKeyClass(Text.class);  
        conf.setOutputValueClass(IntWritable.class);  
  
        conf.setMapperClass(Map.class);  
        conf.setCombinerClass(Reduce.class);  
        conf.setReducerClass(Reduce.class);  
  
        conf.setInputFormat(TextInputFormat.class);  
        conf.setOutputFormat(TextOutputFormat.class);  
  
        FileInputFormat.setInputPaths(conf, new Path(args[0]));  
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));  
        JobClient.runJob(conf);  
    }  
}
```

# Hadoop JobTracker

## 10 Hadoop Map/Reduce Administration

[Quick Links](#)

**State:** RUNNING

**Started:** Tue May 10 17:11:39 GMT 2011

**Version:** 0.21.0, 985326

**Compiled:** Tue Aug 17 01:02:28 EDT 2010 by tomwhite from branches/branch-0.21

**Identifier:** 201105101711

### Cluster Summary (Heap Size is 106.12 MB/1.56 GB)

Queues	Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Slot Capacity	Reduce Slot Capacity	Avg. Slots/Node	Blacklisted Nodes	Excluded Nodes
<a href="#">1</a>	6	1	1	<a href="#">4</a>	6	1	0	0	8	8	4.00	<a href="#">0</a>	<a href="#">0</a>

**Filter (Jobid, Priority, User, Name)**

Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

### Running Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
<a href="#">job_201105101711_0001</a>	NORMAL	hadoop	QuasiMonteCarlo	50.00% <div></div>	20	10	0.00% <div></div>	1	0	NA

### Retired Jobs

*none*

### Local Logs

[Log](#) directory, [Job Tracker History](#)

[Hadoop](#), 2011.

# MapReduce alternative: Pig

```
lines = LOAD '../data/words.txt' USING TextLoader() AS  
(sentence:chararray);  
  
words = FOREACH lines GENERATE FLATTEN(TOKENIZE(sentence)) AS  
word;  
  
groupedWords = GROUP words BY word;  
  
counts = FOREACH groupedWords GENERATE group, COUNT(words);  
  
STORE counts INTO 'output/wordcounts' USING PigStorage();
```



# MapReduce alternative: Hive

```
create table textlines(text string);
```

```
load data local inpath  
'C:\work\ClearPoint\Data20\data\words.txt' overwrite into table  
textlines;
```

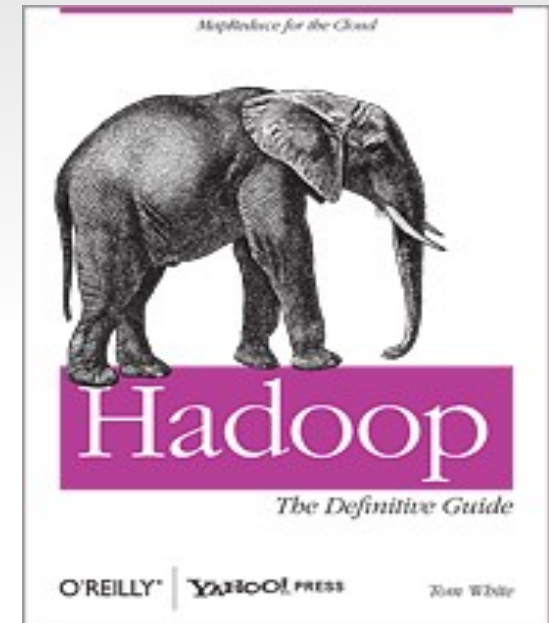
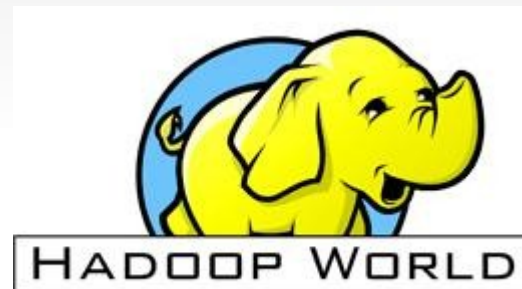
```
create table words(word string);
```

```
insert overwrite table words select explode(split(text, '[ \t]  
+')) word from textlines;
```

```
select word, count(*) from words group by word;
```



# The Hadoop Industry



# Amazon web services



- launched 2006
- largest, most popular cloud computing platform
  - others: Rackspace, Azure, Google App Engine

# Elastic Compute Cloud (EC2)

- rent “Elastic compute units” by the hour
  - one ECU = one 1 GHz processor machine
- can choose Linux, FreeBSD, Solaris, Windows
- virtual private servers running on Xen
- pricing: US\$0.02-2.50 per hour
  - varies by machine capacity
  - spot pricing-varies by demand



# Simple Storage Service (S3)

- store arbitrary files (objects)
- index by “bucket” and “key”
  - `http://s3.amazonaws.com/bucket/key`
- accessible via HTTP, SOAP, BitTorrent
- directly readable from EC2 machines
- over 1 trillion objects now uploaded
- pricing: US\$0.05-0.10 per GB per month
  - similar rates for transfer out, free transfer in

# Other AWS elements

- elastic MapReduce
  - run Hadoop on EC2 machines with S3 storage
  - free data transfer
- relational Database Service
  - SQL database
- many features for running a data-driven website
  - content delivery networks, caches, etc.

# Comparison

	Google	Hadoop	Amazon WS
storage	GoogleFS	HDFS	S3
caching	memcacheg		ElastiCache
locking	chubby	Zookeeper	
key-value	LevelDB		
column-oriented	BigTable	Cassandra	DynamoDB
document-oriented		CouchDB	SimpleDB

# Visualisation

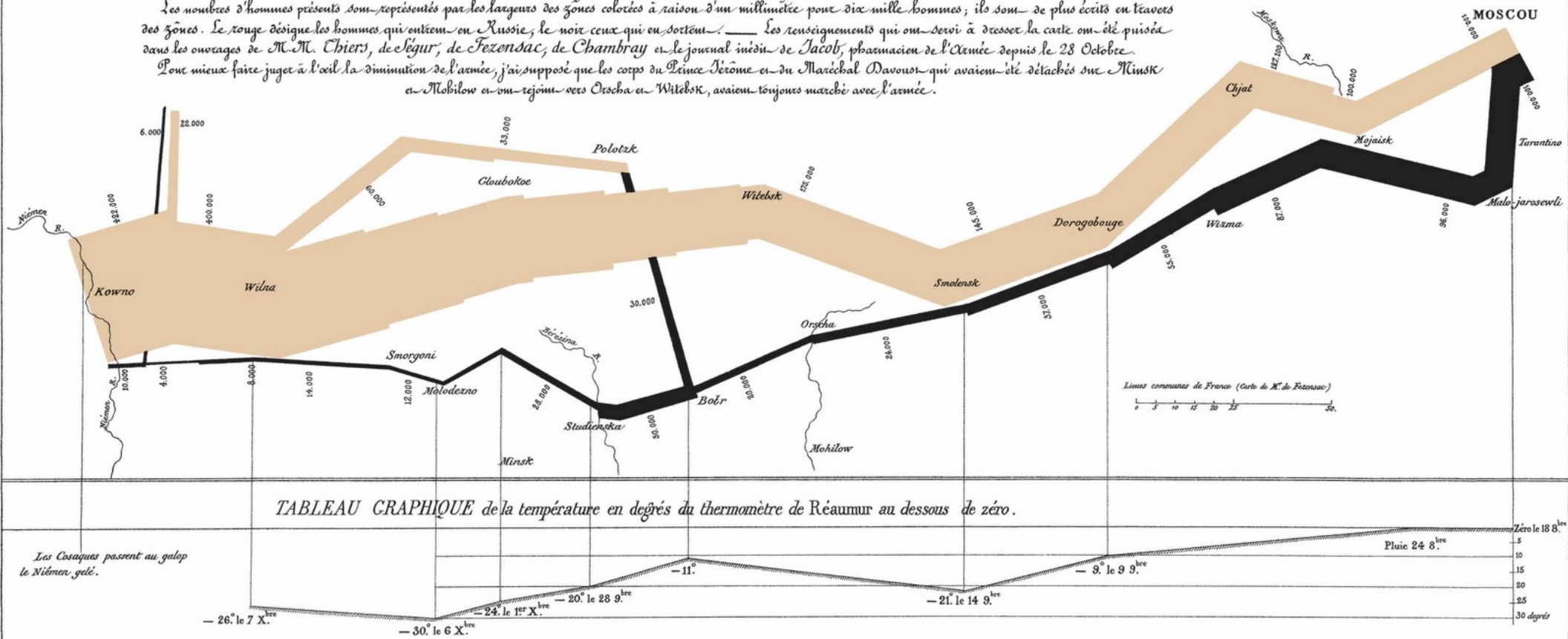
# Visualisation

## Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813.

Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite Paris, le 20 Novembre 1869.

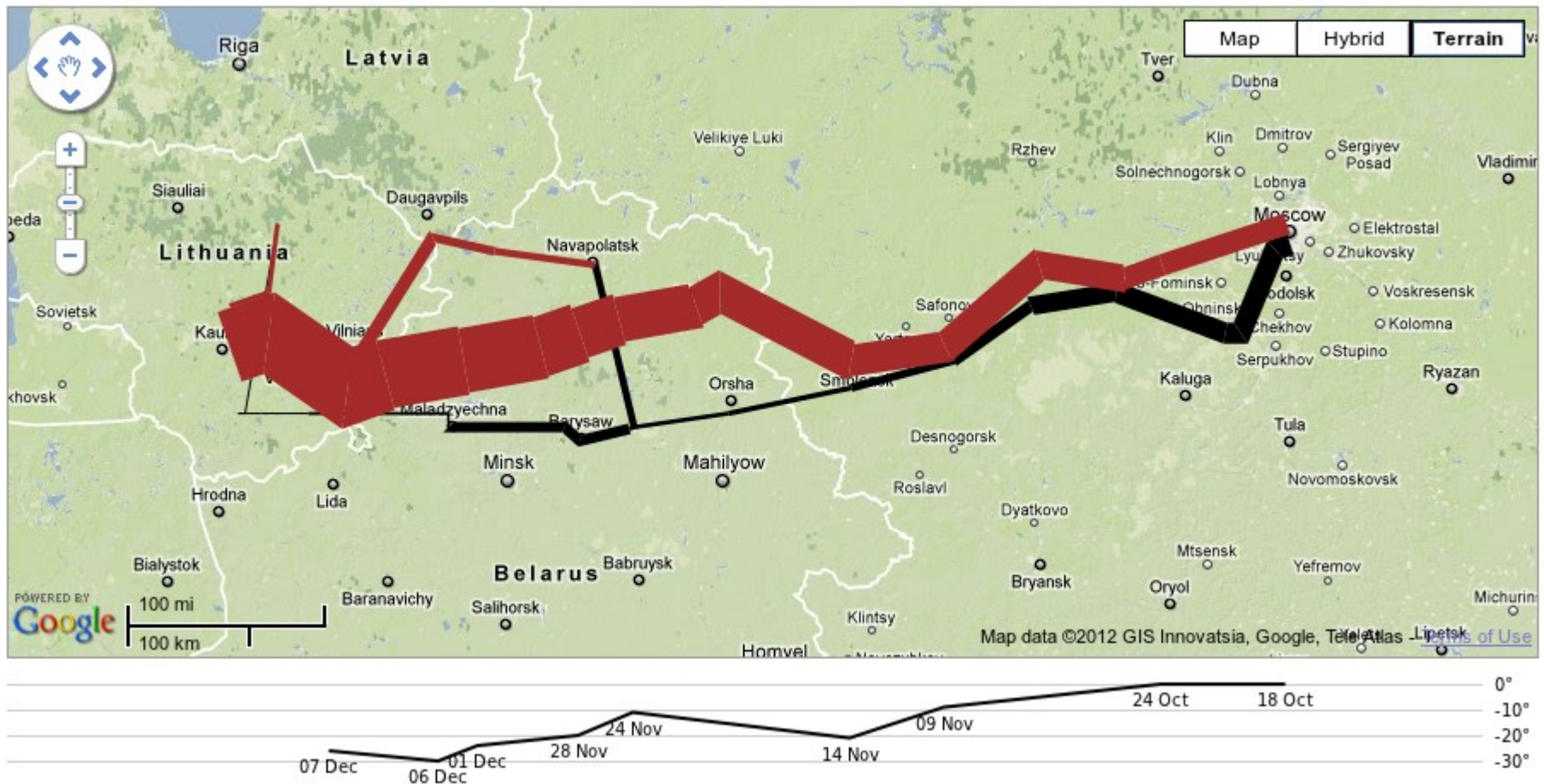
Les nombres d'hommes présents sont représentés par les largeurs des zones colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zones. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M. M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.

Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davout qui avaient été détachés sur Minsk et Mohilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.



# Visualisation

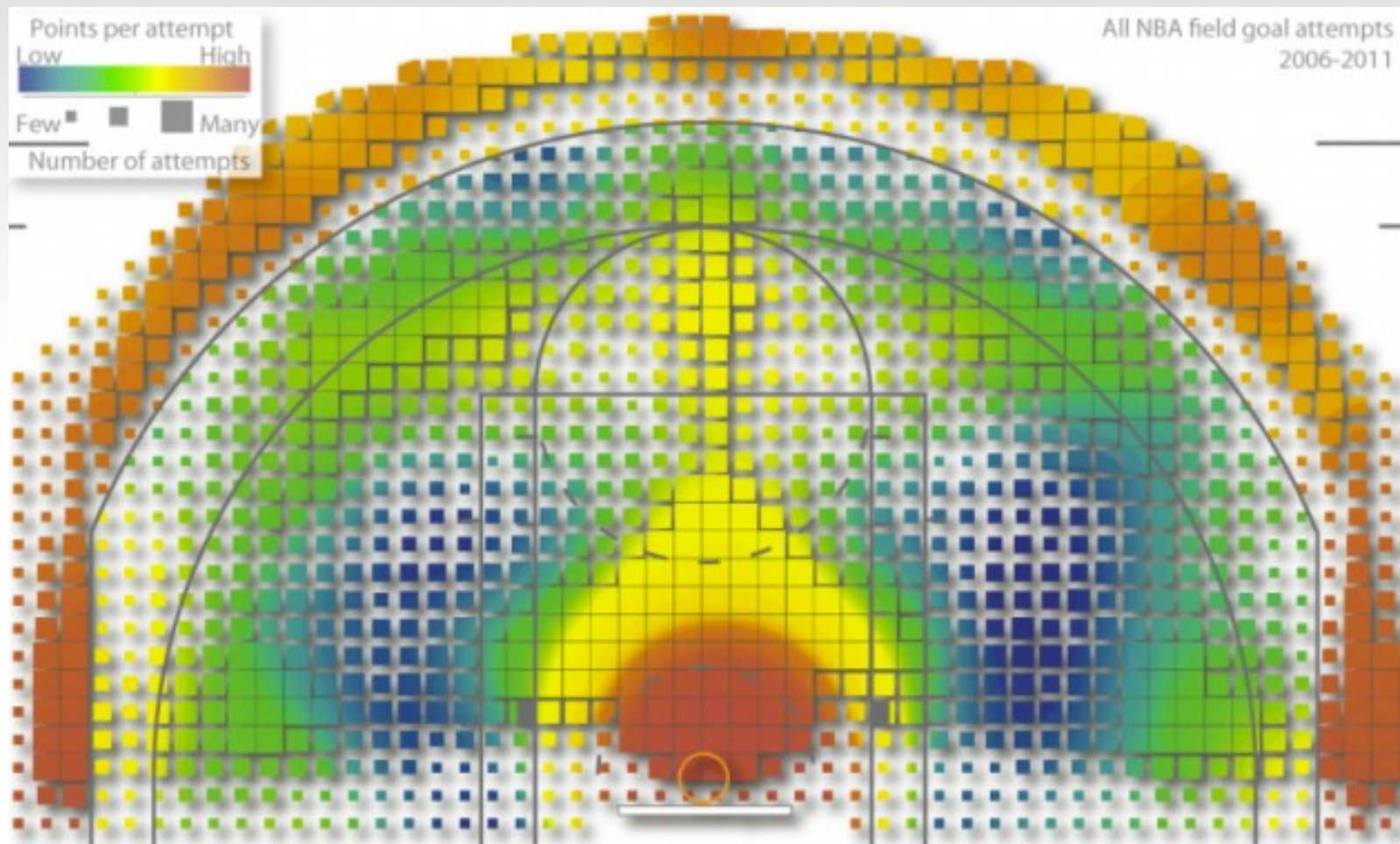
## Minard's Napoleon



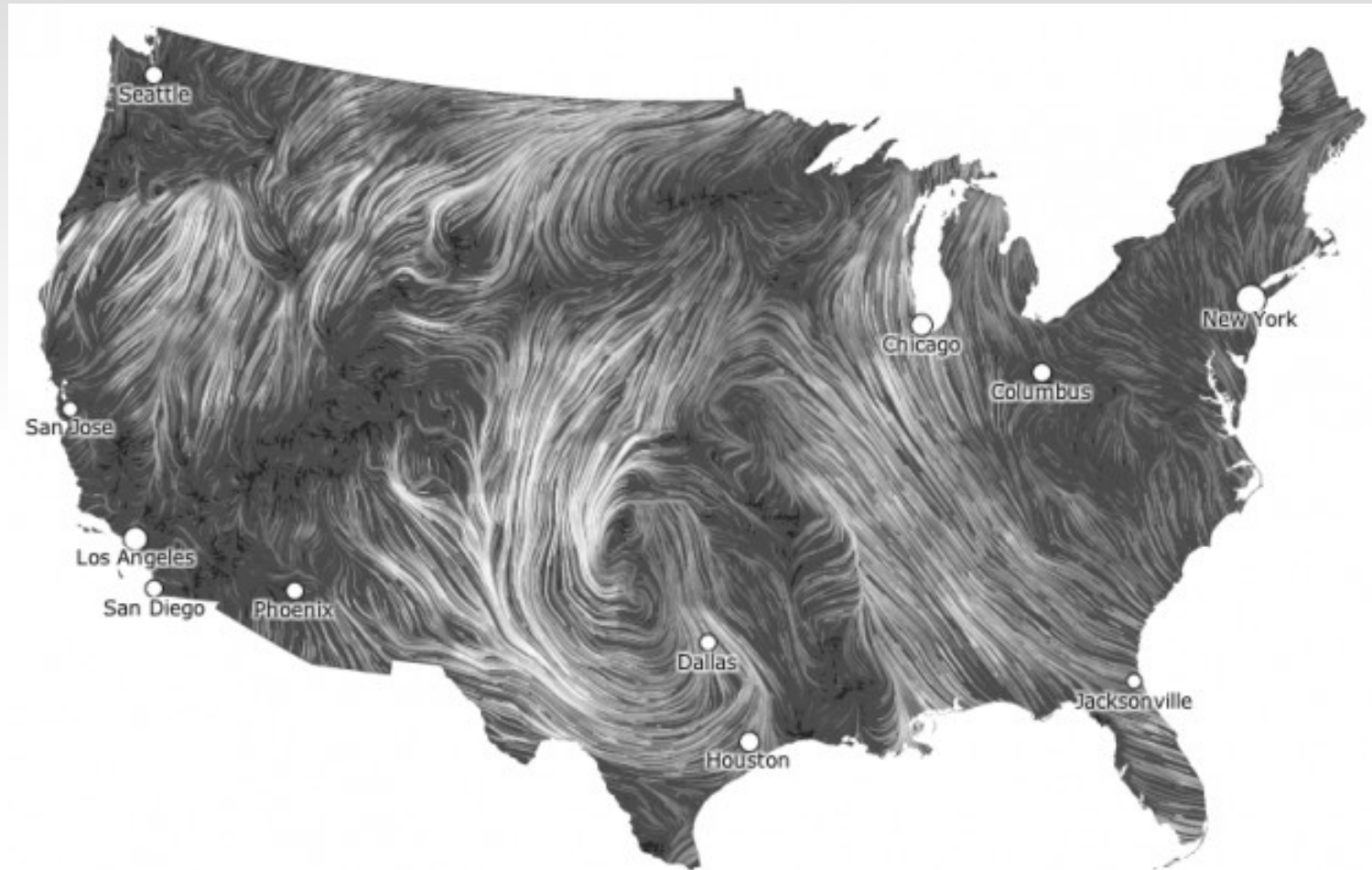
re-creation with Protovis, Google Maps



# Visualisation



# Visualisation



<http://hint.fm/wind/>



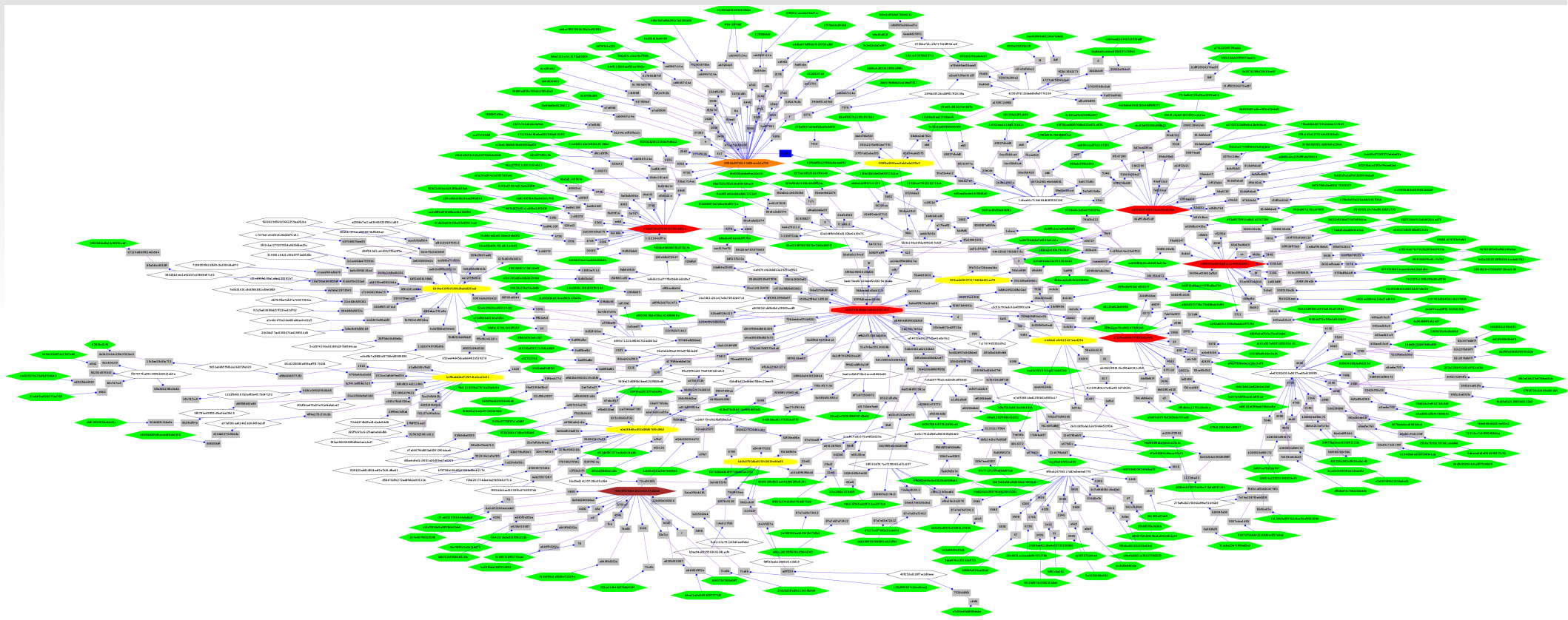
# Visualisation



# Visualisation challenges

- 2 spacial dimensions + 3 colour dimensions
- big data sets can have thousands!
- animation/interactivity often necessary

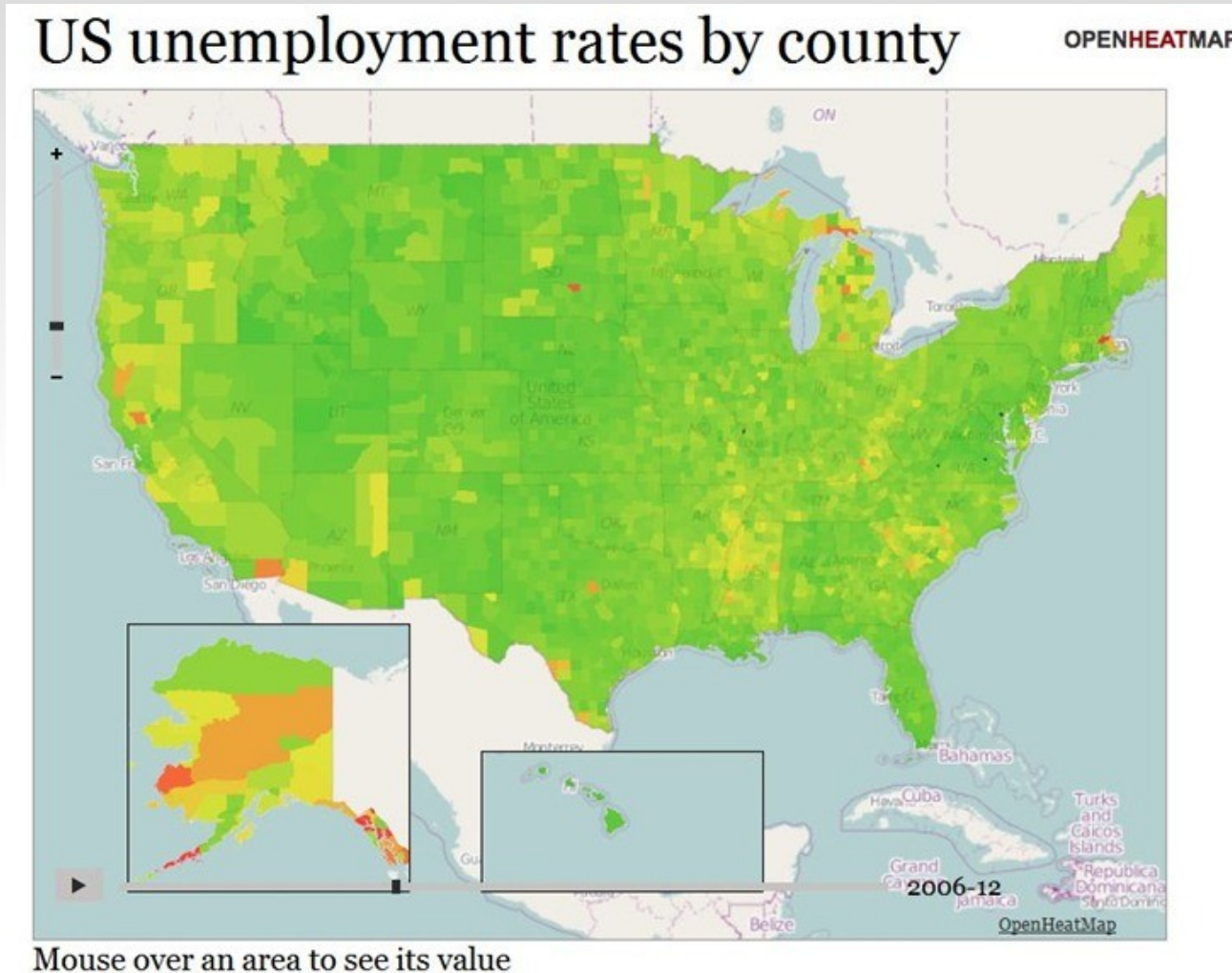
# Graph visualisation tools



GraphViz

See also: Protovis, Gephi, Processing,

# Geographic visualisation tools

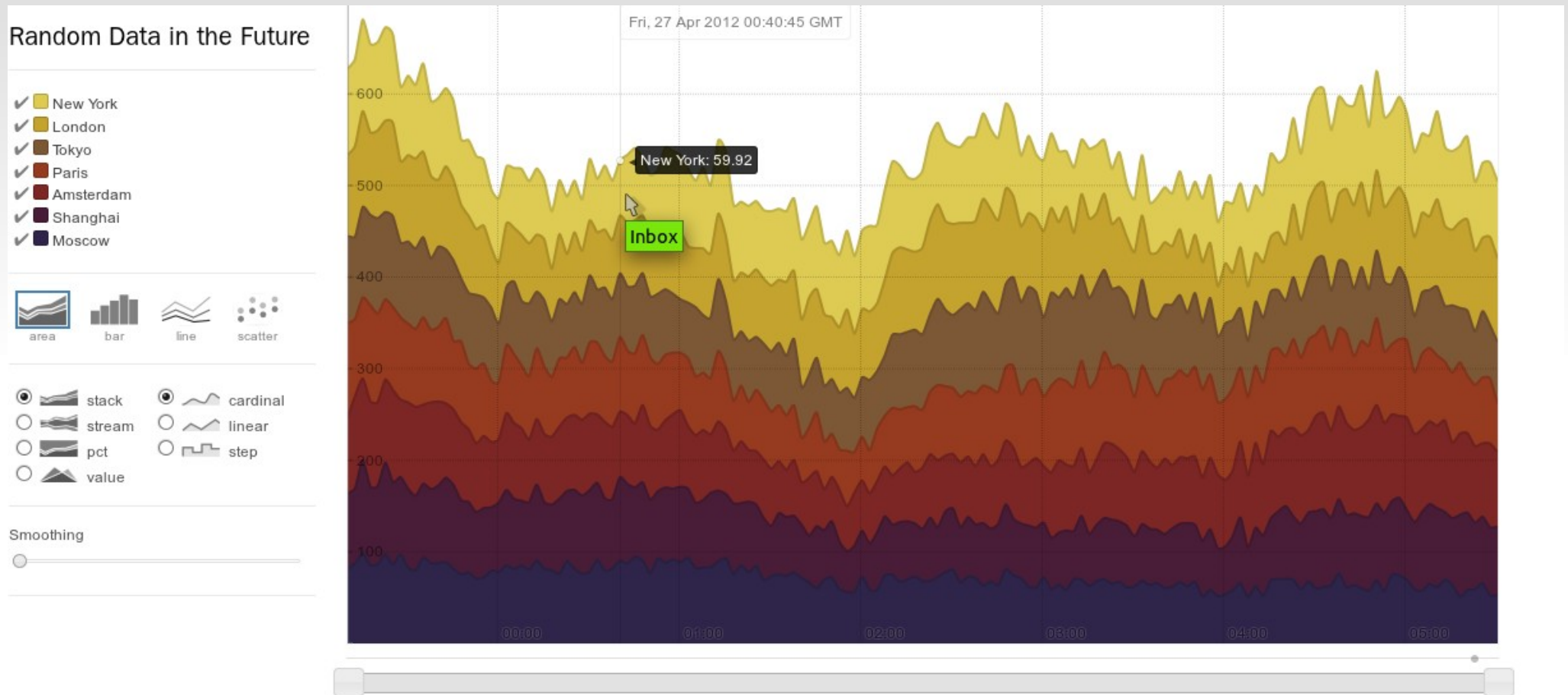


OpenHeatMap

See also: FusionTables, GoogleMaps API



# Interactive chart tools



Rickshaw

See also: Tableau, Highcharts JS, ExtJS, Raphaël, flot, dojox.charting

# Privacy

# Privacy

*Technology is neither good nor bad, it is neutral*

- big data is often generated by people
- obtaining consent is often impossible
- anonymisation is very hard...

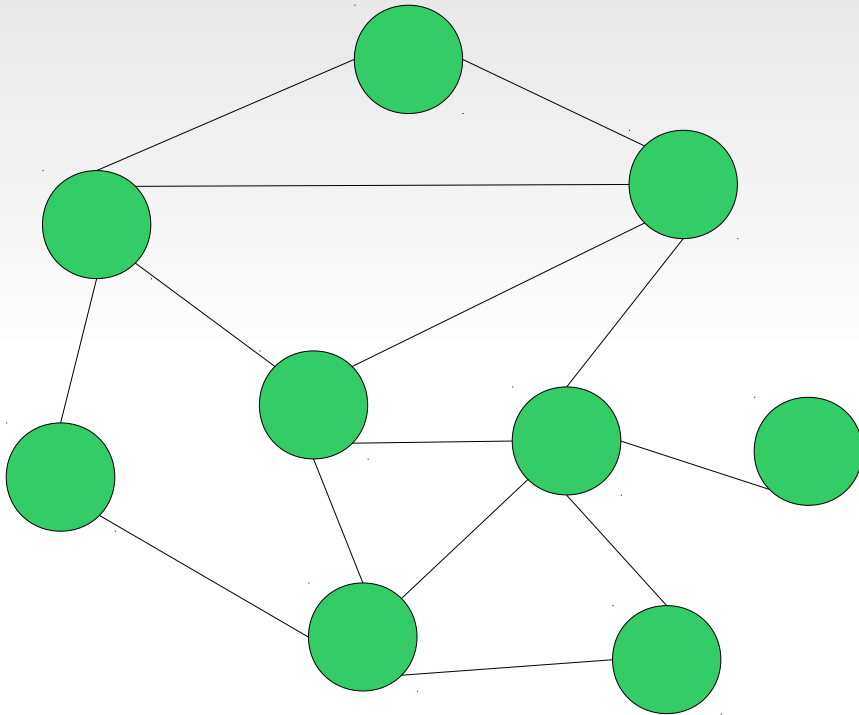
# You only need 33 bits...

- birth date, postcode, gender
  - Unique for 87% of US population (Sweeney 1997)
- preference in movies
  - 99% of 500k with 8 ratings (Narayanan 2007)
- web browser
  - 94% of 500k users (Eckersley 2010)
- writing style
  - 20% accurate out of 100k users (Narayanan 2012)

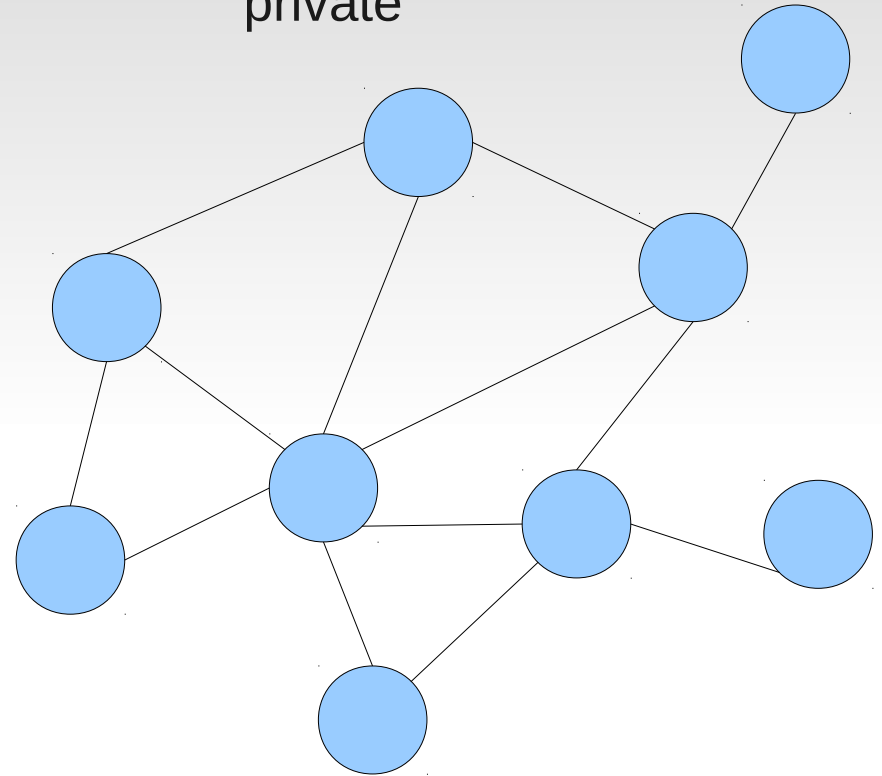


# Cross-graph de-anonymisation

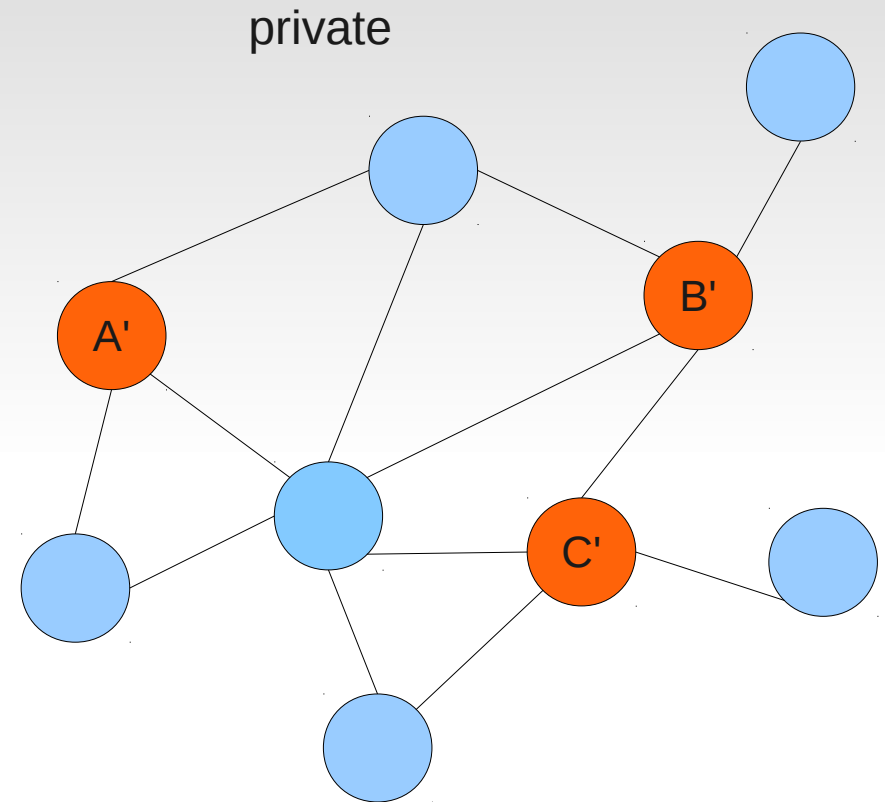
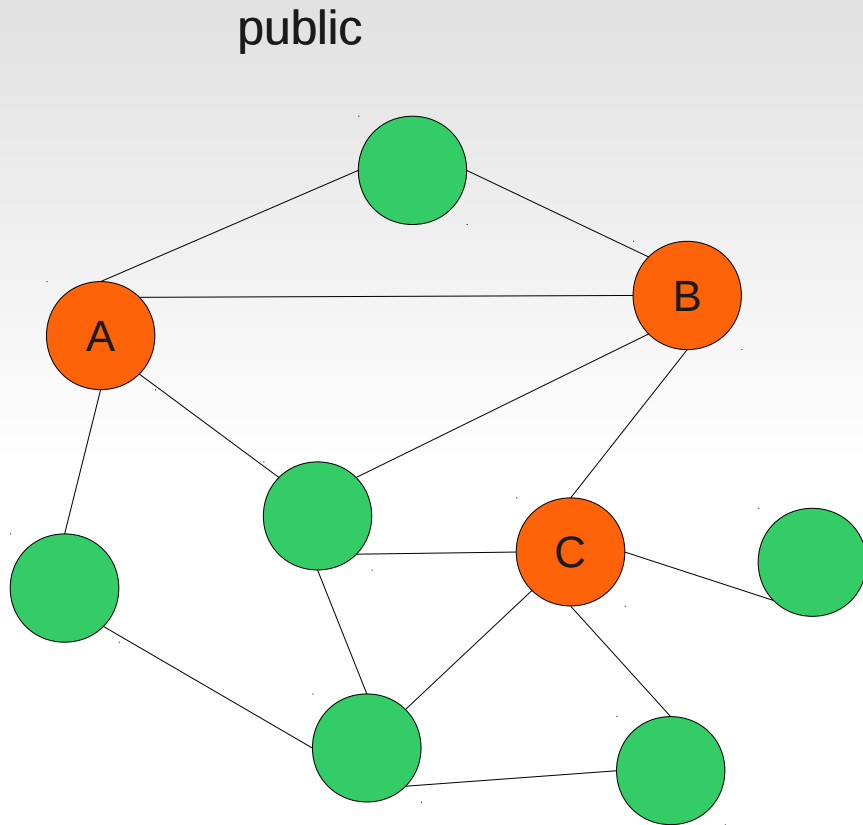
public



private

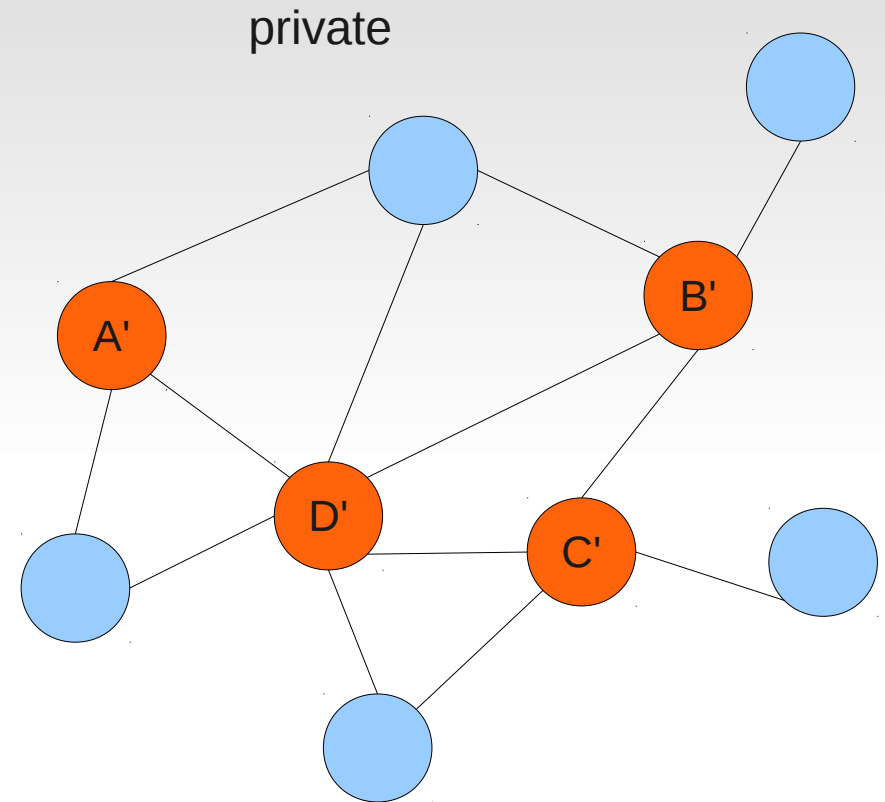
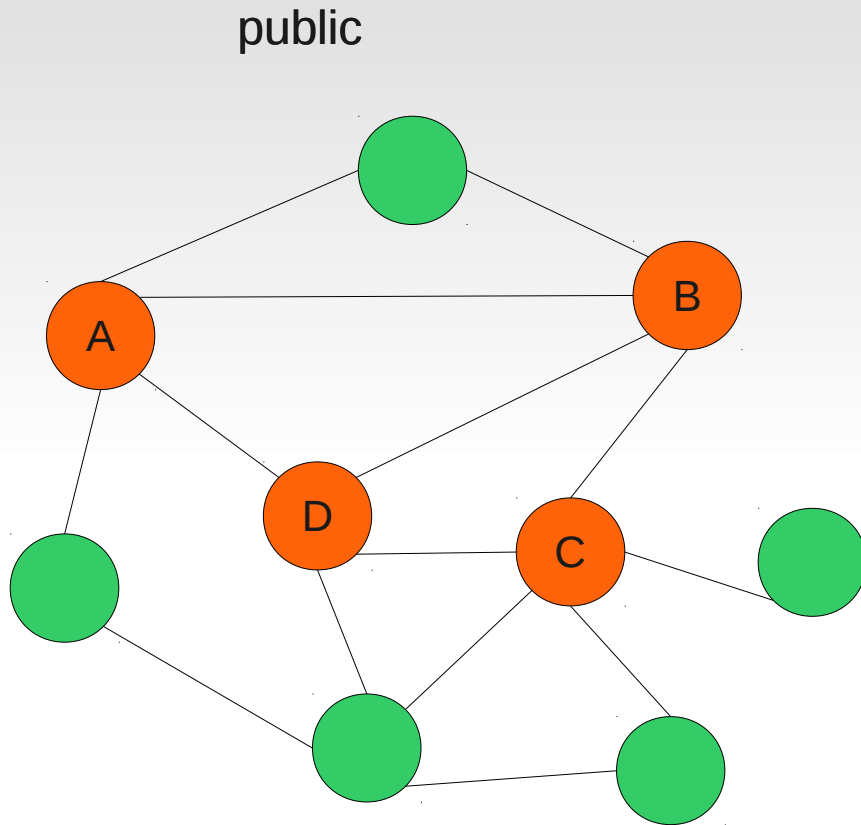


# Cross-graph de-anonymisation



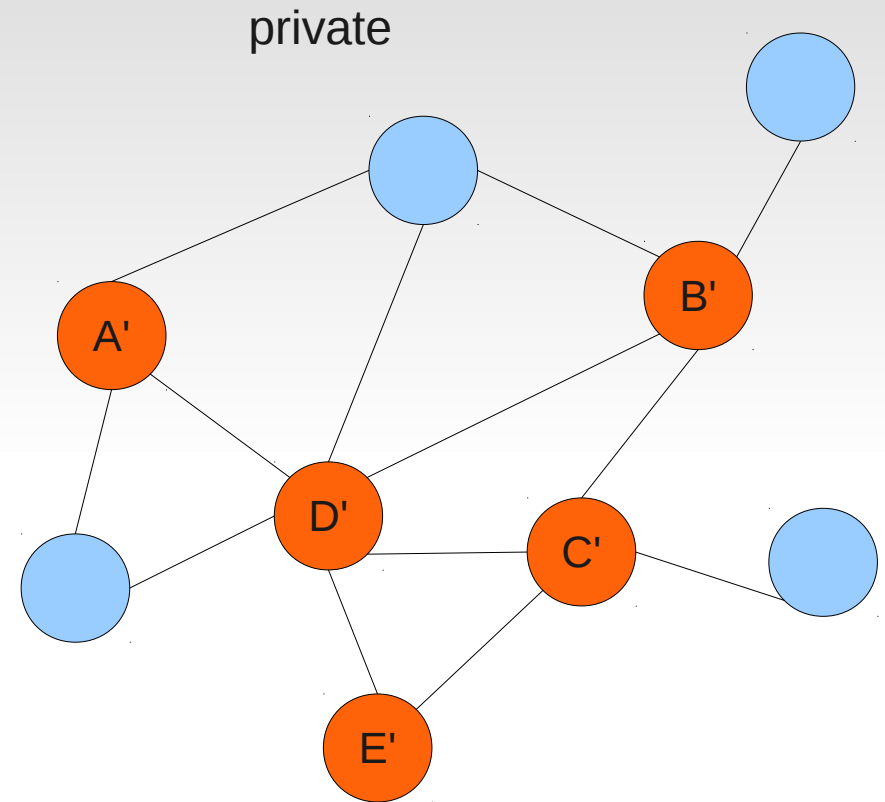
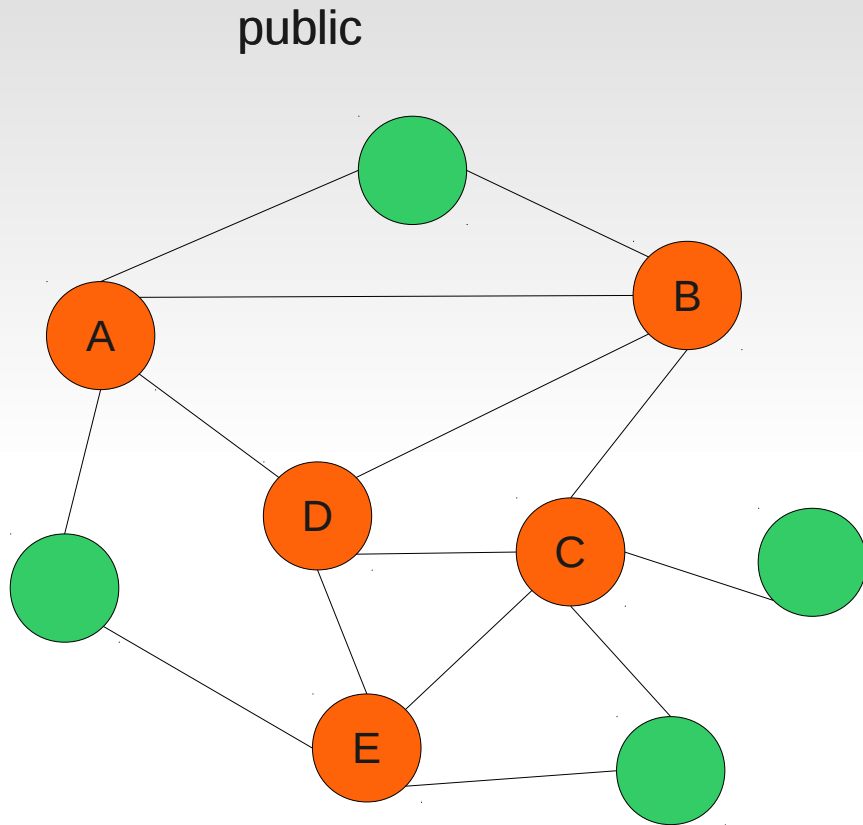
Step 1: identify seed nodes

# Cross-graph de-anonymisation



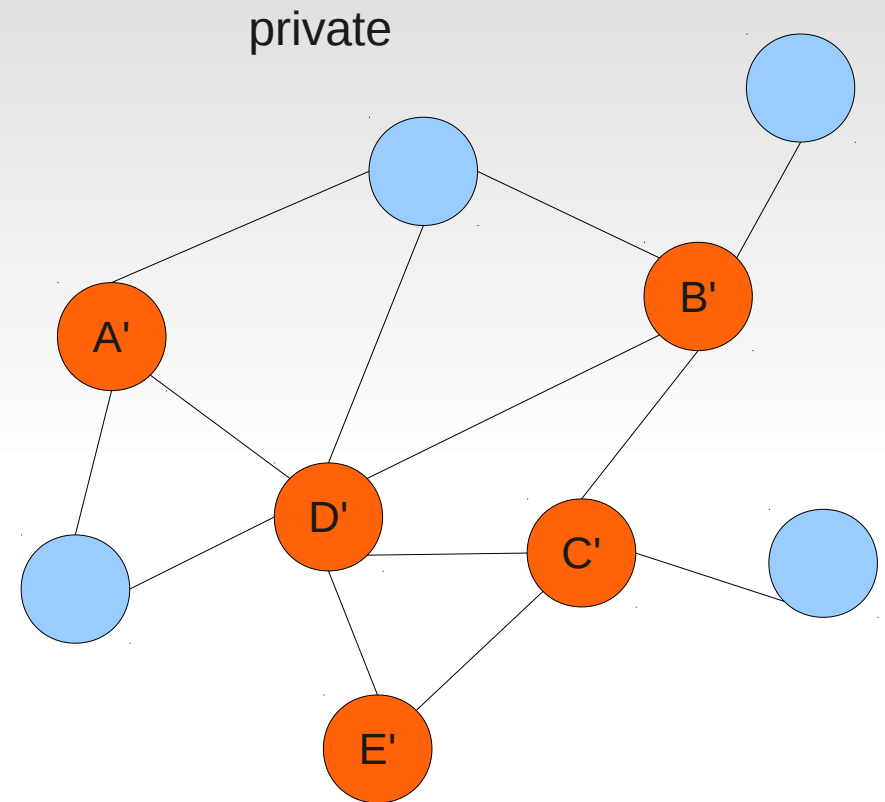
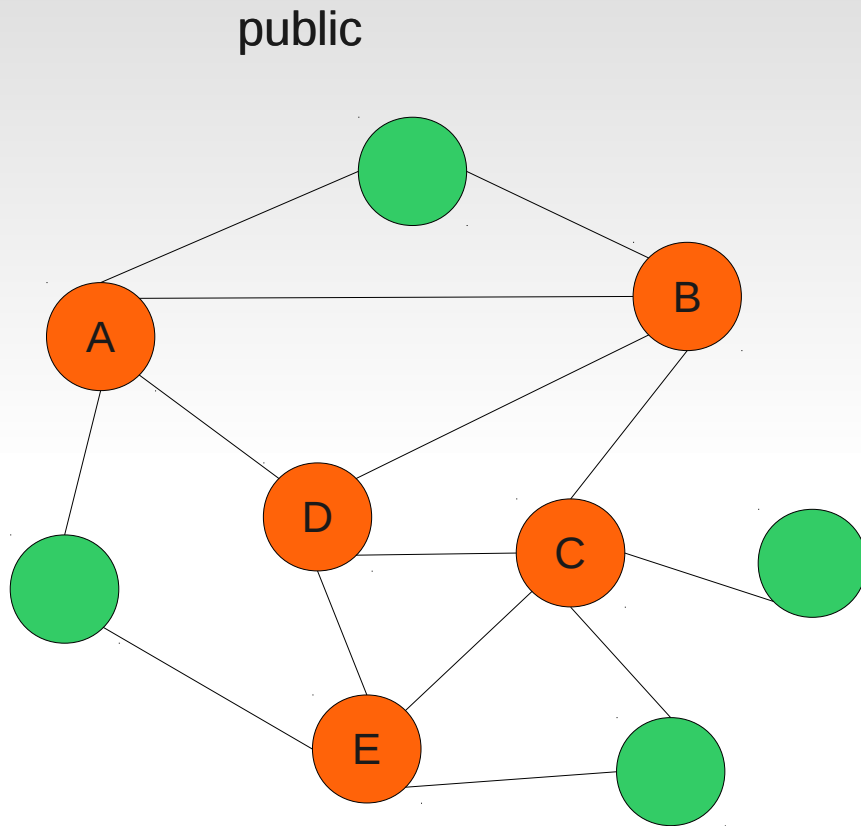
Step 2: assign mappings based on mapped neighbors

# Cross-graph de-anonymisation



Step 3: iterate

# Cross-graph de-anonymisation



Step 3: Iterate

**31%** of common Twitter/Flickr users found given **30** seeds!

# Homophily

*Birds of a feather link together*

- age
- gender
- political orientation
- sexual orientation

**Thank you**