FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

SEMINAR 1 - BIG DATA AND ITS ETHICAL IMPLICATIONS

SHOULD DATA RULE TOMORROW'S WORLD?

UNIVERSITY OF PASSAU | 22-26 FEBRUARY 2021 | BIG DATA

Working on Big Data Datasets

Dr.ing. Cătălin Negru
catalin.negru@cs.pub.ro
Prof.dr.ing. Florin Pop
florin.pop@cs.pub.ro

FIT EUROPE

FUTURE IT LEADERS FOR A MULTICULTURAL, DIGITAL EUROPE

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

CONTENTS

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

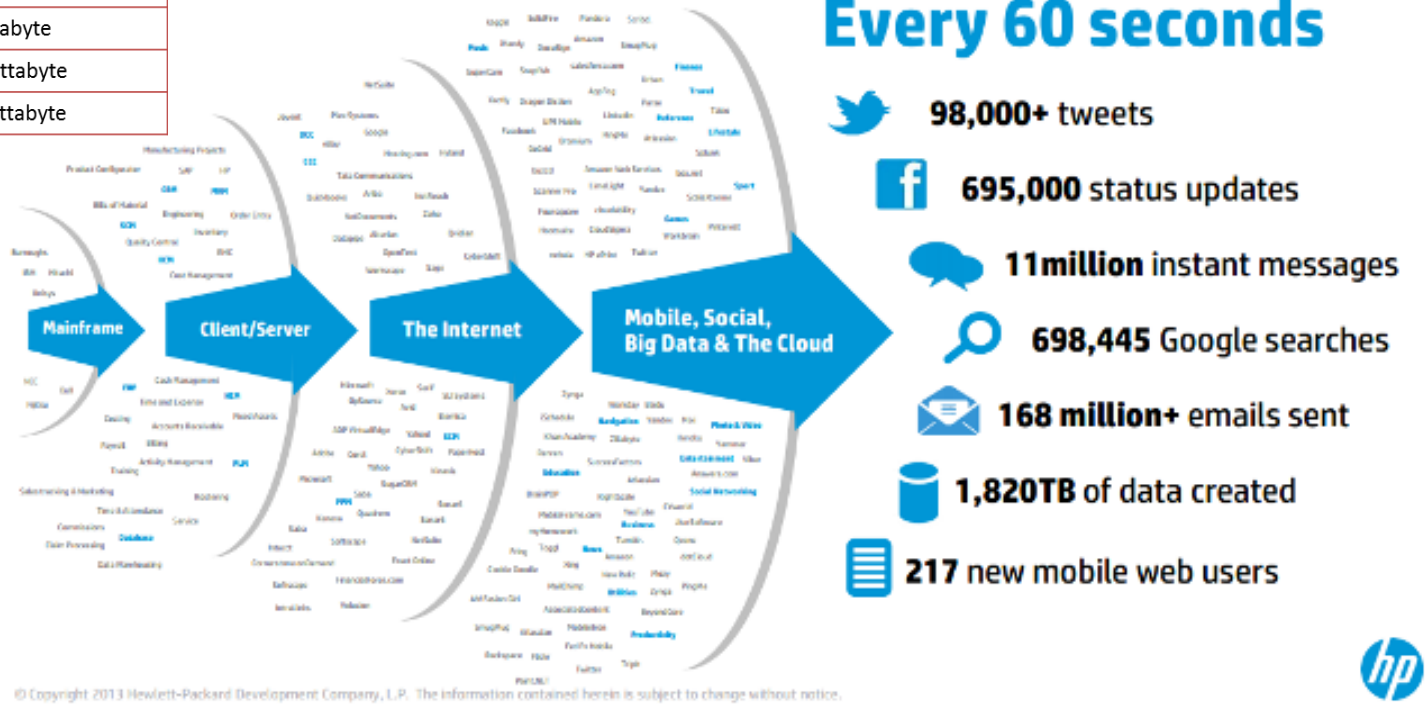
The Human Face of Big Data

- Full movie: https://www.youtube.com/watch?v=3G9lrjcqXbk
- Big Data will impact every part of your life: https://www.youtube.com/watch?v=0Q3sRSUYmys
- What Exactly Is Big Data? https://www.forbes.com/video/4857597029001

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Understanding Big Datasets

Value	Symbol	Name
1024	KB	Kilobyte
1024^2	MB	Megabyte
1024^3	GB	Gigabyte
1024^4	TB	Terabyte
1024^5	PB	Petabyte
1024^6	EB	Exabyte
1024^7	ZB	Zettabyte
1024^8	YB	Yottabyte



2.5 quintillion bytes of data each day

Dataset whose volume, velocity, variety and complexity are beyond the ability of commonly used tools to capture, process, store, manage and analyze them can be termed as ***BIG DATA****.*

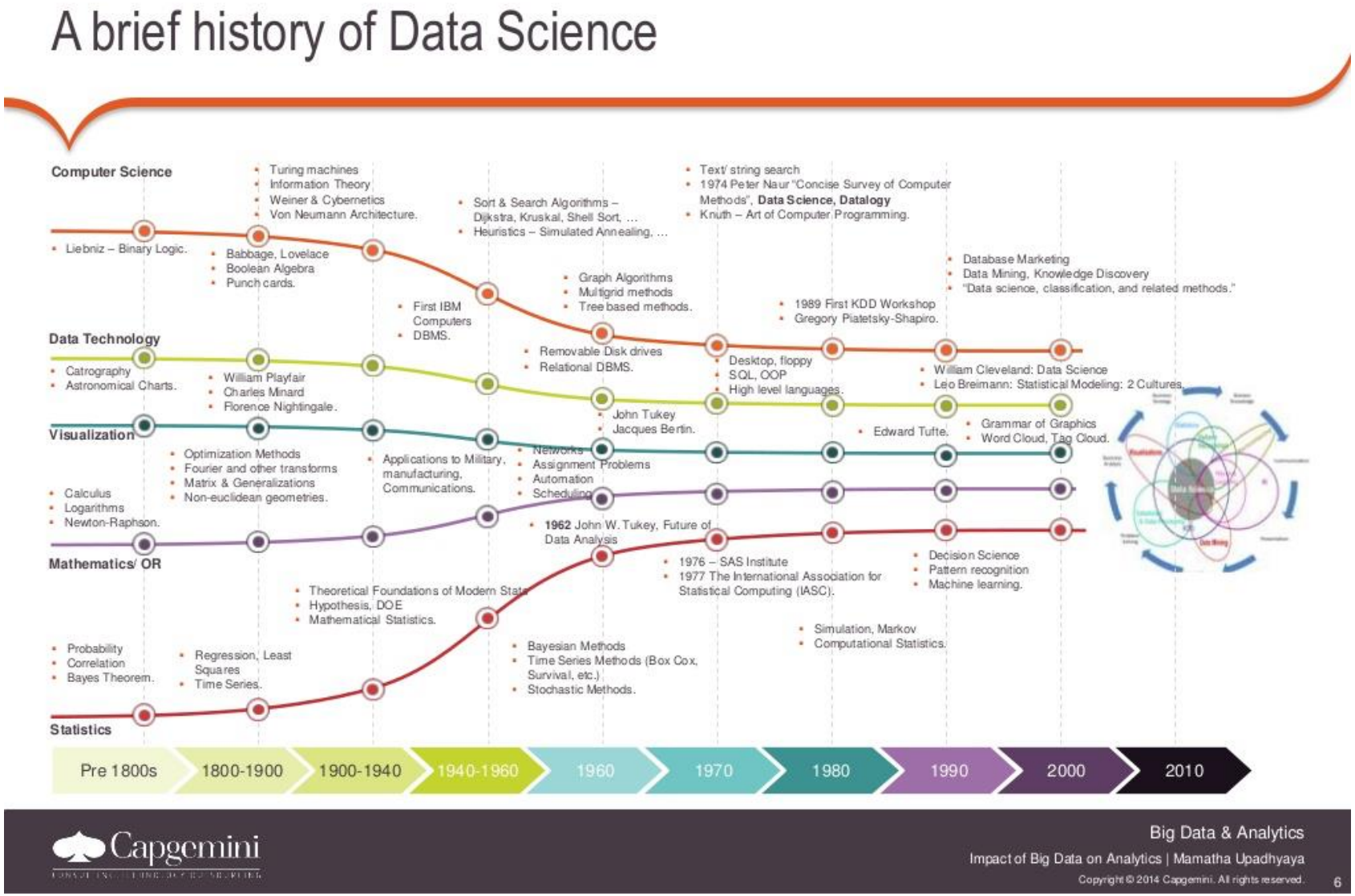
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Complex situations?

- **Data distribution**
  - The large data set is split into chunks or smaller blocks and distributed over N number of nodes or machines.
  - Distributed File System.
- **Parallel processing**
  - The distributed data gets the power of N number of servers and machines in which data is residing and works in parallel for the processing and analysis
  - MapReduce (Google), Hadoop, Spark, Flink, etc.
- **Fault tolerance**
  - We keep the replica of a single block (or chunk) of data more than once.
- **Commodity hardware**
  - We don’t need specialized hardware with special RAID as Data container.
- **Flexibility and Scalability**
  - add more and more of rack space into the cluster as the demand for space increases.

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Big Data Timeline



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Big Data Research Challenges

- **Heterogeneity and incompleteness**
  - machine analysis algorithms expect homogeneous data;
  - even after data cleaning and error correction, some incompleteness and some errors in data are likely to remain.
- **Scale**
  - managing large and rapidly increasing volumes of data;
  - clock speeds have largely stalled and processors are being built with increasing numbers of cores;
  - parallelism across nodes in a cluster;
  - move towards cloud computing;
  - large clusters requires new ways of determining how to run and execute data processing jobs.
- **Timeliness**
  - acquisition rate challenge and a timeliness challenge;
  - many situations in which the result of the analysis is required immediately;
  - ex: a full analysis of a user's purchase history is not likely to be feasible in real-time.
- **Privacy**
  - important to rethink security for information sharing in Big Data use cases;
  - we do not understand what it means to share data.
- **Human collaboration**
  - analytics for Big Data will be designed to have a human in the loop;
  - crowd-sourcing;
  - issues of uncertainty and error => participatory-sensing;
  - the extra challenge here is the inherent uncertainty of the data collection devices.

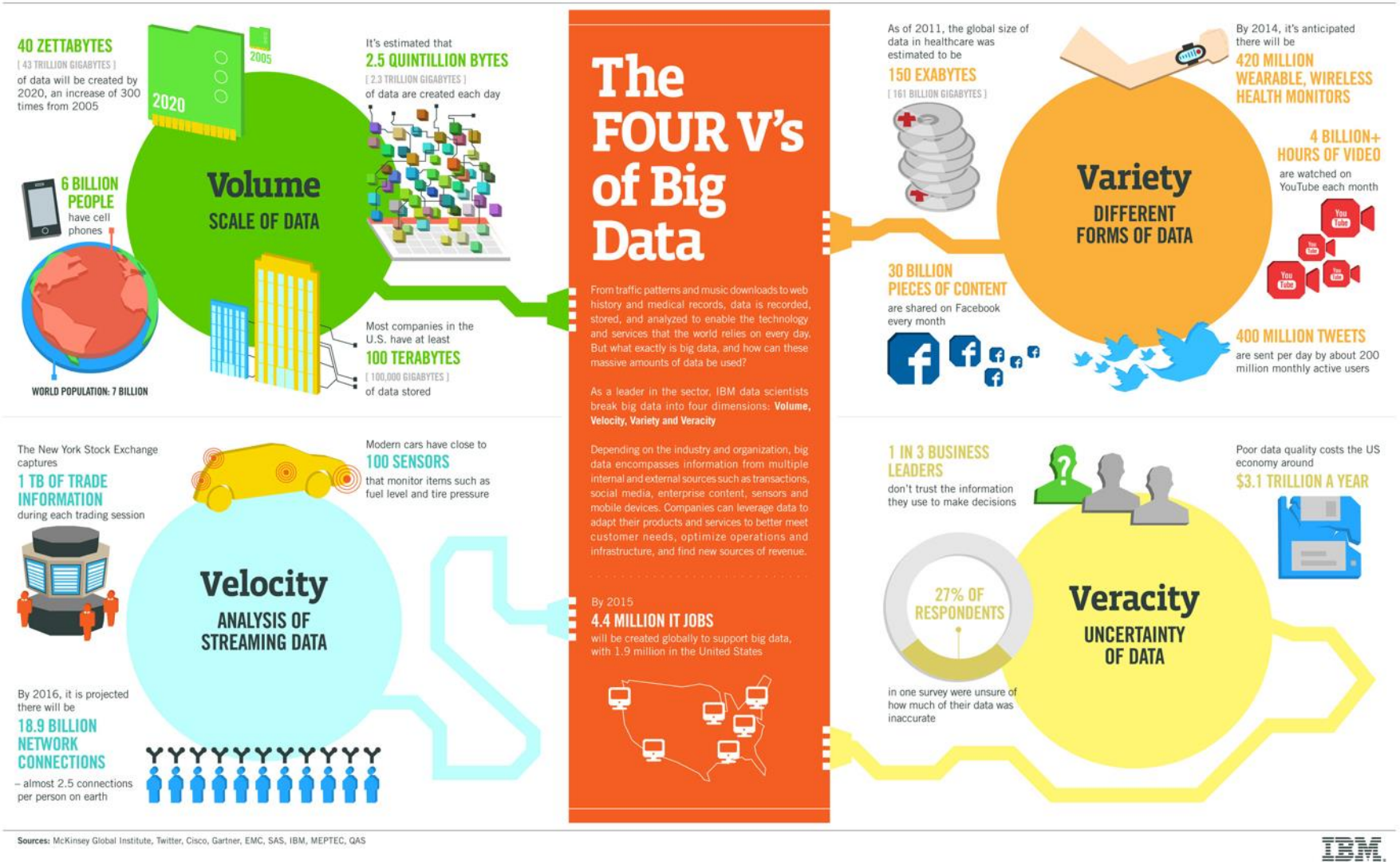
Data Science and the Values of datasets

The 8V's: Volume, Velocity, Variety, Variability, Visualization, Value, Veracity, Vicissitude;

Use cases and current challenges

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

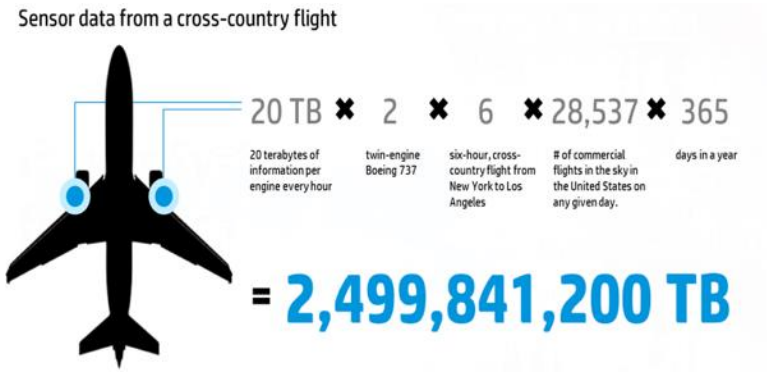
Starting V



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Volume

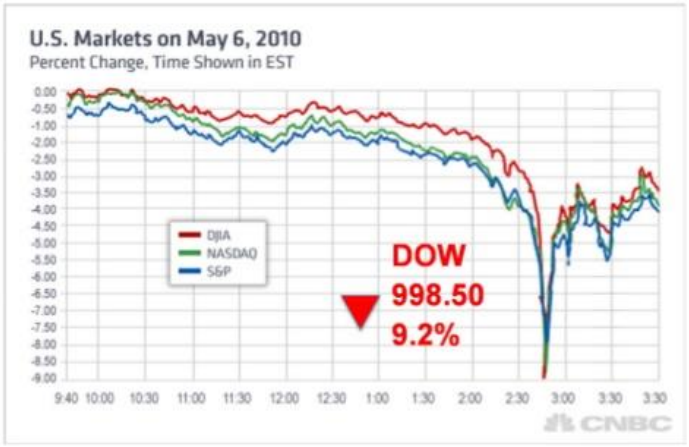
- Big Data **Volume** refers to the large amounts of generated data:
  - 40 ZettaBytes by 2020;
    - 300 times more than 2005;
  - ~ 2.5 Quintillion (2.3 Trillions GB) bytes data created daily;
  - 100 Terabytes per company stored;
  - 6 billion cell phones;
  - 7 billon peoples.
- Volume challenges:
  - Storage, analysis and processing;
  - Even the smallest optimization represents a step ahead in getting the meaningful data in time and at a low cost;



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Velocity

- **Velocity** refers to the speed of generating data (the speed at which the data is flowing):
  - New York Stock Exchange captures 1TB of trade information during each trading session;
  - Modern cars have about 100 sensors;
  - ~18.9 billion network connections
    - about 2.5 connections per person;
  - Social media messages are spread in seconds;

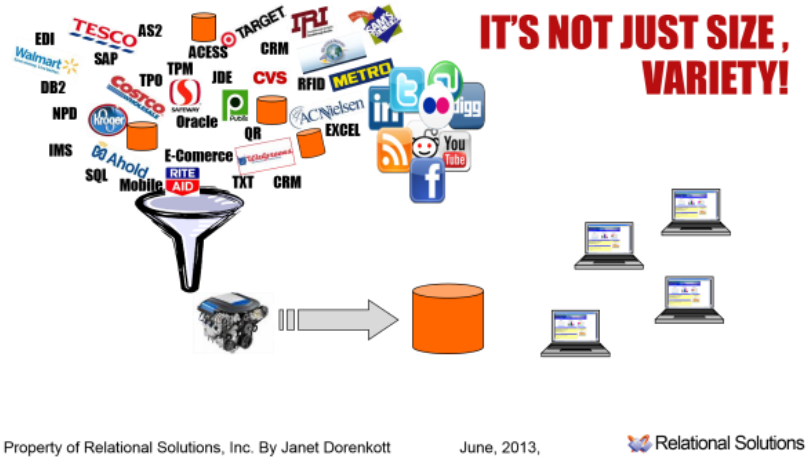


- Velocity challenges:
  - Response time of delivery services (for financial markets is real-time);
  - Speed of manipulating and analyzing complex data;
  - The Big Data system must be able to adjust and deal with the high loads of data at peak times;

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Variety

- **Variety** refers to the diversity of data:
  - 150 Exabytes in healthcare (2011);
  - ~30 billion pieces of contend shared on Facebook;
  - By 2014 ~420 mil. wearable, wireless health monitors;
  - 4 billion hours of videos watched on YouTube per month;
  - 400 million tweets send /day by 200 mil. active users;



- Variety challenges:
  - Big Data systems should handle complex data: relational data, raw data, semi-structured or unstructured data;
  - Almost 80% of the world's data is now unorganized;
    - traditional DB can't be used anymore for storage and management
  - different types of data must be processed and mapped to dedicated resources;

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Variability

- **Variability** refers to data whose meaning is constantly changing;
- For example words don't have static definitions and meaning vary in context:
  - "Delicious muesli from the @imaginarycafe- what a **great** way to start the day!"
  - "**Greatly** disappointed that my local Imaginary Cafe have stopped stocking BLTs."
  - "Had to wait in line for 45 minutes at the Imaginary Cafe today. **Great**, well there's my lunchbreak gone…"
  - "great" on its own is not a sufficient signifier of positive sentiment;
- Variability challenges:
  - 'Understand' context and decode the precise meaning of words

Types of Urban Big Data

Urban Big Data	Who?	Example	"V(s)" Illustrated
Sensor systems	Public and private utility and service operators; building/infrastructure managers	Array of Things, Chicago	Velocity, Variability
User-generated content	Various; usually private sector systems	Austin Bike Data Example	Veracity
Administrative systems	Governments & public vendors	Oyster Data, Transport for London	Volume, Visualization
Private sector data	Various	Craigslist Rental Listings Analysis	Value
Arts and Humanities Data	Digital humanities organizations	Mapping Inequality Project	Variety
Hybrid data	Data intermediaries	Smart Locations Database, EPA	Veracity

Categories from: Thakuriah, Piyushimita Vonu, Nebiyou Tilahun, Moira Zellner, P Thakuriah, N Tilahun, and M Zellner. 2015. "Big Data and Urban Informatics: Innovations and Challenges to Urban Planning and Knowledge Discovery." Proc. NSF Workshop on Big Data and Urban Informatics.

TAUBMAN COLLEGE architecture + urban planning University of Michigan

$$Mean(sample) = \mu = \frac{\sum_{i=1}^{k} f_i x_i}{n-1}$$

$$StandardDeviation(sample) = \sigma = \sqrt{\sum_{i=1}^{k} \frac{f_i (x_i - \mu)^2}{n-1}}$$

$$Variance(sample) = \sigma^2 = \sum_{i=1}^{k} \frac{f_i (x_i - \mu)^2}{n-1}$$

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Visualization

- **Visualization** refers to presentation of data in a pictorial or graphical format:
  - Analytics presented visually:
    - to understand difficult concepts;
    - to identify new patters;
  - Interactive visualization:
    - charts;
    - graphs;
    - ex: https://d3js.org/, https://developers.google.com/chart/ , http://www.fusioncharts.com/,

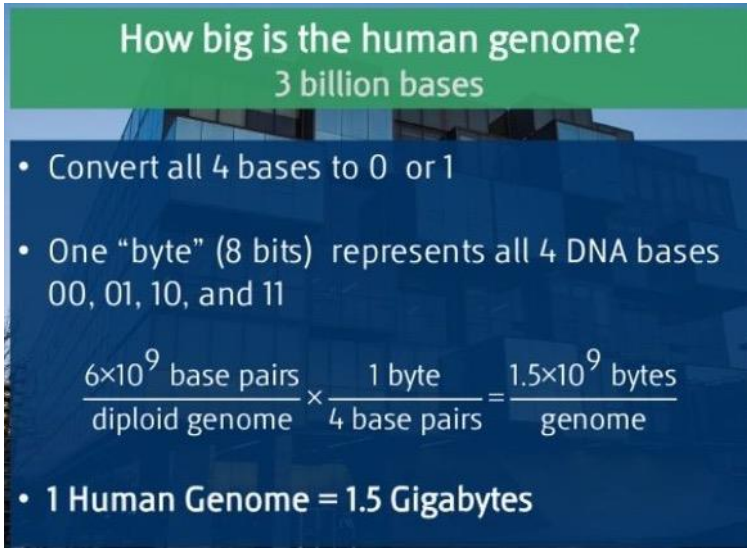
- Visualization challenges:
  - Develop new techniques for data visualization;
  - Data exploration;
  - Inconsistencies in the data structure;

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Value

- **Value** refers to the value that can be extracted form datasets:
  - it does not matter how big the data volume is or how complex,
  - it is unless we are able to extract the meaningful information;
  - storing and processing meaningless data represents:
    - a waste of money, time,

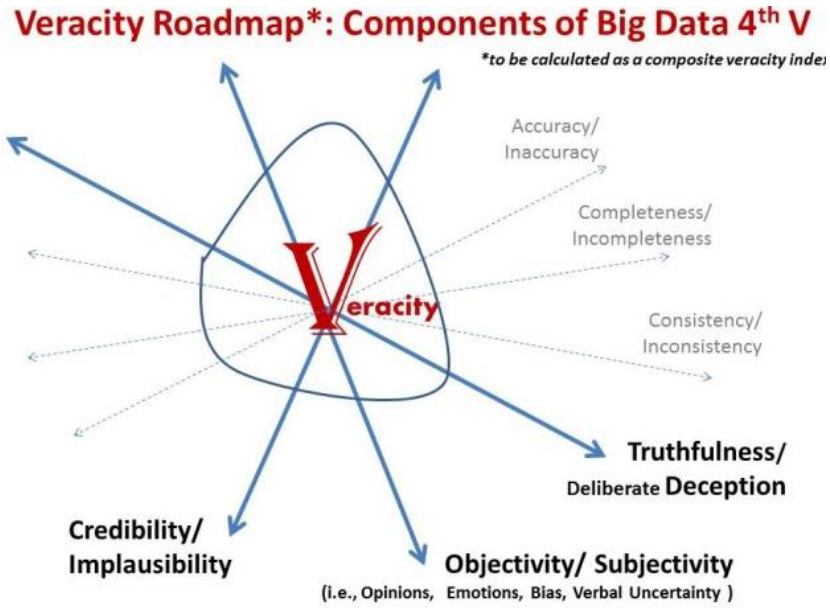
    and obtaining the relevant information becomes harder;
  - Traditional approach to Big Data: systems for storing big data, but getting insights can be difficult:
    - time-intensive, manual query and analysis;
    - modern dashboarding tools;
  - Modern approach to Big Data:
    - Machine brains: specialized algorithms that :
      - can learn to identify patterns,
      - correlations and
      - indicators through training and repetition;
- Big Data Value challenges:
  - developing tools for getting value form Big Data;



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Veracity

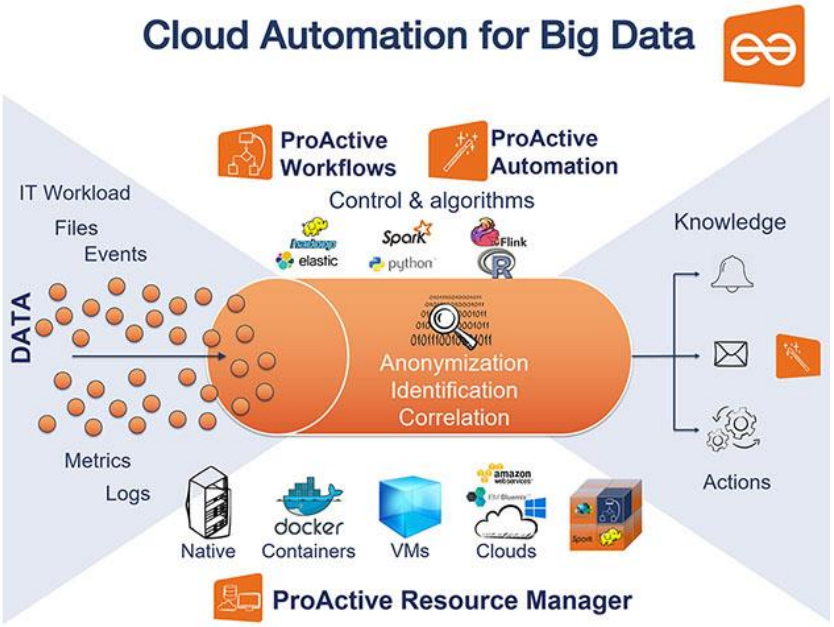
- **Veracity** refers to the trustworthiness of the data (uncertainty of data):
  - 1 in 3 business leaders don't trust the information they use to make decisions;
  - 27% of respondents in one survey were unsure how much of their data was inaccurate;
  - Poor data quality cost US economy about $1.3 trillion/year;
- Causes of veracity:
  - Rumors, spammers, collection errors, entry errors, system errors;
- Veracity challenges:
  - the volume and complexity of data are increasing;
  - quality and accuracy are becoming less controllable;

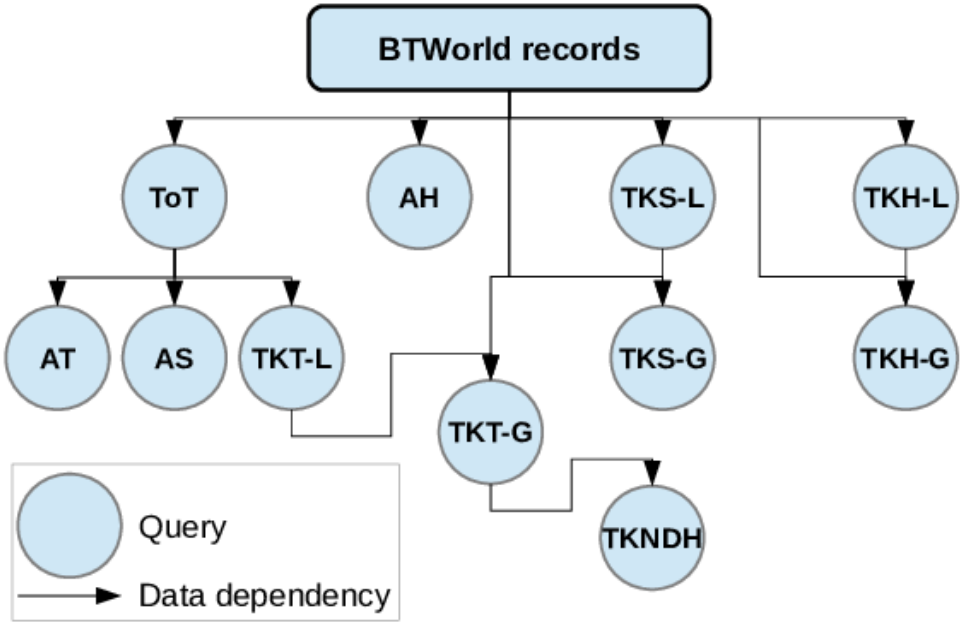


FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Vicissitude

- **Vicissitude** property refers to the challenge of scaling Big Data complex workflows;





BTWorld: A Large-scale Experiment in Time-Based Analytics

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

+ more Vs of Big Data

+ Volatility
+ Veridicity
+ Vision
+ …

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

10 Cool Big Data Projects

- #1. To find exactly what we look for in the internet
- #2. To ride through a city without traffic jams
- #3. To save rare animals, catching poachers
- #4. To make our cities green
- #5. To understand why Indian cuisine is unique
- #6. To fight malaria epidemics in Africa
- #7. To grow ideal Christmas trees
- #8. To understand that our languages are filled with happiness
- #9. To make sport shows even more interesting
- #10. To improve job conditions
- #11. To enhance relationship

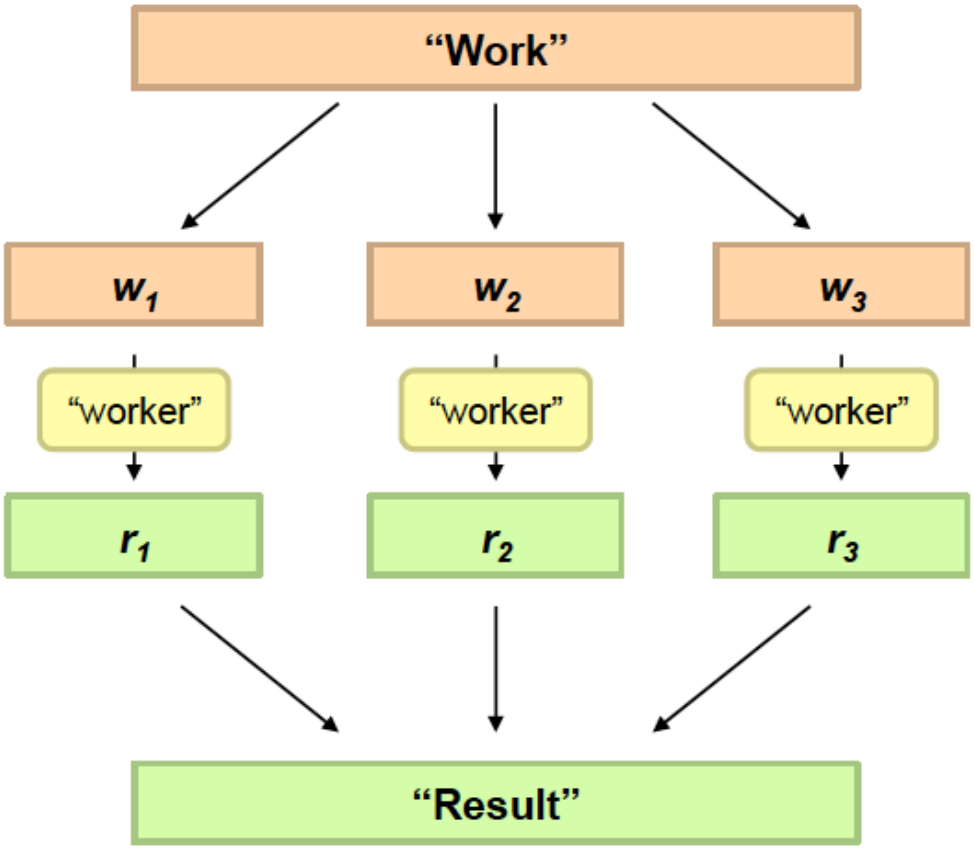
https://www.kaspersky.com/blog/cool-big-data-projects/8186/

Distributed Model for Scalable Batch Computing

Google MapReduce. MapReduce Patterns, Algorithms and use cases

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Philosophy to Scale for Big Data



Divide Work

Combine Results

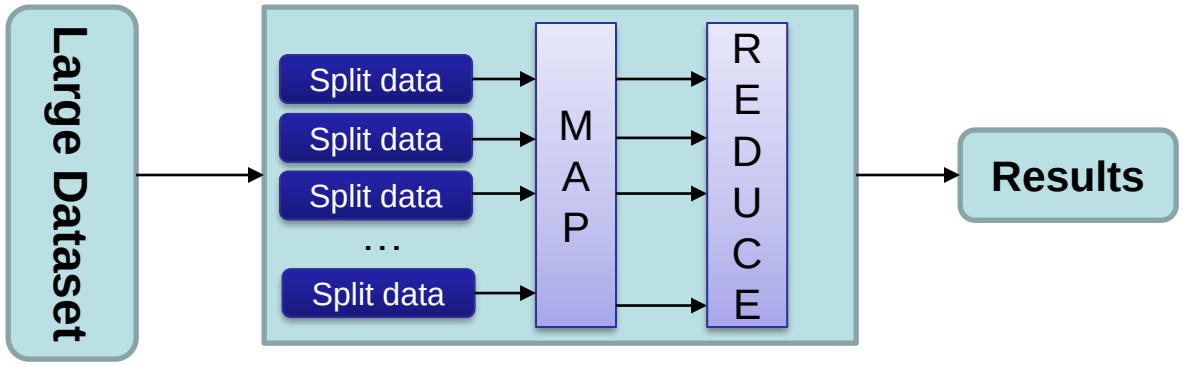
Distributed processing is non-trivial

- 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?
- 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 …

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Google MapReduce

- MapReduce is:
  - a programming model;
  - implementation for processing and generating big data sets;
- Released in 2004 by Google (Jeffrey Dean and Sanjay Ghemawat)
- Perform simple computations while hiding the details of:
  - Parallelization;
  - Data distribution;
  - Load balancing;
    - I/O scheduling;
  - Fault tolerance;



- Distribute data over multiple nodes
  - Each node receive a small portion of data

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Google MapReduce (2)

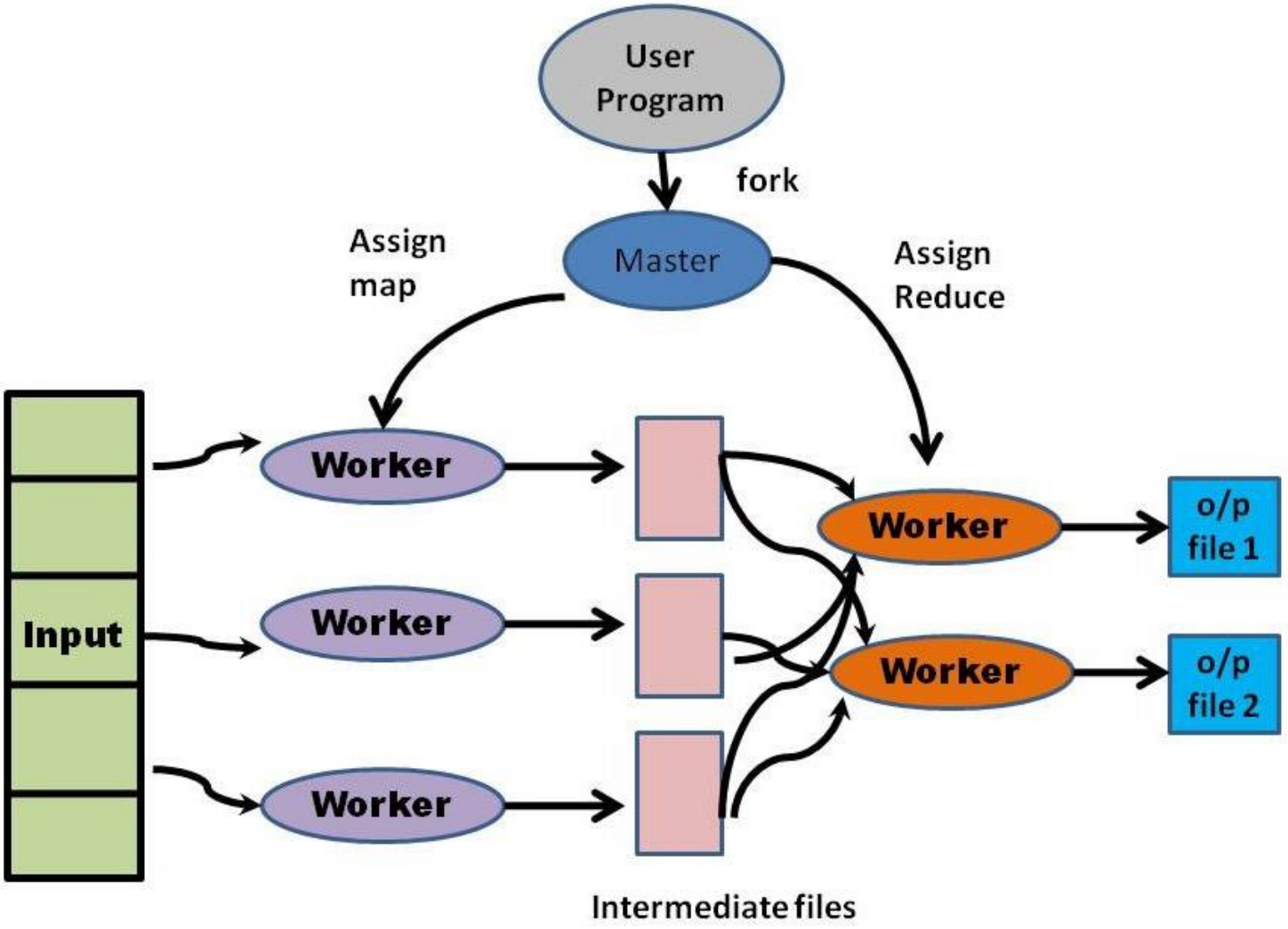
- MapReduce programming model:
  - Input & Output: each a set of key/value pairs
- Programmer must specifies two functions:
  - **map (in_key, in_value) → list(out_key, intermediate_value)**
    - Processes input key/value pair;
    - Produces set of intermediate pairs;
  - **reduce (out_key, list(intermediate_value)) → list(out_value)**
    - Combines all intermediate values for a particular key;
    - Produces a set of merged output values;

```
map(String input_key, String
input_value):
// input_key: document name
// input_value: document contents f
   for each word w in input_value:
        EmitIntermediate(w, "1");
```

```
reduce(String output_key, Iterator
intermediate_values):
// output_key: a word
// output_values: a list of counts int
result = 0;
for each v in intermediate_values:
        result += ParseInt(v);
Emit(AsString(result));
```

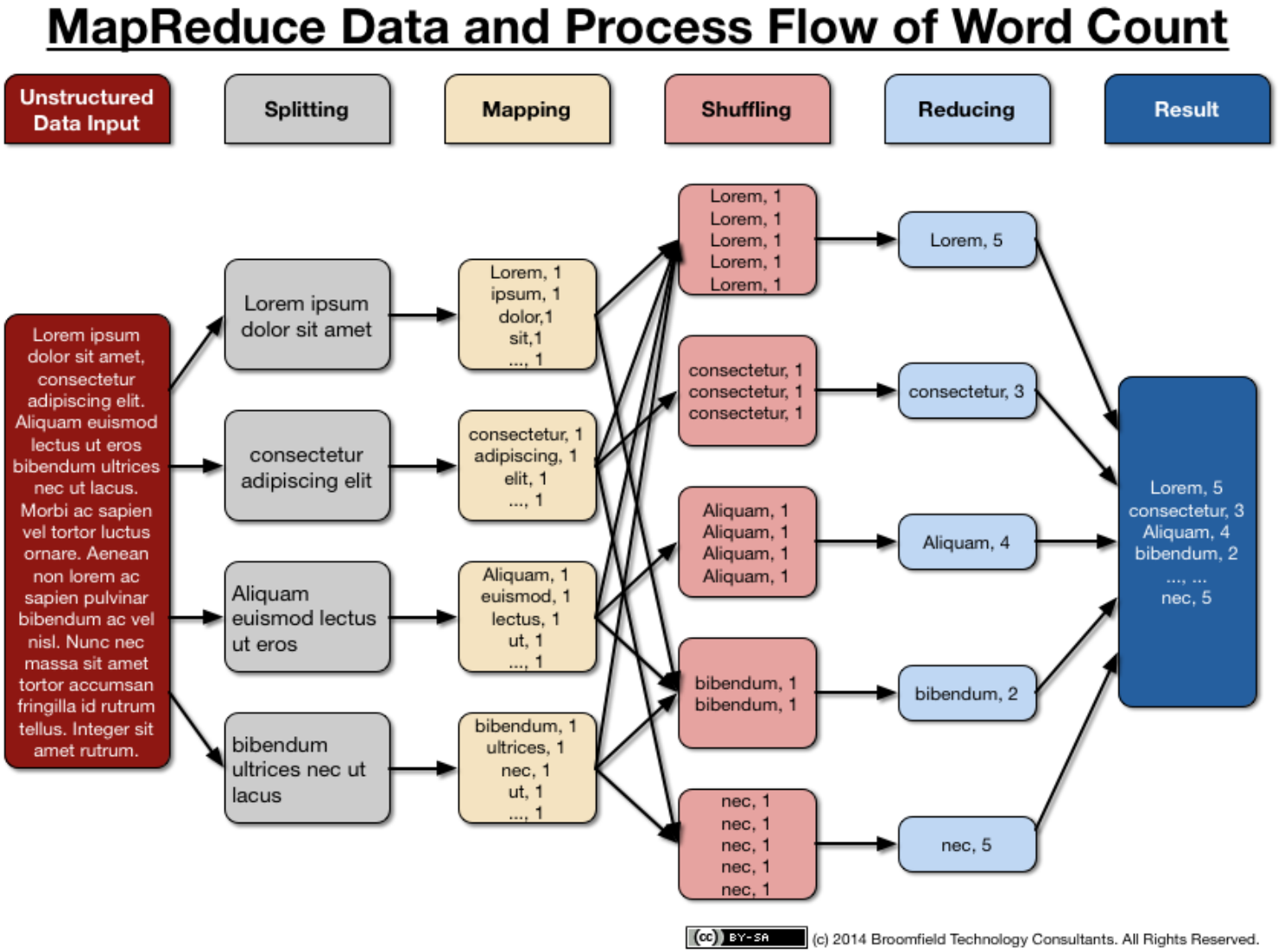
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

MapReduce Architecture



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

MapReduce Data Flow



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

MapReduce Patterns

- **Counting and Summing -** log analysis, data querying:
  - Calculate a total number of occurrences of each word in a large set of documents;
  - Process a log file :
    - each record contains a response time;
    - it is required to calculate an average response time.
- **Collating –** ETL, building of inverted indexes:
  - Save all items that have the same value of a function into one file
- **Filtering ("Grepping"), Parsing, and Validation -** log analysis, data querying, ETL, data validation:
  - Collect all records that meet some condition;
  - Transform each record (independently from other records) into another representation

MapReduce Patterns (2)

- **Distributed Task Execution –** physics and engineering simulations, numerical analysis, performance testing:
  - Large computational divided into multiple parts;
  - Results from all parts are combined together to obtain a final result.
- **Sorting -** ETL, data analysis:
  - Sort large set of records by some rule;
  - Process records in a certain order.
- **Iterative Message Passing (Graph Processing) -** graph analysis, web indexing:
  - Calculate a state of each entity in a network on the basis of properties of the other entities in its neighborhood;
  - State can represent :
    - distance to other nodes;
    - indication that there is a neighbor with the certain properties;
    - characteristic of neighborhood density ;
    - etc.

MapReduce Patterns (3)

- **Distinct Values (Unique Items Counting)** – log analysis, unique users counting
  - Set of records that contain fields F and G;
  - Count the total number of unique values of field F for each subset of records that have the same G (grouped by G).
- **Cross-Correlation -**text analysis, market analysis:
  - A set of tuples of items ;
  - Calculate a number of tuples where some items co-occur for each possible pair.
- **Relational MapReduce Patterns**:
  - Selection
  - Projection
  - Union
  - Intersection
  - Difference
  - GroupBy and Aggregation
  - Joining

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Algorithms and use cases

- PageRank and Mapper-Side Data Aggregation
  - Used by Google to calculate relevance of a web page as a function of authoritativeness (PageRank) of pages that have links to this page.
  - the algorithm is complex
  - its core it is just a propagation of weights between nodes
    - each node calculates its weight as a mean of the incoming weights:

```
class N
    State is PageRank
method getMessage(object N)
    return N.State / N.OutgoingRelations.size()
method calculateState(state s, data [d1, d2,…])
    return ( sum([d1, d2,...]) )
```

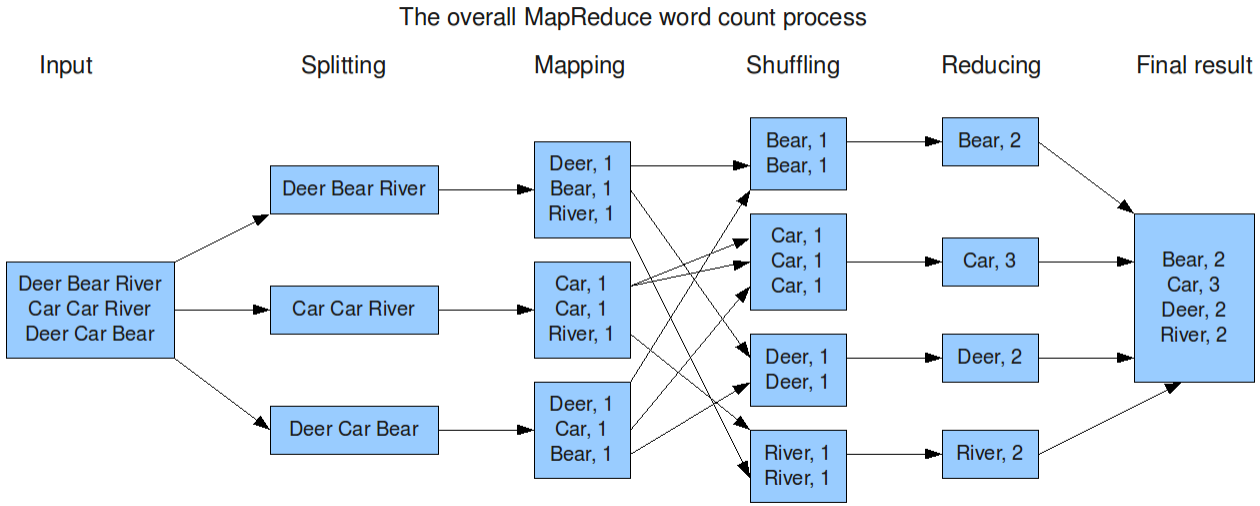
```
class Mapper
   method Initialize
      H = new AssociativeArray
   method Map(id n, object N)
      p = N.PageRank  /
N.OutgoingRelations.size()
      Emit(id n, object N)
      for all id m in N.OutgoingRelations
do
         H{m} = H{m} + p
   method Close
      for all id n in H do
         Emit(id n, value H{n})
```

```
class Reducer
   method Reduce(id m, [s1, s2,...])
      M = null
      p = 0
      for all s in [s1, s2,...] do
         if IsObject(s) then
            M = s
         else
            p = p + s
      M.PageRank = p
      Emit(id m, item M)
```

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Word Count Example

- Mapper
  - Input: value: lines of text of input
  - Output: key: word, value: 1
- Reducer
  - Input: key: word, value: set of counts
  - Output: key: word, value: sum
- Launching program
  - Defines this job
  - Submits job to cluster



Owen O'Malley (Yahoo!)

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Word Count Mapper

```
public static class Map extends MapReduceBase implements
Mapper<LongWritable,Text,Text,IntWritable> {
    private static final IntWritable one = new IntWritable(1);
    private Text word  = new Text();

    public static void map(LongWritable key, Text value,
    OutputCollector<Text,IntWritable> output, Reporter reporter)
    throws IOException {
          String line = value.toString();
          StringTokenizer = new StringTokenizer(line);
          while(tokenizer.hasNext()) {
             word.set(tokenizer.nextToken());
             output.collect(word,one);
             }
          }
    }
```

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Word Count Reducer

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

    public static 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));
        }
}
```

Putting it all together

```
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.setReducer(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);
```

Apache Hadoop

Programming environment and use cases

* Based on slides by Dhruba Borthakur, Owen O'Malley (Yahoo!)

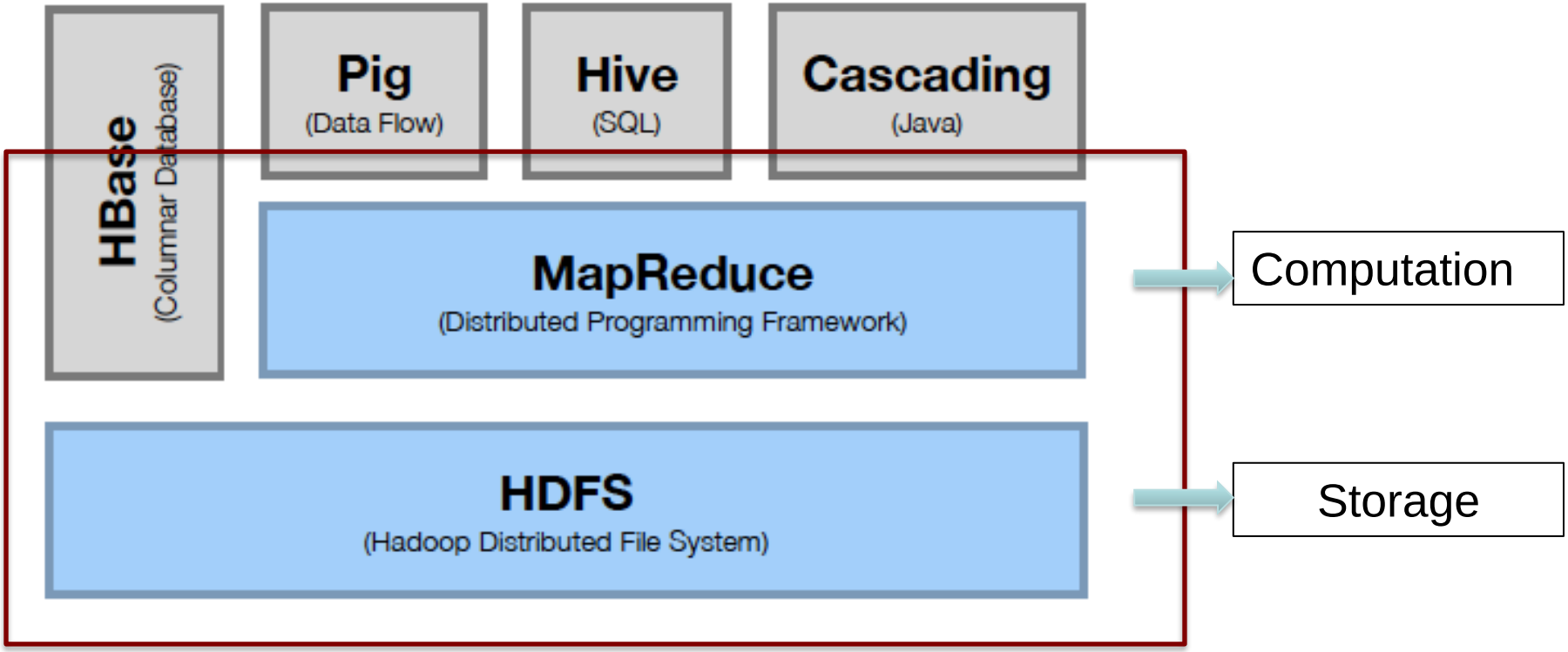
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

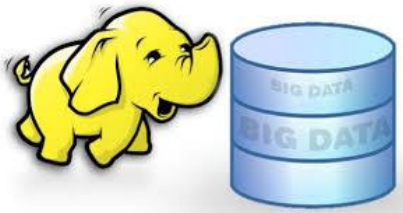
The history of an elephant

- 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 is an **Apache** open source project from 2006
- Why Hadoop?
  - Need to process huge datasets on large clusters of computers
  - Very expensive to build reliability into each application
  - Nodes fail every day
    - Failure is expected, rather than exceptional
    - The number of nodes in a cluster is not constant
  - Need a common infrastructure
    - Efficient, reliable, easy to use

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Apache Hadoop Basic Modules





FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Distributed File System

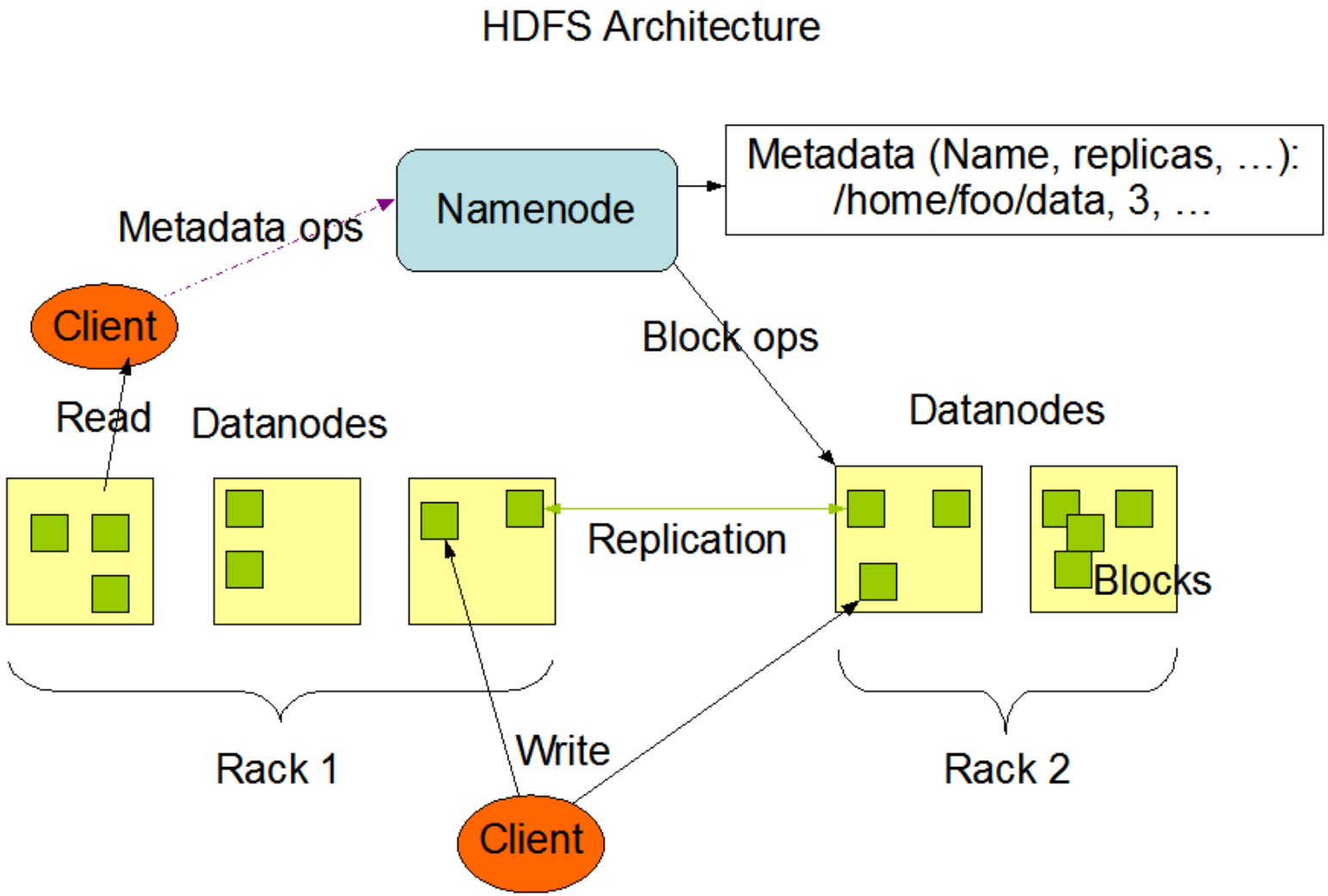
- Single Namespace for entire cluster
- Data Coherency
  - Write-once-read-many access model
  - Client can only append to existing files
- Files are broken up into blocks
  - Typically 64MB block size
  - Each block replicated on multiple Data Nodes
- Intelligent Client
  - Client can find location of blocks
  - Client accesses data directly from Data Node

Hadoop HDFS

- Hadoop distributed File System (based on Google File System (GFS) paper, 2004)
  - Serves as the distributed file system for most tools in the Hadoop ecosystem
  - Scalability for large data sets
  - Reliability to cope with hardware failures
- HDFS good for:
  - Large files
  - Streaming data
- Not good for:
  - Lots of small files
  - Random access to files
  - Low latency access

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

HDFS Architecture



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Who uses Hadoop?

- Amazon/A9
- Facebook
- Google
- New York Times
- Veoh
- Yahoo!
- …. many more

Hadoop Cluster at Yahoo! (Credit: Yahoo)

A real world example of New York Times

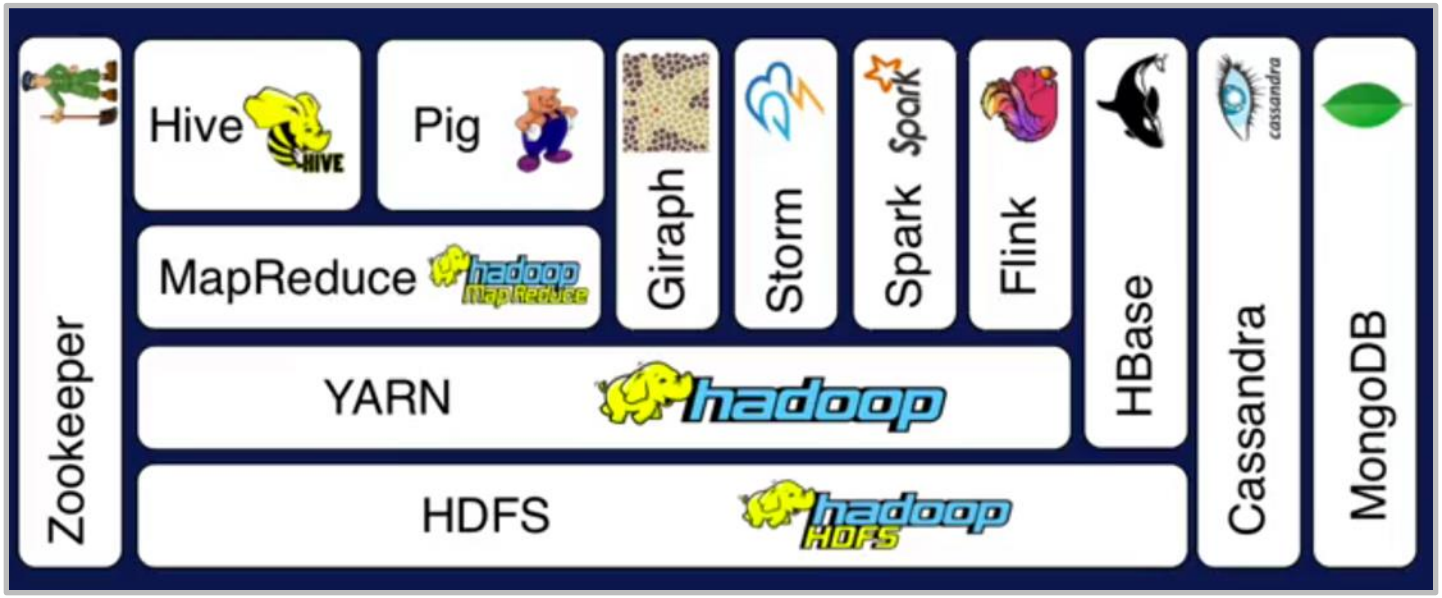
- **Goal:** Make entire archive of articles available online: 11 million, from 1851
- **Task:** Translate 4 TB TIFF images to PDF files
- **Solution:** Used Amazon Elastic Compute Cloud (EC2) and Simple Storage System (S3)
- **Time: < 24 hours**
- **Costs: ~ $240**

The New York Times

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Hadoop Ecosystem

- Pig: High-level language for data analysis (Yahoo! Research)
- Hive: SQL-like Query language and Metastore (Facebook)
- HBase: Table storage for semi-structured data (Modeled on Google's Bigtable)
- Zookeeper: Coordinating distributed applications
- Mahout: Machine learning



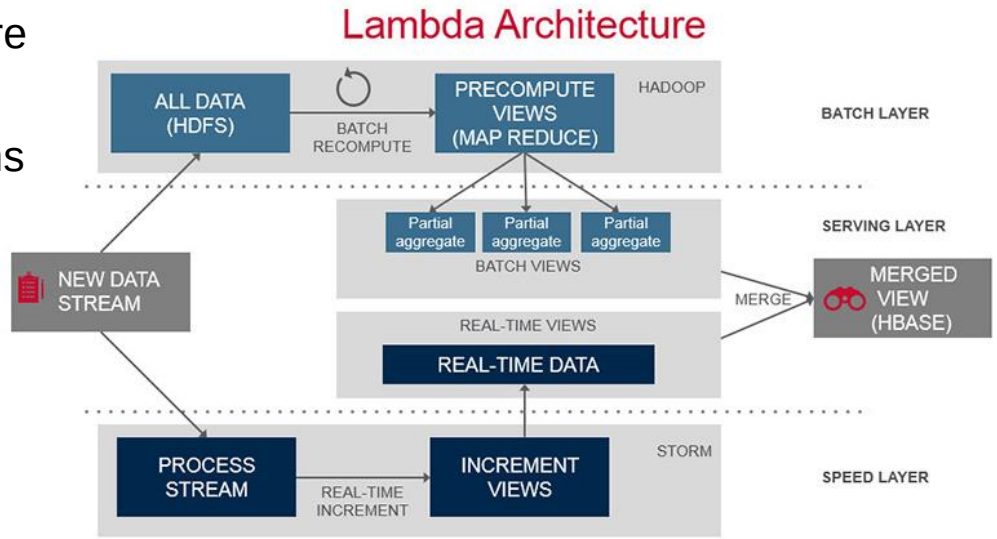
In-Memory Cluster Computing

Apache Spark

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Overview of LAMBDA Architecture

- The Lambda Architecture has three major components:
  - Batch layer provides the following functionality
    - managing the master dataset, an immutable, append-only set of raw data
    - pre-computing arbitrary query functions, called batch views.
  - Serving layer
    - indexes the batch views so that they can be queried in ad hoc with low latency.
  - Speed layer
    - accommodates all requests that are subject to low latency requirements.
    - use fast and incremental algorithms
    - deals with recent data only.

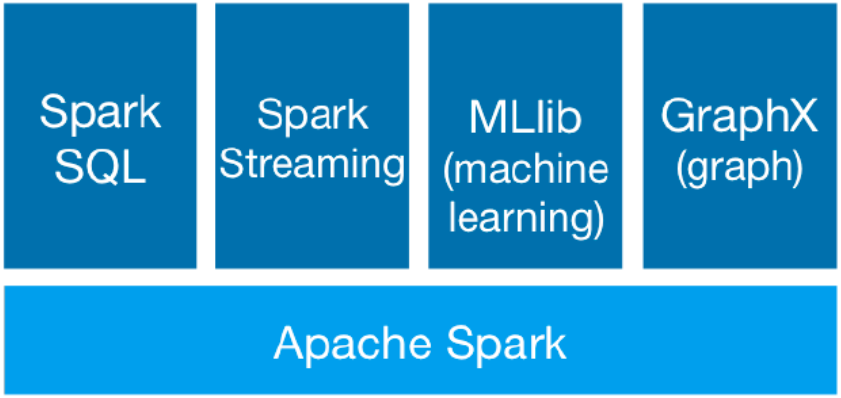


FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Apache Spark

- Spark is a general-purpose computing framework for iterative tasks
- API is provided for Java, Scala and Python
- The model is based on MapReduce
  - enhanced with new operations and an engine that supports execution graphs
- Tools include Spark SQL, MLLlib for machine learning, GraphX for graph processing and Spark Streaming

- Unifies batch, streaming, interactive computing
- Making it easy to build sophisticated applications



Key idea in Spark

- Resilient distributed datasets (RDDs)
  - Immutable collections of objects across a cluster
  - Built with parallel transformations (map, filter, …)
  - Automatically rebuilt when failure is detected
  - Allow persistence to be controlled (in-memory operation)
- Transformations on RDDs
  - Lazy operations to build RDDs from other RDDs
  - Always creates a new RDD
- Actions on RDDs
  - Count, collect, save

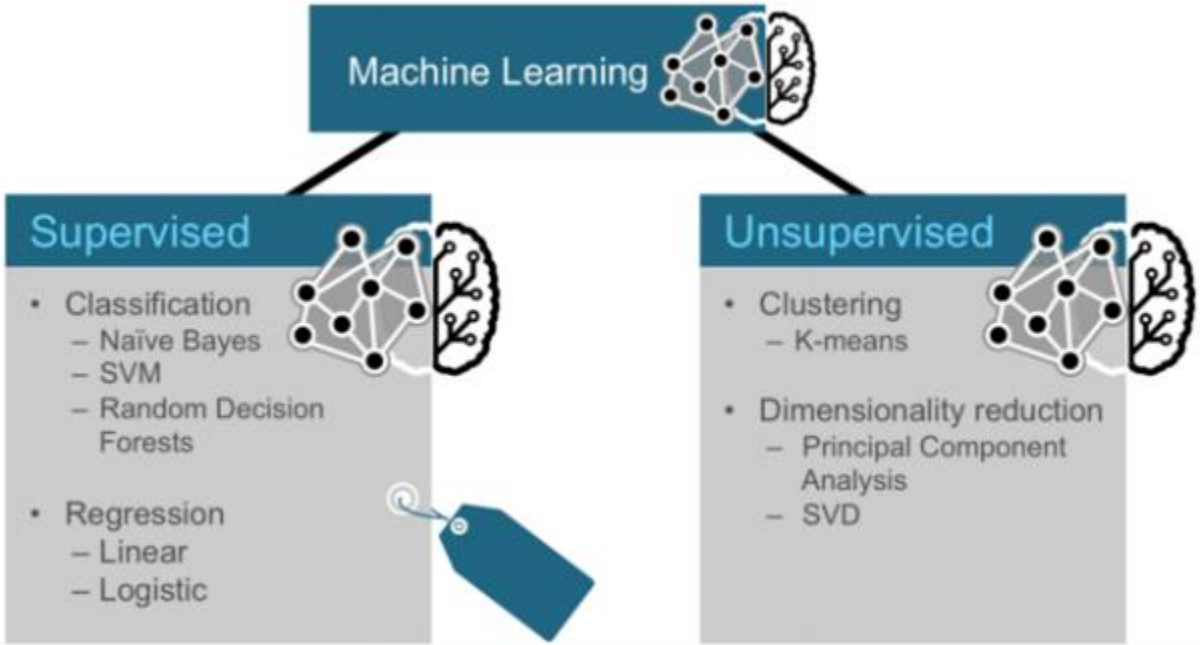
Implementing Spark Algorithms

- **Broadcast everything**
  - Master broadcasts data and initial models
  - At each iteration updated models are broadcast by master (driver program)
  - Does not scale well due to communication overhead
- **Data parallel**
  - Worker loads data
  - Master broadcasts initial models
  - At each iteration updated models are broadcast by master
  - Works for large datasets, because data is available to workers
- **Fully parallel**
  - Workers load data and they instantiate the models
  - At each iteration, models are shared via join between workers
  - Much better scalability

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Machine Learning and Spark

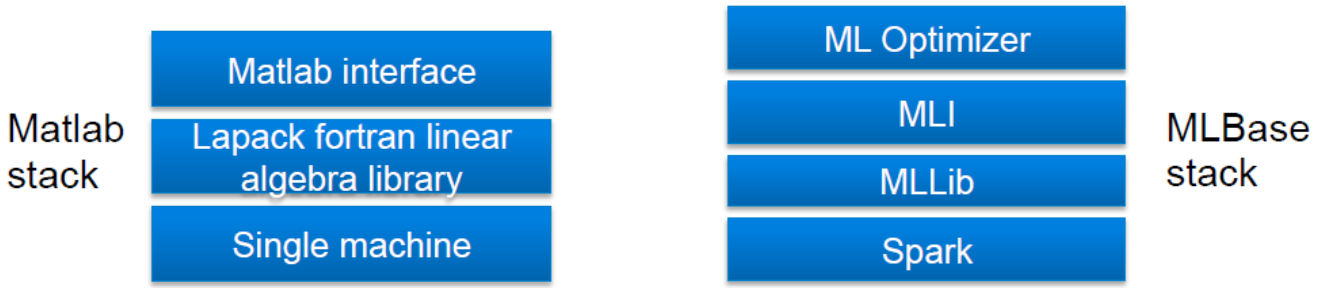
- Spark RDDs support efficient data sharing
- In-memory caching increases performance
  - Reported to have performance of up to 100 times faster than Hadoop in memory or 10 times faster on disk
- High-level programming interface for complex algorithms



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

MLBase and MLlib

- MLBase has been designed for simplifying the development of machine learning pipelines:
  - MLlib is a machine learning library
  - MLI (ML Developer API) is an API for machine learning development that aims to abstract low-level details from the developers
  - MLOpt is a declarative layer that aims to automate the machine learning pipeline
    - The idea is that the system searches feature extractors and models best fit for the ML task



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

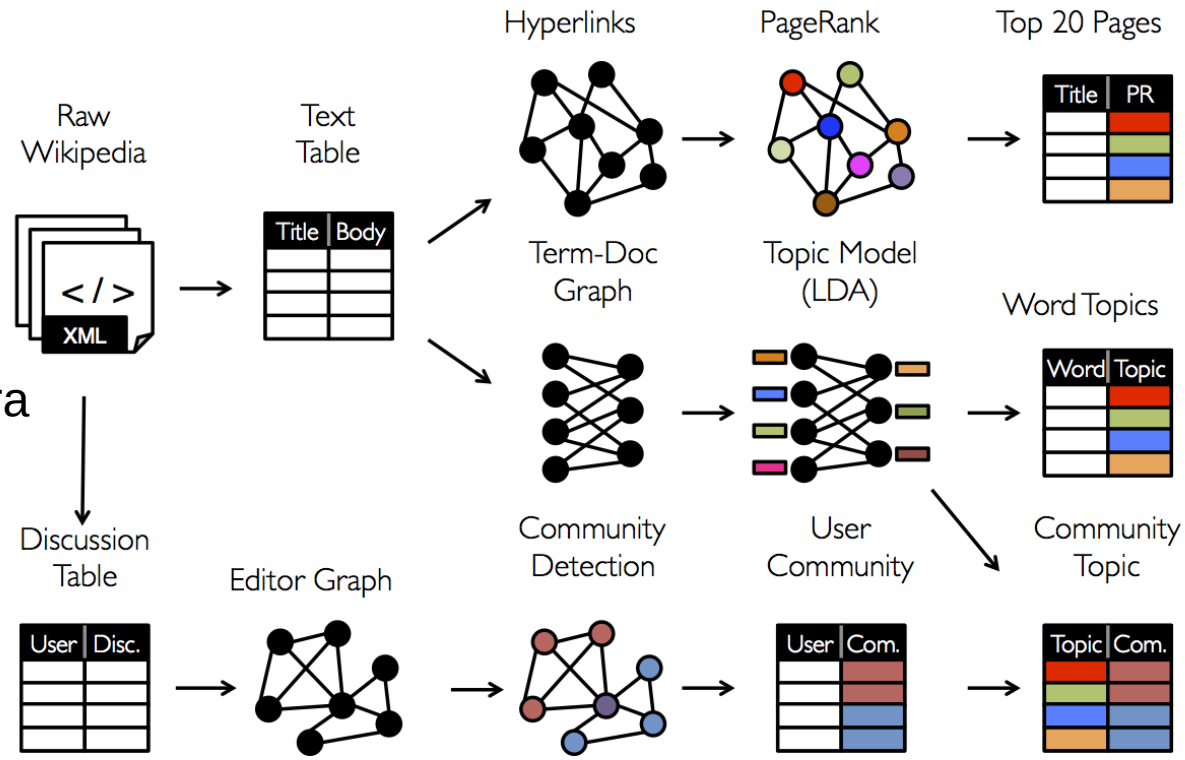
Graph-Parallel Systems

- Graph-based computation depends only on the neighbors of a particular vertex
  - "Think like a Vertex." – Pregel (SIGMOD 2010)
- Systems with specialized APIs to simplify graph processing
  - **Pregel from Google**
    - Push abstraction: Vertex programs interact by sending messages
    - Receive msgs, process, send msgs
- **GraphLab**
  - Pull abstraction: Vertex programs access adjacent vertices and edges
  - Foreach (j in neighbours) calculate pagerank total for j

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

GraphX

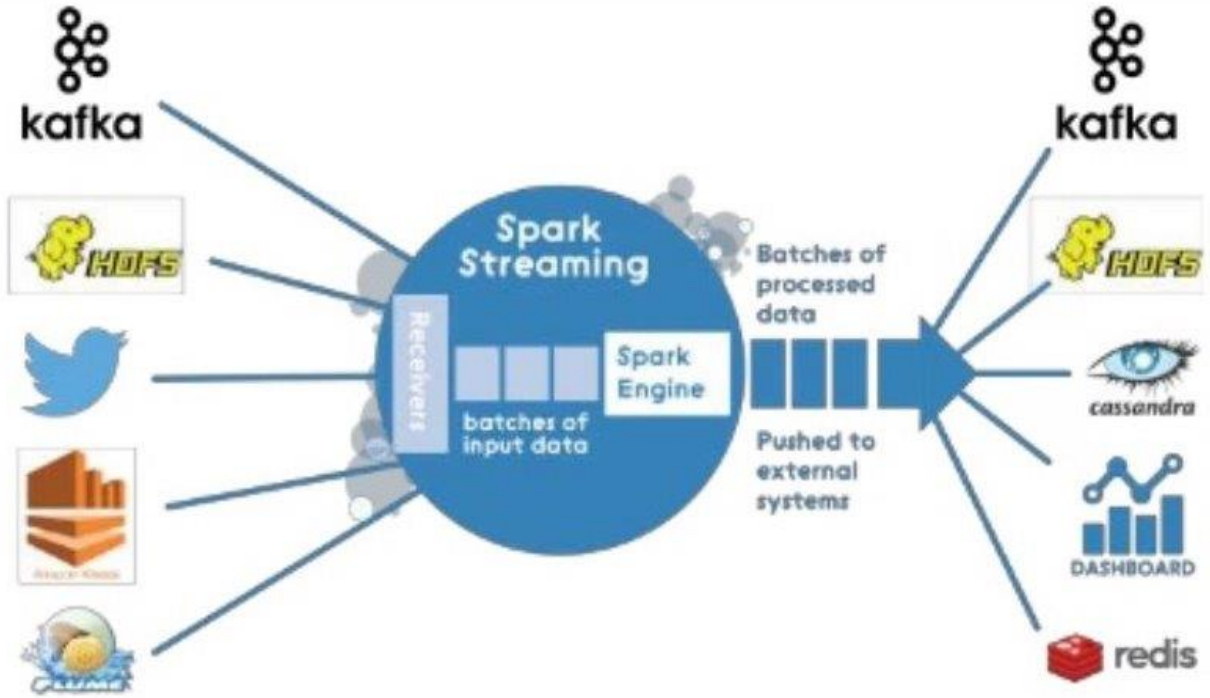
- Separation of system support for each view (table, graph) involves expensive data movement and duplication
- GraphX makes tables and graphs views of the same physical data
- The views have their own optimized semantics
  - Table operators inherited from Spark
  - Graph operators form relational algebra



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Spark Streaming

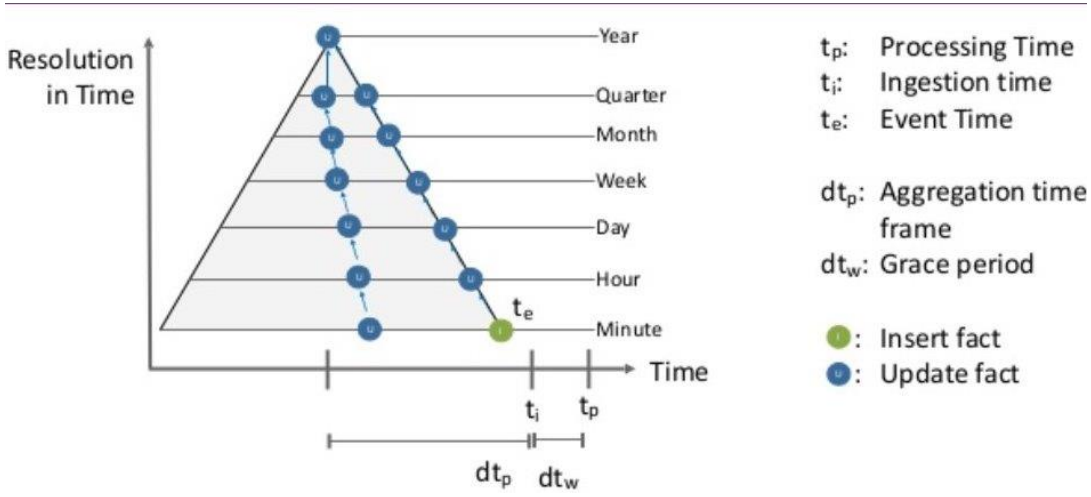
- Spark extension of accepting and processing of streaming high-throughput live data streams
- Data is accepted from various sources
  - Kafka, Flume, TCP sockets, Twitter, …
- Machine learning algorithms and graph processing algorithms can be applied for the streams
- Similar systems
  - Twitter (Storm), Google (MillWheel), Yahoo! (S4)



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Stream Processing: Discretized

- Streaming computation: a series of very small deterministic batch jobs
  - Live stream is divided into batches of x seconds
  - Each batch of data is an RDD and RDD operations can be used
  - Results are also returned in batches
  - Batch size as low as 0.5 seconds, results in approx. One second latency
- **Can combine streaming and batch processing**



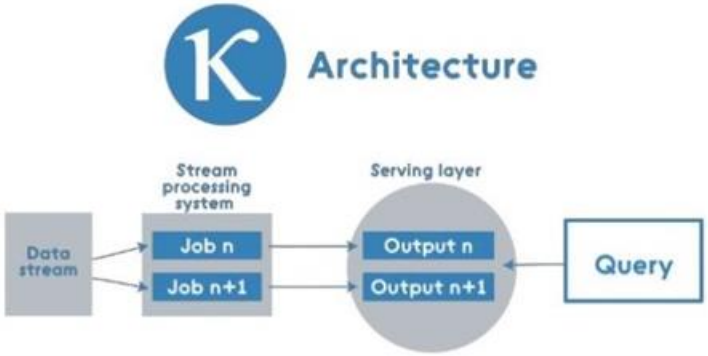
Click stream processing example
Event vs. Processing Time

- There's a difference between even time (t_e) and processing time (t_p)
- Events arrive out-of order even during normal operation
- Events may arrive arbitrary late
- Apply a grace period before processing events
- Allow arbitrary update windows of metrics

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

K – Architecture

- A distributed data operating system is emerging
  - Supported by YARN and MESOS
- Various data services on top of this (Hadoop and Spark)
- Some services are being integrated (for example in Spark) for better coherence and performance
- Important points
  - Data format (row/column, block size)
  - Network topology and data/code placement
  - Algorithm structure and coordination
  - Scheduling and resource management



- Big Data Frameworks are evolving
  - Spark represents unification of streaming, machine learning and graphs
- Big Data pipeline management is at an early stage
  - How to achieve better mapping between cluster resources, scheduling, and pipelines?

Ethical Issues on Big Data

Rules and Regulations, Data Economy, Business Models

Big Data Ethics

- "Big Data" Revolution comparable with Industrial Revolution
  - all kinds of human activities and decisions are beginning to be influenced by big data predictions
    - dating, shopping, medicine, education, voting, law enforcement, terrorism prevention, and cybersecurity
- Privacy
  - Privacy as Information Rules
  - Shared Private Information Can Remain Confidential
  - Transparency
- Identity

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Big Data Ethics Principles

- Beneficial:
  - it should deliver value to all concerned parties
- Progressive:
  - deliver better and more valuable results
  - minimize data usage to promote more sustainable and less risky analysis
- Sustainable:
  - provide value that is sustainable over a reasonable time frame
    - Data sustainability
    - Algorithmic sustainability
    - Device- and/or manufacturer-based sustainability
- Respectful:
- Fair

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Case Studies

- Trump election
- Target pregnancy ad
- Brexit campaign
- Ted Cruz campaign surge
- UK Tory election campaign
- Obama campaign
- Air France 447 crash
- Compas system

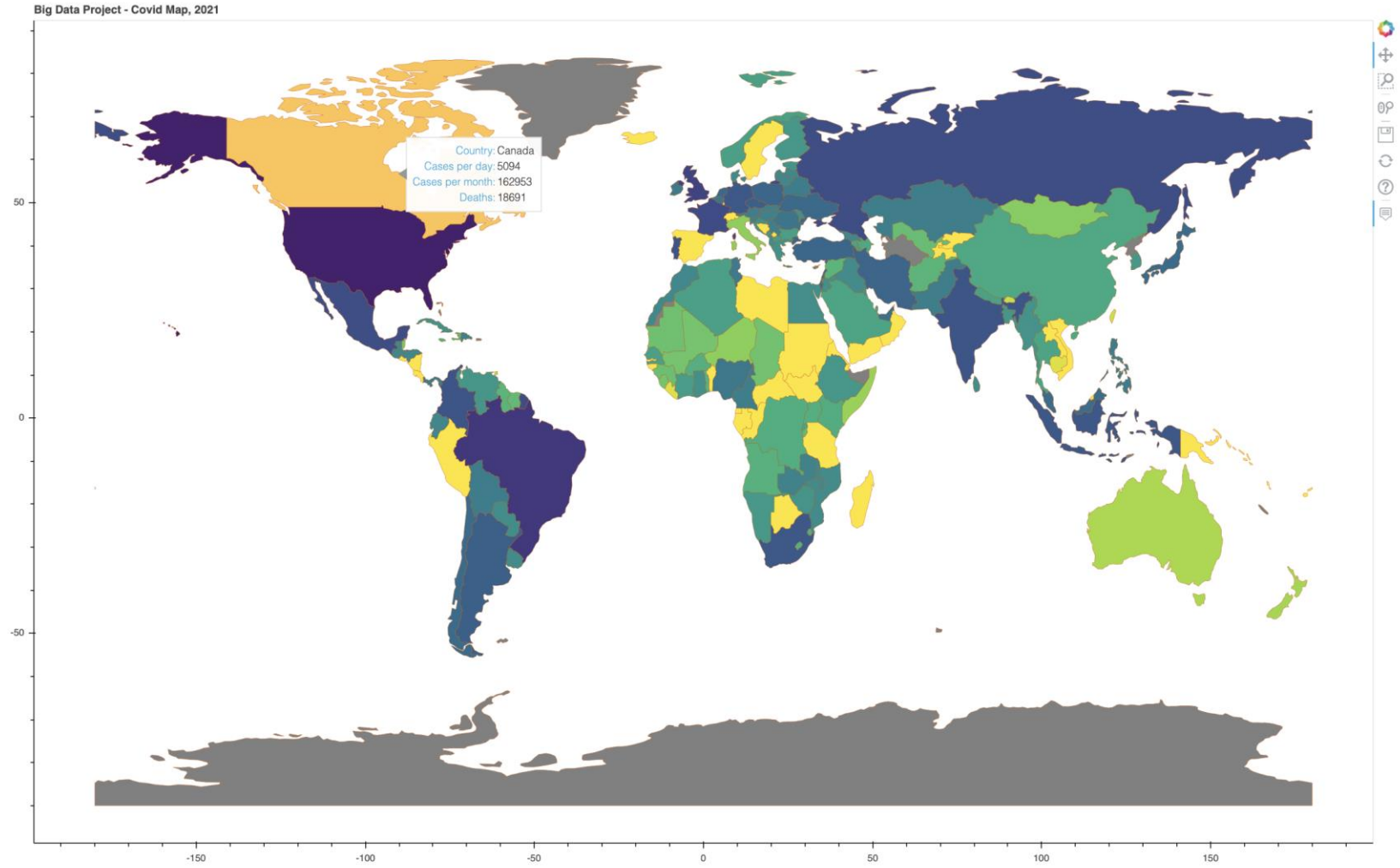
Working on Big Datasets: Use Cases

- Interactive Map for Coronavirus Cases
  - Hadoop cluster for data collection and manipulation;
  - the generated statistics are managed by a Python application that uses the Pandas and Bokeh libraries in order to be displayed to the user;
  - COVID-19 dataset:
    - is acquired from https://ourworldindata.org/coronavirus-data,
    - It is downloaded as a CSV file with columns describing the number of new cases, the number of deaths etc for each country.
  - Dataset is the parsed by the Hadoops' mapper and re-ducer and the result is a CSV file also that describes for each country the number of cases registered in the current day, the total cases registered in the current month and the total deaths registered.

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Working on Big Datasets: Use Cases

- Interactive Map for Coronavirus Cases

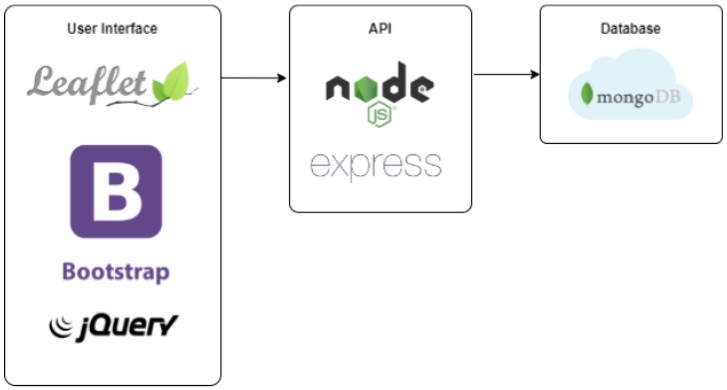


g or sets: s

ap for in New York City.

ken fror ork City - dataset by
sdark | contains crimes reported or
police in all 5 boroughs of New York City.

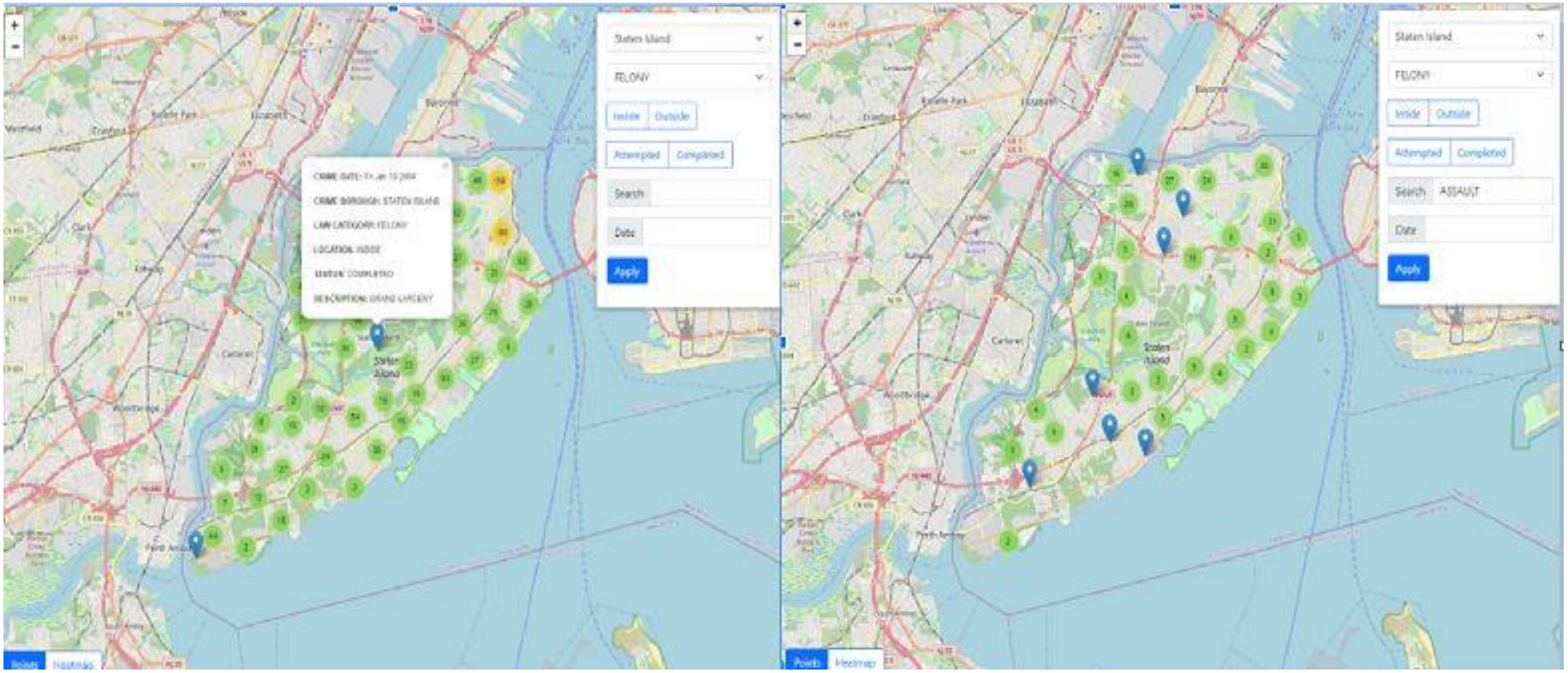
US	LAW_CATEGORY	CRIME_DISTRICT	CRIME_TYPE	REPORTED_	Latitude	Longitude	Location
	FELONY	BRONX	INSIDE	BAR/NIGHT	40.82885	-73.9167	{ type: "Point" , coordinates: [40.82884833,-73.91666114] }
	FELONY	QUEENS	OUTSIDE		40.69734	-73.7846	{ type: "Point" , coordinates: [40.69733814,-73.78455674] }
	FELONY	MANHATTAN		OTHER	40.80261	-73.9451	{ type: "Point" , coordinates: [40.80260661,-73.94505191] }
	MISDEMEANOR	QUEENS	INSIDE	RESIDENCE	40.65455	-73.7263	{ type: "Point" , coordinates: [40.65454944,-73.72633879] }
	MISDEMEANOR	MANHATTAN	FRONT OF	OTHER	40.738	-73.9879	{ type: "Point" , coordinates: [40.7380024,-73.98789129] }
	FELONY	BROOKLYN	FRONT OF	DRUG STOF	40.66502	-73.9571	{ type: "Point" , coordinates: [40.66502269,-73.95711076] }
	MISDEMEANOR	MANHATTAN	OPPOSITE OF	STREET	40.7202	-73.9887	{ type: "Point" , coordinates: [40.7202,-73.98873508] }
	FELONY	BRONX	FRONT OF	STREET	40.84571	-73.9104	{ type: "Point" , coordinates: [40.84570715,-73.91039803] }
	MISDEMEANOR	BRONX	INSIDE	RESIDENCE	40.85671	-73.8919	{ type: "Point" , coordinates: [40.85671129,-73.89189996] }
12/31/2015 PETIT LARCENY	COMPLETED	MISDEMEANOR	MANHATTAN	INSIDE	DRUG STOF	40.76562	-73.9636 { type: "Point" , coordinates: [40.76561769,-73.96362342] }
12/31/2015 PETIT LARCENY	COMPLETED	MISDEMEANOR	BRONX	INSIDE	FAST FOOD	40.82204	-73.8917 { type: "Point" , coordinates: [40.82203994,-73.89173227] }



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Working on Big Datasets: Use Cases

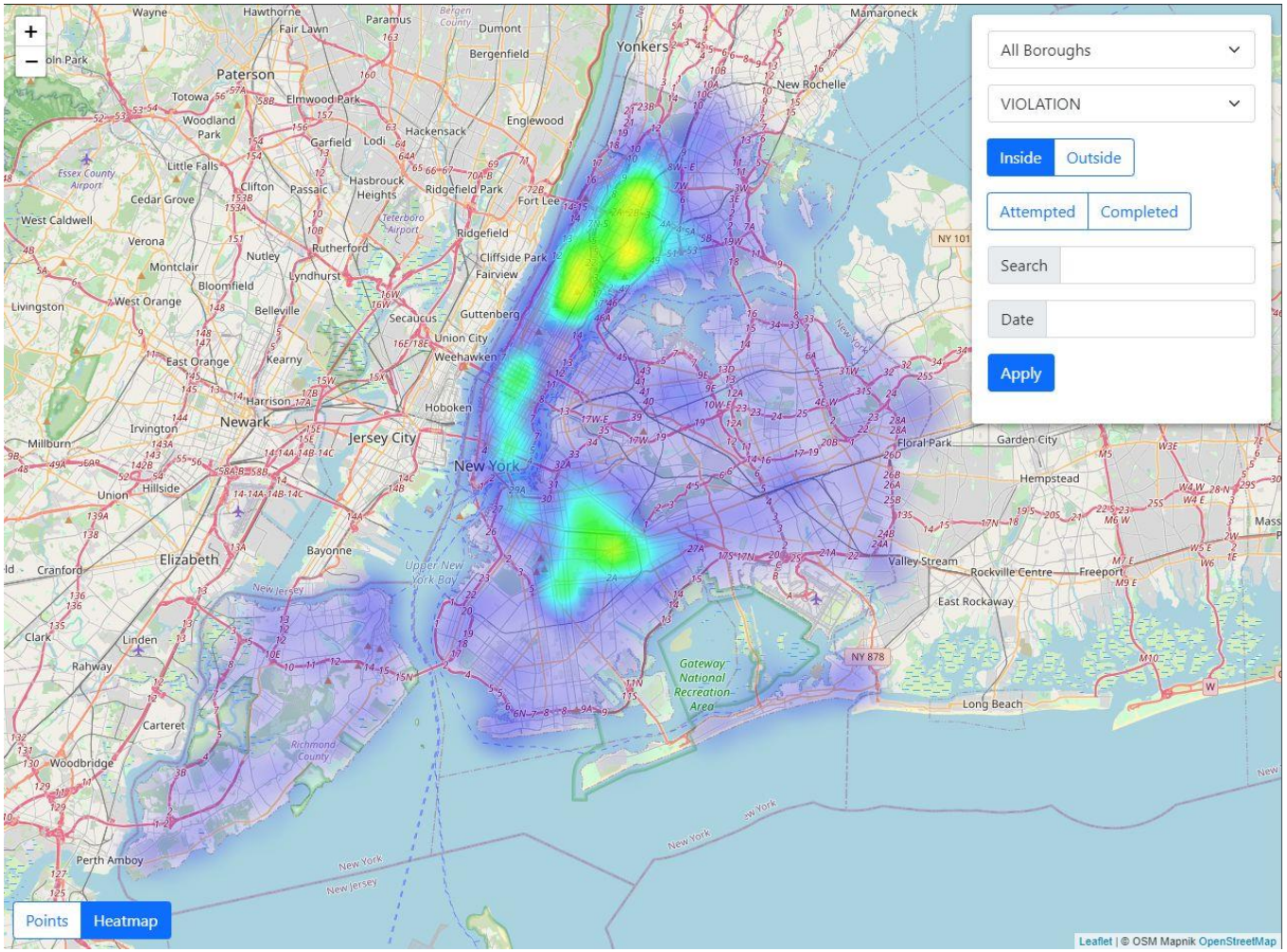
- Interactive map for analysing Crime in New York City:



FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Working on Big Datasets: Use Cases

- Interactive map for analysing Crime in New York City:



Working on Big Datasets: Use Cases

- Visualization using D3js library for COVID-19:
  - Dataset offered by Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE)
    - extracted from sources such as: World Health Organization (WHO)1 European Centre for Disease Prevention and Control (ECDC)2 and World Meters3
  - Implementation:
    - Node.js framework for creating the web server
    - MongoDB DBMS for the persistent storage of the data collected and the results of the processing tasks.
    - The MongoDB database can be launched as a Docker container, which can be afterwards accessed from the outside through the mongoose library, which provides object modelling capabilities for interfacing the database to the Node.js server.
    - Express framework to listen and respond to the clients' requests for data statistics.
    - example of aggregation is computing the worldwide report that contains the total number of detected cases, deaths and the infection rate per 1000 inhabitants up to the latest countries reports.

FACULTY OF AUTOMATIC CONTROL AND COMPUTERS

Working on Big Datasets: Use Cases

- Visualization using D3js library for COVID-19: