

MA-INF 4223 - Lab Distributed Big Data Analytics

Introduction

Dr. Hajira Jabeen, Gezim Sejdiu, Denis Lukovnikov

Summer Semester 2019

About Us



Smart Data Analytics (SDA)

- ❖ Prof. Dr. Jens Lehmann
 - Institute for Computer Science , University of Bonn
 - Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS)
 - Institute for Applied Computer Science, Leipzig
- ❖ Machine learning techniques ("analytics") for Structured knowledge ("smart data")
- ❖ The group aims at covering the full spectrum of research including theoretical foundations, algorithms, prototypes and industrial applications



Dr. Hajira Jabeen

- ❖ Group Leader “*Distributed Semantic Analytics*”
at University of Bonn, since 2016
- ❖ PostDoc at AKSW, University of Leipzig, 2015
- ❖ Assistant Lecturer at IT University of Copenhagen, 2014
- ❖ Assistant Professor, Iqra University 2010-2013
- ❖ Research Interests :
 - *Big Data, Data Mining*
 - *Semantic Web*
 - *Machine Learning and Analytics*
 - *Structured Machine learning*
 - *Optimization, Evolutionary Computation*



Gezim Sejdiu

- ❖ Research Associate/PhD Student at University of Bonn, since 2016
- ❖ Guest Researcher at AKSW group at the University of Leipzig in 2015
- ❖ System Analyst & Senior Software Developer 2008 - 2015
- ❖ Research Interests :
 - Big Data, Data Mining and Data Analysis,
 - Semantic Web and Semantic Search,
 - Machine Learning,
 - Distributed Computing



Denis Lukovnikov

- ❖ PhD Student at University of Bonn, since 2015
- ❖ Research Interests :
 - Question Answering,
 - Deep Learning,
 - Semantic Web



Group Members

- ❖ Dr. Maria Maleshkova
- ❖ Dr. Günter Kniesel
- ❖ Dr. Hamed Shariat
- ❖ Dr. Giulio Napolitano
- ❖ ...
- ❖ ...Many others



Group's Research Interests

- ❖ Distributed Semantic Analytics
- ❖ Semantic Question Answering
- ❖ Structured Machine Learning
- ❖ Software Engineering for Data Science
- ❖ Semantic Data Management
- ❖ Smart Services
- ❖ Knowledge Graph Analysis





Projects

- ❖ Big Data Europe, EU H2020, Big Data
- ❖ Boost4.0- Big Data for Industry 4.0, EU H2020
- ❖ Big Data Ocean, EU H2020
- ❖ GEISER, BMWi
- ❖ HOBBIT, EU H2020, Big Data
- ❖ SLIPO, EU H2020, Big Data
 - Scalable linking and integration
- ❖ QROWD, EU H2020, Big Data
 - Big data integration of cities e.g. geographic, transport, meteorological
- ❖ SAKE, BMWi
 - Semantic Analysis of Complex Events



Projects

- ❖ Domain Specific Languages (DSLs) for Machine Learning
- ❖ Smoothed Analysis of Structured Machine Learning Algorithms from Knowledge Graphs
- ❖ Cognitive Robotics
- ❖ Experimental Analysis of Class CS Problems
- ❖ Tensor Factorisation and Visualization for Knowledge Graphs



Software Projects

- ❖ SANSA - Distributed Semantic Analytics Stack
- ❖ AskNow - Question Answering Engine
- ❖ DL-Learner - Supervised Machine Learning in RDF / OWL
- ❖ LinkedGeoData - RDF version of OpenStreetMap
- ❖ DBpedia - Wikipedia Extraction Framework
- ❖ DeFacto - Fact Validation Framework



Distributed Semantic Analytics



- ❖ Leader: Dr. Hajira Jabeen
 - Prof. Dr. Jens Lehmann (Mentor)
 - Dr. Anisa Rula
 - Claus Stadler
 - Patrick Westphal
 - Simon Bin
 - Gezim Sejdiu
 - Harsh Thakkar
- Heba Ibrahim



Semantic Question Answering



- ❖ Leader: Dr. Giulio Napolitano
 - Prof. Dr. Jens Lehmann (Mentor)
 - Dr. Ioanna Lytra (Member / Mentor)
 - Mohnish Dubey
 - Hamid Zafar
 - Debanjan Chaudhuri
 - Konrad Höffner
 - Denis Lukovnikov
 - Gaurav Maheshwari
 - Priyansh Trivedi
 - Debayan Banerjee
 - Kuldeep Singh
 - Jewgeni Rose
 - Ashwini Jaya Kumar



Structured Machine Learning



- ❖ Leader: Prof. Dr. Jens Lehmann
 - Lorenz Bühmann (Deputy Leader)
 - Dr. Mohamed Sherif
 - Patrick Westphal
 - Simon Bin



Software Engineering for Data Science



- ❖ Leader: Dr. Günter Kniesel, Dr. Hamed S. Yazdi
 - Prof. Dr. Jens Lehmann (Mentor)
 - Dr. Luís Paulo F. Garcia
 - Dr. Tiansi Dong
 - Afshin Sadeghi
 - Shima Ibrahim



Semantic Data Management



- ❖ Leader: Dr. Christoph Lange, Dr. Steffen Lohmann
 - Prof. Dr. Jens Lehmann (Mentor)
 - Dr. Anisa Rula
 - Sahar Vahdati
 - Michael Galkin
 - Said Fathalla
 - Jean Claude Hernandez
 - Gabriel Gimenez
 - Diego Collaranra
 - Mohamed Nadjib Mami
 - Niklas Petersen
 - Lavdim Halilaj
 - Irlán Grangel-González
 - Mirette Elias

Organisational Matters



Organisational Matters

❖ Overview

- Distributed Big Data Analysis consists of two modules : lecturers/lab + project
- Mailing list : sign-up sheet
- Lecture notes will be provided at :
<http://sda.cs.uni-bonn.de/teaching/dbda/>
- Changes will be announced at the website and via mailing list



Objectives of the Lab

- At the end of the Lab we want you to be able to :
 - Program in Scala for Spark
 - Know the working model of Spark
 - Data structures and their behaviour
 - Shuffling/Cache
 - Create and execute parallel programs in Spark (Scala)
 - Use of Spark web UI for the cluster/program analysis



Outcomes

- Given an algorithm we want you to be able to :
 - Analyze the algorithm for parallel programming
 - Implement the algorithm using the distributed execution model



Lab Schedule: Thursday 10:00 - 13:00

April 4	Introduction: What is Big Data. Major tools used by data scientists. Class overview
April 11	Spark Fundamentals
April 25	Spark Fundamentals II (Spark ML), BigDL
May 02	SANSA - Semantic Analytics Stack, Project Allocation
May 16	First presentation for the Project
	Lab work
May 23, June 27	Meetings
July 5	Project report and source code submission
July 11	Final Project Presentations



Grading

- ❖ Test the proficiency in
 - Scala
 - Use of Spark / Scala (Parallel Programming) constructs
 - Complexity/efficiency of the developed algorithm
- ❖ Final Project.
 - Code,
 - Documentation,
 - Presentation

What is BigData?



Big Data



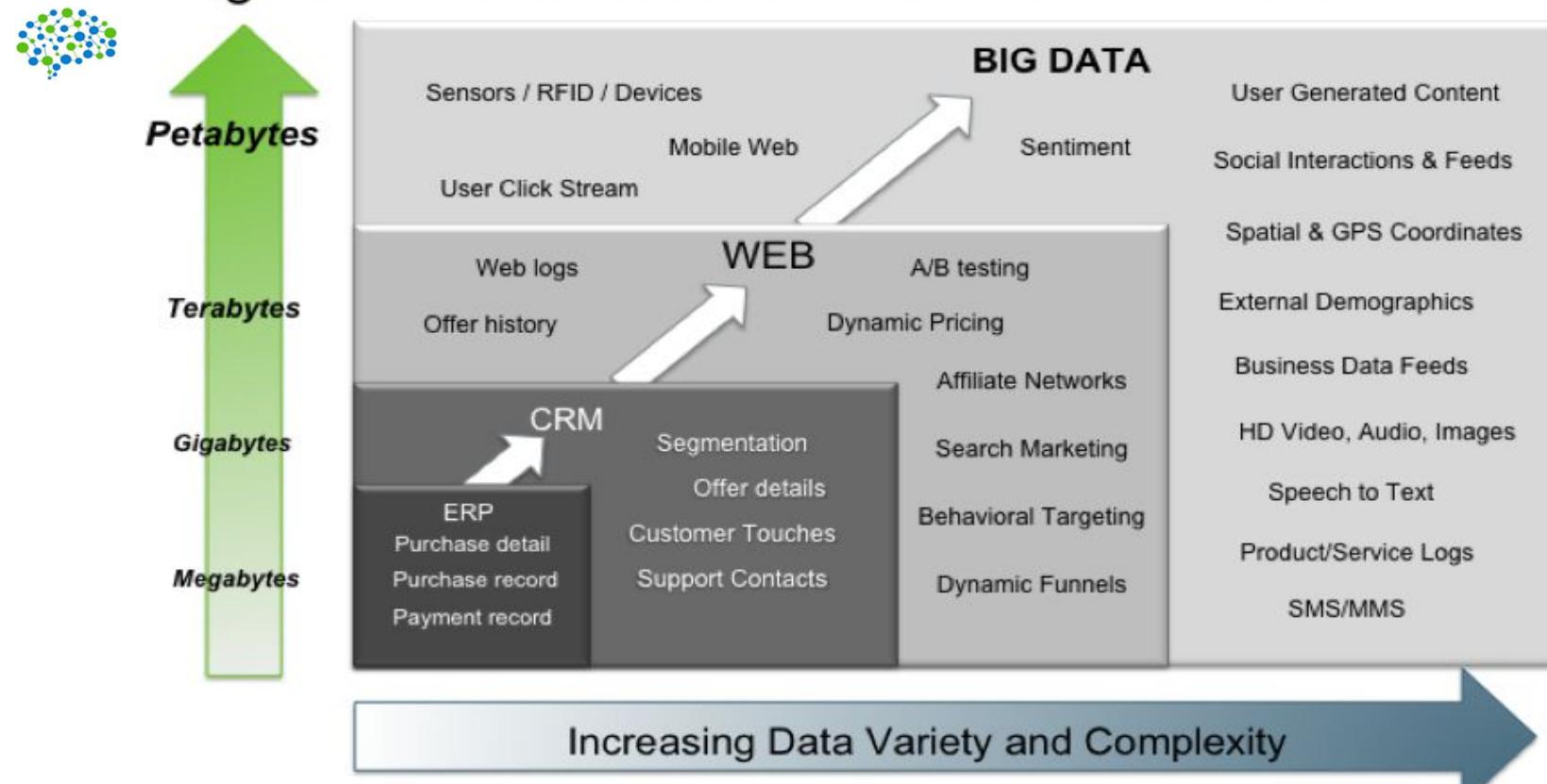
- ❖ No Single Definition
- ❖ Extremely large data sets that may be analysed computationally to reveal patterns, trends, and associations, especially relating to human behaviour and interactions
- ❖ Big data is a term for data sets that are so large or complex that traditional data processing application softwares are inadequate to deal with them



Big Data

- ❖ Every day, there are 2.5 quintillion bytes of data created - so much that 90% of the data in the world today has been created in the last two years alone
- ❖ It is not only about data collection, or data querying, its is about learning from this tremendous data for informed decision making

Big Data = Transactions + Interactions + Observations

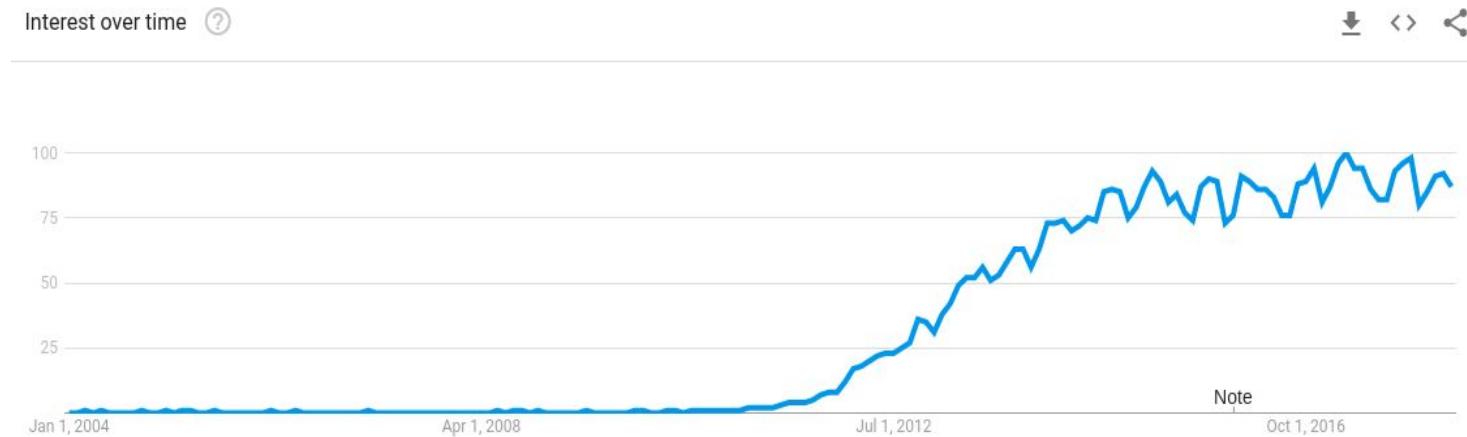


Source: <https://media.lcdn.com/mpr/mpr/AEAAQAAAAAAAATqAAAAJDg3ODgxMGRjLWUxOWItNDYxMC1hOTEzLTMwZWIxYTdjMTQ4ZA.png>



Why 'BigData' is so important?

- ❖ Its relevance is increasing drastically and Big Data Analytics is an emerging field to explore

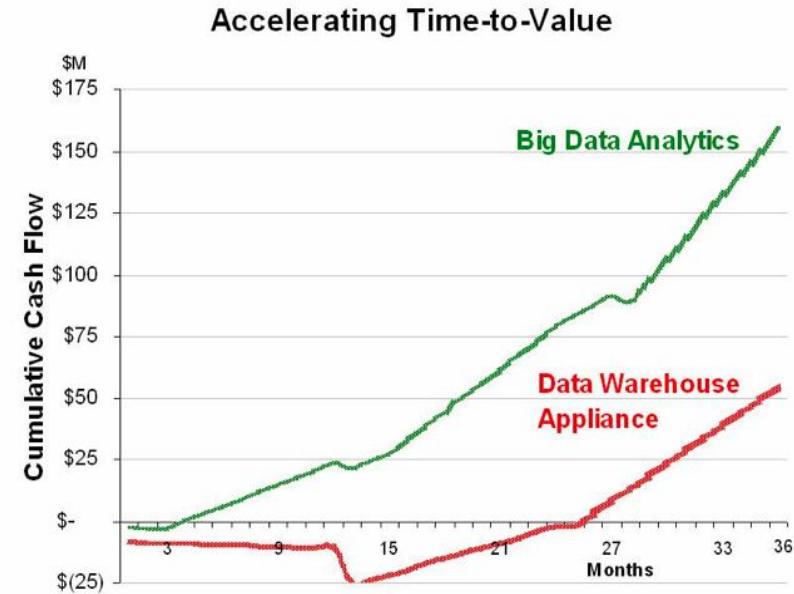


<https://www.google.com/trends/explore?date=all&q=%22big%20data%22>



Big Data Analytics

- ❖ Big data is more real-time in nature than traditional DW applications
- ❖ Traditional DW architectures (e.g. Exadata, Teradata) are not well-suited for big data apps
- ❖ Shared nothing, massively parallel processing, scale out architectures are well-suited for big data apps

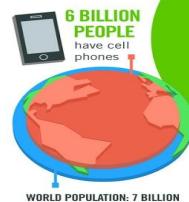


Source: http://wikibon.org/wiki/v/Enterprise_Big-data



Big Data Dimensions

40 ZETTABYTES
[43 TRILLION GIGABYTES]
of data will be created by
2020, an increase of 300
times from 2005



Volume SCALE OF DATA

It's estimated that
2.5 QUINTILLION BYTES
[2.3 TRILLION GIGABYTES]
of data are created each day

Most companies in the
U.S. have at least
100 TERABYTES
[100,000 GIGABYTES]
of data stored

The New York Stock Exchange captures
1 TB OF TRADE INFORMATION
during each trading session



Velocity ANALYSIS OF STREAMING DATA

Modern cars have close to
100 SENSORS
that monitor items such as
fuel level and tire pressure

By 2016, it is projected
there will be
18.9 BILLION NETWORK CONNECTIONS
- almost 2.5 connections
per person on earth



<http://www.ibmbigdatahub.com/infographic/four-vs-big-data>

As of 2011, the global size of data in healthcare was
estimated to be

150 EXABYTES
[161 BILLION GIGABYTES]

30 BILLION PIECES OF CONTENT
are shared on Facebook
every month



1 IN 3 BUSINESS LEADERS
don't trust the information
they use to make decisions

27% OF RESPONDENTS

in one survey were unsure of
how much of their data was
inaccurate

Variety DIFFERENT FORMS OF DATA

By 2014, it's anticipated
there will be

**420 MILLION WEARABLE, WIRELESS
HEALTH MONITORS**

4 BILLION+ HOURS OF VIDEO
are watched on
YouTube each month



400 MILLION TWEETS
are sent per day by about 200
million monthly active users

Poor data quality costs the US
economy around
\$3.1 TRILLION A YEAR



Veracity UNCERTAINTY OF DATA

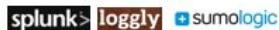
Big Data Landscape



Vertical Apps



Log Data Apps



Ad/Media Apps



Business Intelligence



Analytics and Visualization



Data As A Service



Analytics Infrastructure



Operational Infrastructure



Infrastructure As A Service



Structured Databases



Technologies

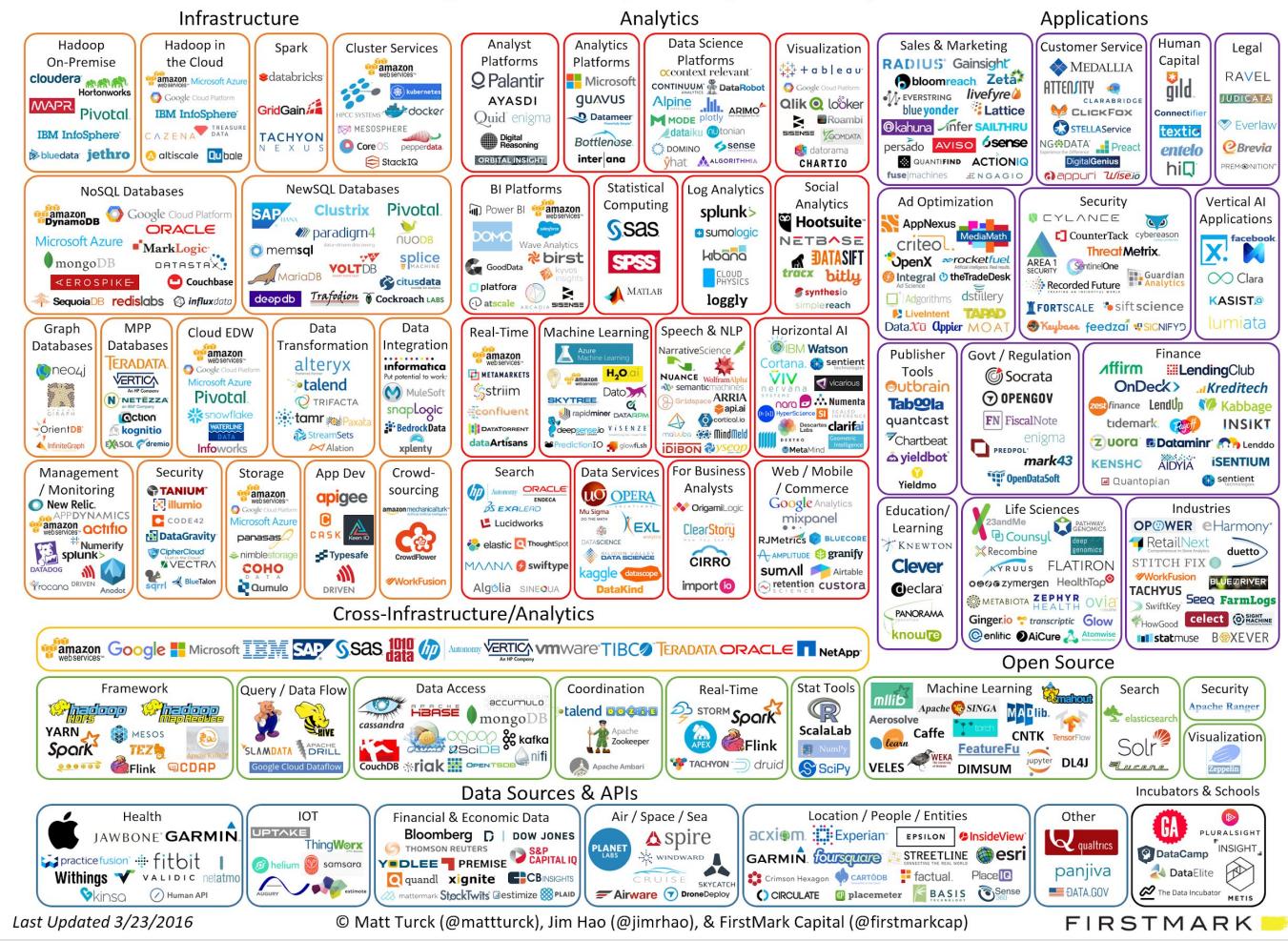


Big Data Landscape (Version 2.0)





Zoo



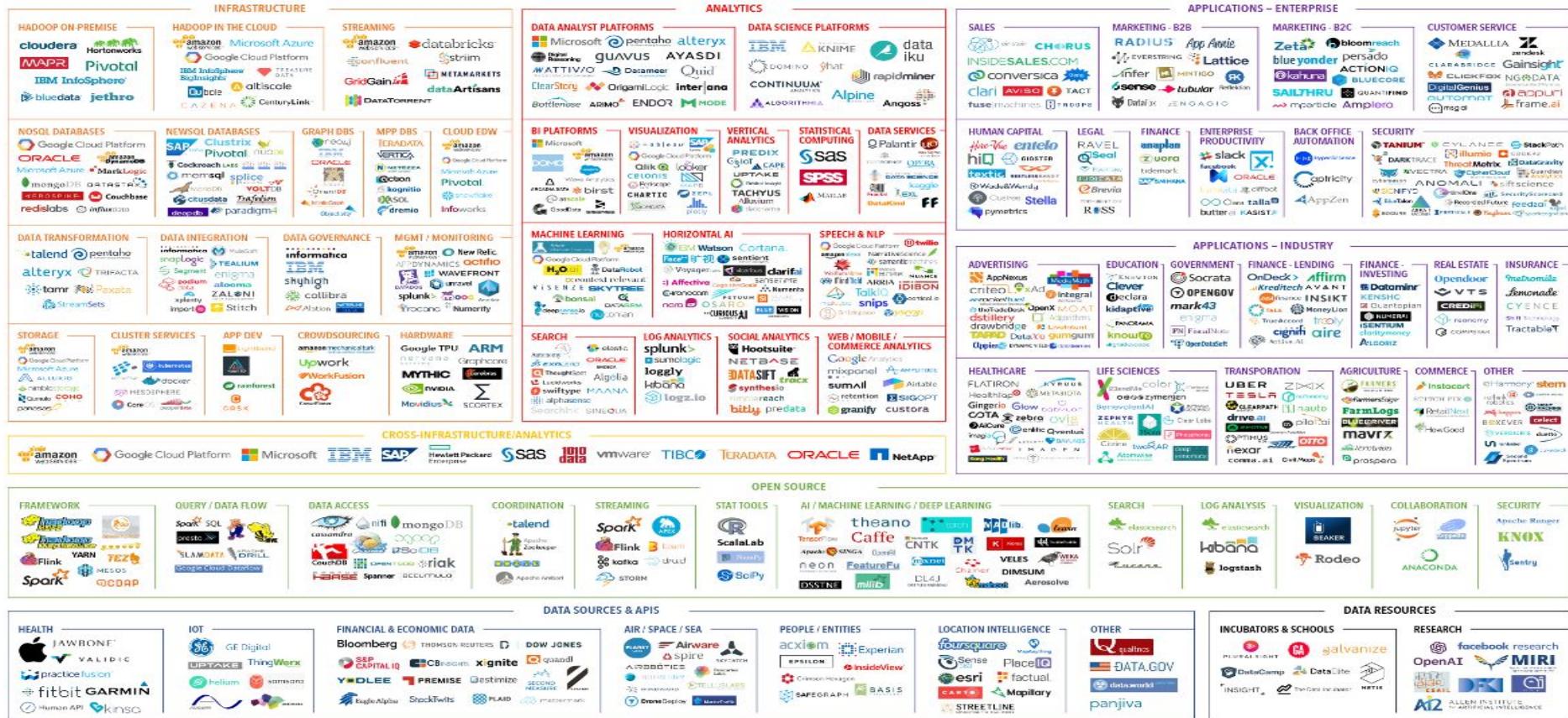
Source

Last Updated 3/23/2016

© Matt Turck (@mattturck), Jim Hao (@iimrhao), & FirstMark Capital (@firstmarkcap)

FIRSTMARK

BIG DATA LANDSCAPE 2017



Last updated 4/5/2017

BIG DATA & AI LANDSCAPE 2018



Final 2018 version, updated 07/15/2018

© Matt Turck (@mattturck), Demi Obavomi (@demi_ obavomi), & FirstMark (@firstmarkcap)

mattturck.com/bigdata2018



Big Data Ecosystem

File system	HDFS, NFS
Resource manager	Mesos, Yarn
Coordination	Zookeeper
Data Acquisition	Apache Flume, Apache Sqoop
Data Stores	MongoDB, Cassandra, Hbase, Project Voldemort
Data Processing	<ul style="list-style-type: none">• Frameworks Hadoop MapReduce, Apache Spark, Apache Storm, Apache FLink• Tools Apache Pig, Apache Hive• Libraries SparkR, Apache Mahout, MLlib, etc
Data Integration	<ul style="list-style-type: none">• Message Passing Apache Kafka• Managing data heterogeneity SemaGrow, Strabon
Operational Frameworks	<ul style="list-style-type: none">• Monitoring Apache Ambari



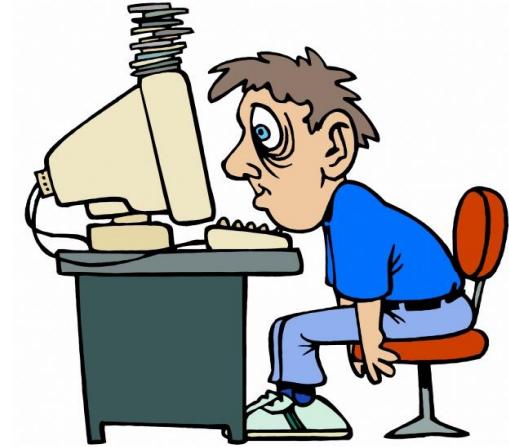
Data Flow Engines

- ❖ The Data flow engines are useful because
 - They save time and effort on the part of the programmer by providing abstractions
 - They scale well
 - Many algorithms have been redesigned to fit into the paradigm
 - They have become common on clusters



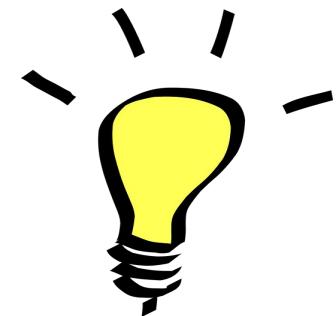
Data Flow Models

- ❖ Data flow vs. traditional network programming
 - Message Passing Interface
 - Programmer managed data locality and Code
 - Hardware failure
 - Slow Hardware
 - Data Communication is slow over the network





Map reduce



- ❖ First popular data flow model comprises the programmer's ability to control functionality in order to handle the problems
- ❖ In the Map-Reduce model, the user provides two functions (map and reduce)
 - Map() must output key-value pairs, and as a result
 - reduce() is guaranteed that its input is partitioned by key across machines



Drawbacks of Mapreduce

- ❖ Force data analysis workflow into a map and a reduce phase
 - You might need
 - Join
 - Filter
 - Sample
 - Complex workflows that do not fit into map/Reduce
 - Mapreduce relies on reading data from disk
 - Performance bottleneck
 - Especially for iterative algorithms



Solution:

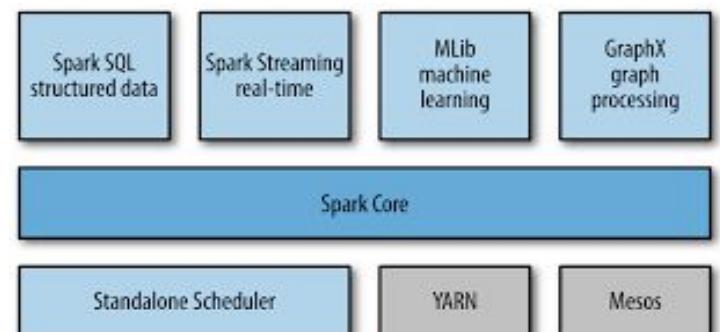
- ❖ A tool that works in the same environment
- ❖ Provides an Interactive shell
- ❖ Compatible with the existing environment
- ❖ No need to replace the stack, but replace map--reduce

Big Data distributed in-memory frameworks



Apache Spark

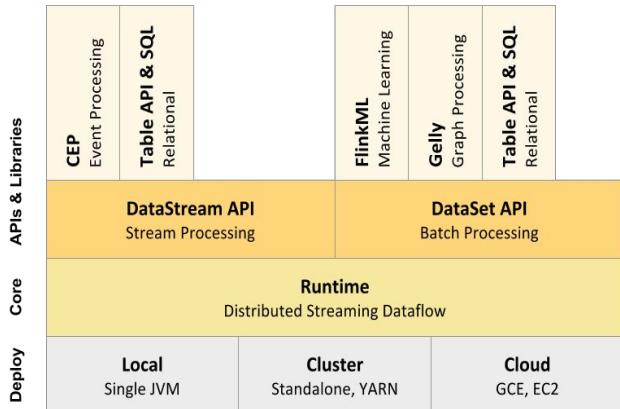
- ❖ **Apache Spark** is an open-source distributed and highly scalable in-memory data processing and analytics system. It provides APIs in Scala, Java, Python and R which try to simplify the programming complexity by introducing the abstraction of Resilient Distributed Datasets (RDD), i.e. a logical collection of data partitioned across machines.
- ❖ On top of its core, Spark provides 4 libraries:
 - **Spark SQL** for SQL and structured data processing
 - **Spark Streaming** stream processing
 - **MLlib** machine learning algorithms
 - **GraphX** graph processing.





Apache Flink

- ❖ Apache Flink is an open-source stream processing framework for distributed, high-performing, always-available, and accurate data streaming applications
- ❖ Flink provides:
 - [DataStream API](#) for unbounded streams
 - [DataSet API](#) for static data
 - [Table API & SQL](#) with a SQL-like expression
- ❖ It bundles libraries for domain-specific use cases:
 - [CEP](#), a complex event processing library,
 - [Machine Learning library](#), and
 - [Gelly](#), a graph processing API and library



Big Data Projects

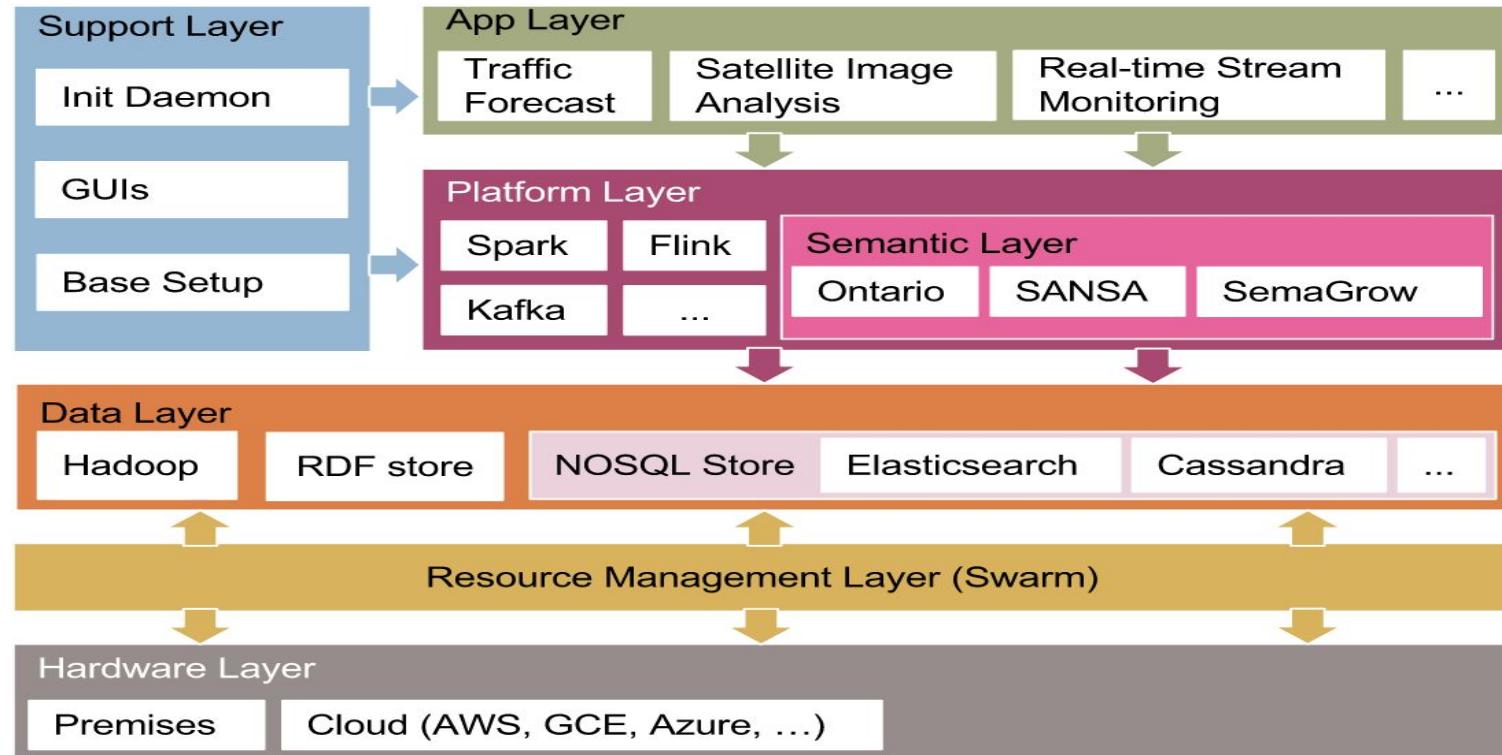


Big Data Projects

- ❖ Related to this lab
 - Big Data Europe
 - SANSA
 - Big Data Ocean
 - Boost4.0



BDE Platform





Key Observation From BDE

- ❖ Heterogeneity AKA Variety

A Single View to the Customer

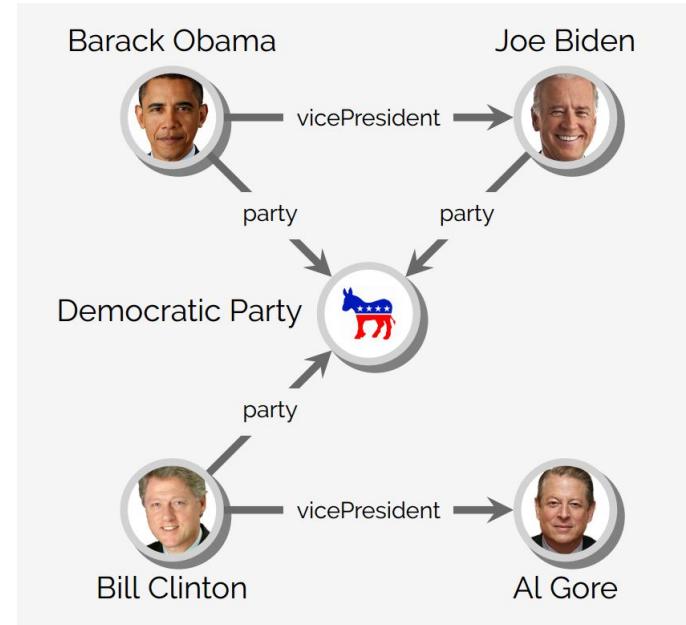


Smart Big Data



Knowledge Graphs

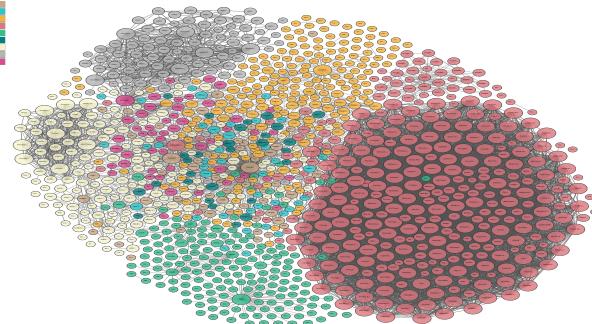
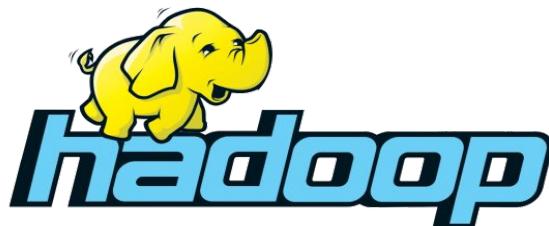
- ❖ Modelling entities and their relationships
- ❖ Analysis: finding underlying structure of graph e.g. to predict unknown relationships
- ❖ Examples: Google Knowledge Graph, DBpedia, Facebook, YAGO, Twitter, LinkedIn, MS Academic Graph, WikiData





Smart Big Data

- ❖ Over the last years, the size of the Semantic Web has increased and several large-scale datasets were published



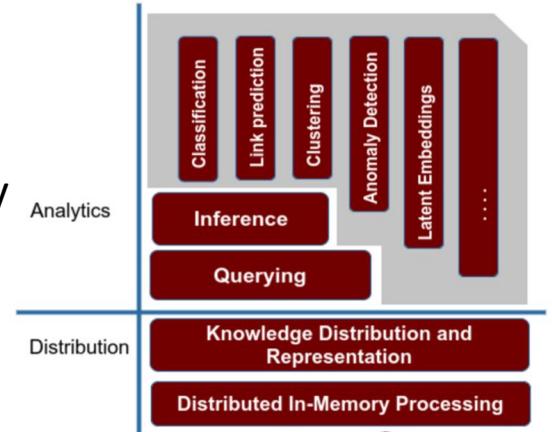
Source: LOD-Cloud (<http://lod-cloud.net/>)

- ❖ Hadoop ecosystem has become a standard for Big Data applications → use this infrastructure for Semantic Web as well



SANSA - Semantic Analytics Stack

- ❖ Knowledge Graphs become increasingly popular (via graph databases and semantic technologies)
- ❖ But:
 - Most ML algorithms work on simple feature input (not graphs)
 - Advanced algorithms for knowledge graphs usually do not scale horizontally
- ❖ **SANSA** is a suite of APIs for distributed reading, querying, inferencing and analysis of RDF knowledge graphs



APACHE
 Spark Flink

<http://sansa-stack.net/>

Introduction to Scala



- ❖ A functional Programming Language that follows a rich concise syntax
- ❖ Multiparadigm (you can also write OO code)
- ❖ Interoperates with Java
- ❖ Simplifies the concurrent programming , (No threads/Locks)
- ❖ Static typing, Immutable objects, Closure, elegant pattern matching
- ❖ Strongly Typed language, with functional and concurrency support



- ❖ Conciseness (100 lines in Java = 10-15 lines of code in scala)
- ❖ Sometimes harder to understand
 - Currying,
 - Function passing,
 - High order functions



- ❖ Everything is an object
 - Primitive types e.g. numbers, bool
 - Functions
- ❖ Numbers are objects:
 - $1+2*3 \rightarrow (1).+(2).* (3)$
- ❖ Functions are objects:
 - Pass functions as arguments
 - Store them in variables
 - Return them from other functions
- ❖ Function declaration
 - Def functionName ([list of parameters]:[return type])



Scala REPL

- ❖ Read evaluate and Print Loop.
- ❖ Interactive shell session,
- ❖ In interactive mode, the REPL reads expressions at the prompt, wraps them in an executable template, and then compiles and executes the result.



Basics

```
scala> val msg = "Hello, world!"  
msg: String = Hello, world!  
scala> println(msg)  
Hello, world!  
scala> def max(x: Int, y: Int): Int  
= if (x < y) y else x  
max: (x: Int, y: Int)Int
```

```
scala> max(3, 5)  
res1: Int = 5
```

<http://www.artima.com/scalazine/articles/steps.html>



Anonymous Functions

```
>val plusOne = (x: Int) => x + 1
>(x: Int) => x * x * x // returns cube
>(x: Int, y: Int) => x + y // sums two numbers
    //Alternatively
{
  def f(x: Int, y: Int) = x + y;
}
printIn(plusOne(0)) // Prints: 1
anotherFunction(plusOne(20)) // Prints: 21
```



High Order Functions

- ❖ First order functions :
 - Acts on simple data types
- ❖ High order Function:
 - Act on other functions
 - Sum function in the next example is a High order function
 - Functions are treated as first-class values
 - passed as parameters
 - returned as a value
 - Flexible way to compose programs



High order Functions

```
def id(x:int):int = x
```

```
def cube (x:int):int = x*x*x
```



High order Functions

```
def sum(f:int => int, a:int, b:int) : int =  
    if(a>b) 0  
    else f(a) + sum(f, a+1, b)
```



High order Functions

```
def sumInts(a:Int, b:Int) = sum( id, a, b)
```

```
def sumCubes(a:Int, b:Int) = sum(cube, a, b)
```

sumInts(2,3) = 5

sumCubes(2,3) = 35



Anonymous functions

Assignment:

- Use anonymous functions in the previous example



Anonymous functions

Assignment:

- **val** sumInts(a:Int, b:Int) = sum($x \Rightarrow x$, a, b)
- **val** sumInts(a:Int, b:Int) = sum($x \Rightarrow x * x * x$, a, b)



High order Functions and Tail recursion

```
def sumInts(a: Int, b: Int): Int =  
    if (a > b) 0 else a + sumInts(a + 1, b)  
  
def sum(f:Int => Int, a:Int, b:Int) = { // f is a function  
def loop(a:Int, acc:Int):Int =  
    if(a>b) acc  
    else loop(a+1, f(a) + acc)  
        loop(a, 0) // return value  
}  
sum(x =>x*x, 3, 5) // 50
```



References

- ❖ "Linking Open Data cloud diagram 2017, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak.
<http://lod-cloud.net/>".
- ❖ <http://www.artima.com/scalazine/articles/steps.html>
- ❖ <http://sda.cs.uni-bonn.de/>
- ❖ <https://github.com/SANSA-Stack>
- ❖ <https://github.com/big-data-europe>
- ❖ <https://github.com/SmartDataAnalytics>

THANK YOU !

< <http://sda.cs.uni-bonn.de/teaching/dbda/> >



Dr. Hajira Jabeen
jabeen@cs.uni-bonn.de
Room 1.062
(Appointment per e-mail)



Gezim Sejdiu
sejdiu@cs.uni-bonn.de
Room 1.068
(Appointment per e-mail)



Denis Lukovnikov
lukovnikov@cs.uni-bonn.de
Room 1.062
(Appointment per e-mail)