Week06 - Summary

Big Data and Ethics

How to analyse Data on such a large scale?

- Big Data is a challenge for the data engineering side of Data Science
  - We will look at some approaches on how to solve these challenges
- Big Data is also a challenge for data privacy, security, and ethics

Big Data: Volume

- Very relative due to Moore's Law
  - What once was considered big data, is considered a main-memory problem nowadays
  - e.g. Excel: In 2003 max 65000 rows, now max 1 million rows, still…

Big Data: Velocity

- Conventional scientific research:
  - months to gather data from 100s cases, weeks to analyse the data and years to publish
  - Example: Iris flower data set by Edgar Anderson and Ronal Fisher from 1936
- On the other end of the scale: Twitter
  - average 6000 tweets/sec, 500 million per day over 200 billion per year

Big Data: Variety

- Structured Data, such as csv or RDBMS
- Semi-structured Data, such as JSON or XML

- Unstructured Data, i.e. text, emails, images, video
  - An estimated 80% of enterprise data is unstructured

→ **Variety is the biggest challenge in Big Data**

Sources of Big Data/More Vs

- Human-generated Big Data
  - E.g., photos, posts, likes, etc
- Machine-generated data
  - Communication logs, Internet-of-Things, etc

- More Vs of Big Data:
  - Validity (data quality), Variability (data consistency), Veracity (data accuracy/trustworthiness), Value…

Big Data challenges beyond technical aspects

- **Data Privacy**
  - Some data sources, such as “Internet-of-Things”, allow tracking anyone
    - Do you really need to know who was travelling a route in order to predict, e.g., traffic densities?
    - Personal data can be inferred sometimes ⇒ New York Taxis data set example
  - Privacy laws
    - Always check: Are you allowed to use some data or process is anywhere?
    - Some personal data, especially regarding health or tax, is specially protected; e.g., not allowed to leave a jurisdictive area
- **Data Security**
  - Can your users trust you to keep their data safe?

- Big data can expose your organisation to serious privacy and security attacks!

- **Data Discrimination**
  - Is it acceptable to discriminate against people based on data on their lives?
  - Credit card scoring? Health insurance?
- Check:
  - Are you working on a representative sample of users/consumers?
  - Do your algorithms prioritise fairness? Aware of the biases in the data?
  - Check your Big Data outcomes against traditionally applied statistics practices

Analysing Big Data

Case for Data Science Platforms

- Data is either
  - too large (volume)
  - too fast (velocity), or
  - needs to be combined from diverse sources (variety) for processing with scripts or on single server
- Need for:
  - scalable platform
  - processing abstractions

Scale-Up

- The traditional approach:
  - To scale with increasing load, buy more powerful, larger hardware
    - from single workstation
    - to dedicated db server

        - to large massive-parallel database appliance

The Alternative: Scale-Out

- A single server has limits…
- For real Big Data processing, need to **scale-out** to a cluster of multiple servers (nodes)

Challenges

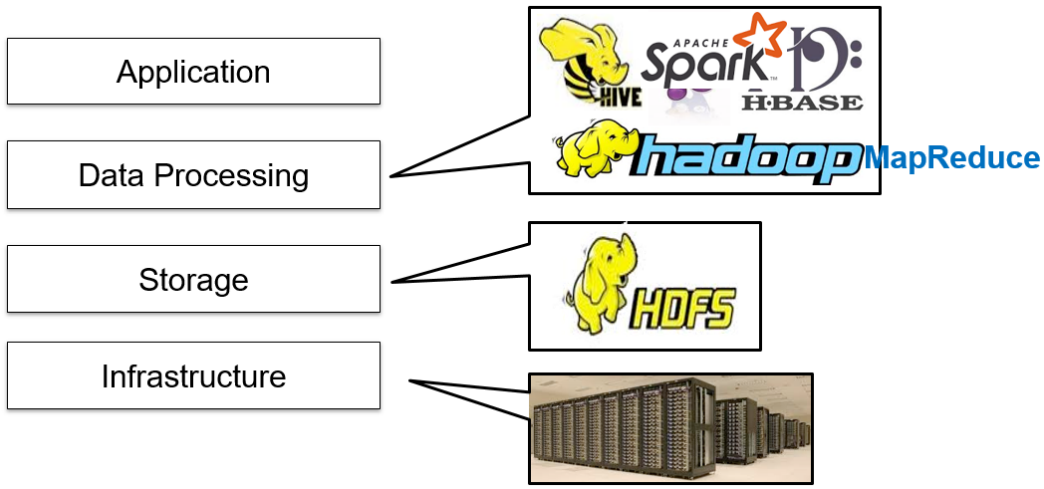
- **Scale-Agnostic Data Management**
    - **sharding** for performance
    - **replication** for availability
    - ideally such that applications are unaware of underlying complexities

- **Scale-Agnostic Data Processing**
    - Nowadays we collect massive amounts of data; how can we analyse it?
        - Answer: use lots of machines…
    - Performances: parallel processing
    - Availability: Ideally, the system never down; can handle failures transparent

      ⇒ Distributed Data Science Platforms

Distributed Data Science Platforms

Big Data Analytics Stack



Distributed Data Analytics Frameworks

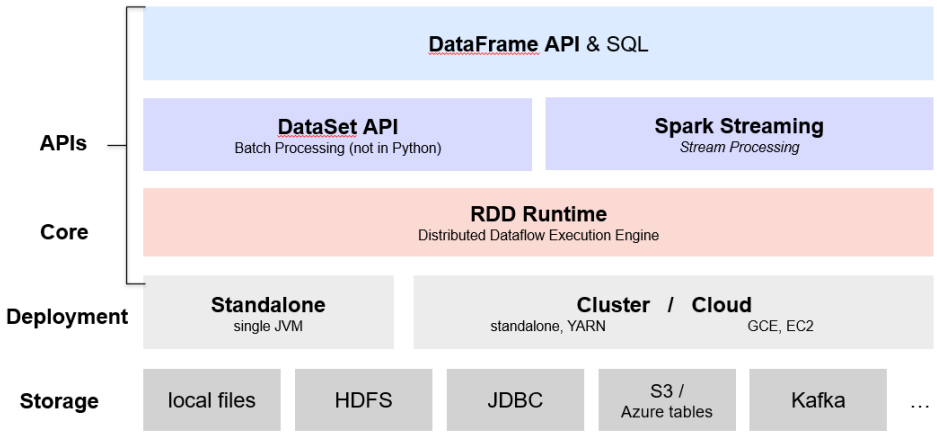
- **Apache Hadoop**
  - Open-source implementation of original MapReduce from Google; Apache top-level project
  - Java framework, but also provides a Python interface nowadays
  - Parts: own distributed file system (HDFS), job scheduler (YARN), MR framework (Hadoop)
- **Apache Spark**
  - Distributed cluster computing framework on top of HDFS/YARN
  - Concentrates on **main-memory** processing and more **high-level data flow control**
  - Originates from research project from UC Berkeley
- **Apache Flink**
  - Efficient data flow runtime on top of HDFS/YARN
  - Similar to Spark, but more emphasise on **build-in dataflow optimiser** and **pipelined processing**
  - Strong for data stream processing

- **Apache Hive**
  - Provides an SQL-like interface on top of Hadoop/HDFS
  - Allows to define a relational schema on top of HDFS files, and to query and analyse data with HIVEQL (SQL dialect)
  - Queries automatically translated to MR jobs and executed in parallel in cluster
  - Example: WordCount in HIVE

```
CREATE TABLE docs (line STRING);
LOAD DATA INPATH 'input_file' OVERWRITE INTO TABLE docs;
CREATE TABLE word_counts AS
  SELECT word, count(1) AS count
    FROM (SELECT explode(split(line, '\s')) AS word FROM docs) temp
  GROUP BY word
  ORDER BY word;
```

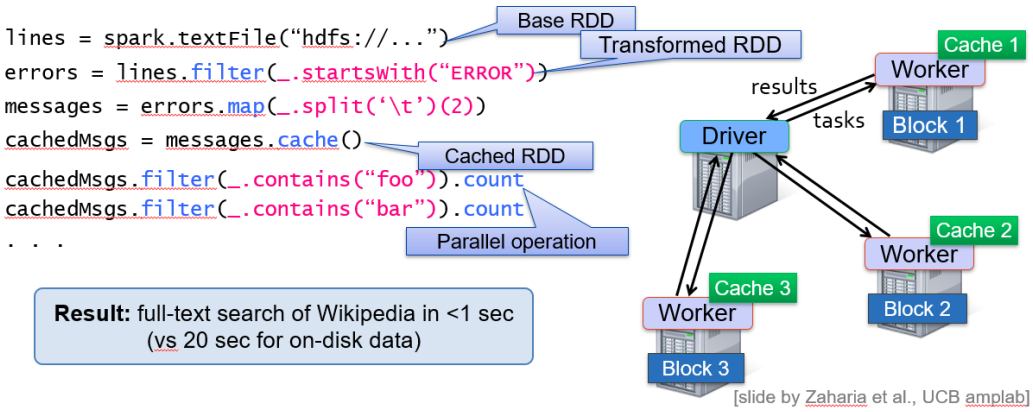
- **Many more high-level frameworks for advanced data analytics**

Example Apache Spark System Stack



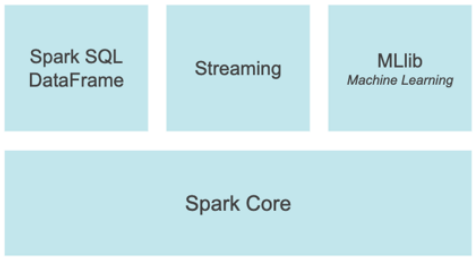
Example: Log Mining with Apache Spark

- Load error messages from a log into memory, then interactively search for various patterns



PySpark

- PySpark allows you to write Spark applications using Python APIs, but also provides the PySpark shell for interactively analysing data in a distributed environment
- PySpark supports most of Spark's features such as Spark SQL, DataFrame, Streaming, MLib (Machine Learning) and Spark Core



Spark and Jupyter Notebooks

- PySpark can also be used from Jupyter notebooks
- Either local install of FindSpark package

  install findspark, start pyspark, start Jupyter

```
import findspark
findspark.init()
import pyspark.sql
```

```
spark = SparkSession.builder.appName("...")...
...
```

- Or use in a cloud platform such as **Databricks**

Tips and Tricks

- **Big Data** is one driver behind Data Science; definition somewhat general though
- Map/Reduce paradigm very powerful to tackle the petabyte scale problems of today's majors
  - Especially for big internet companies such as Google, Amazon, Linkedin, Twitter, Facebook,…
- Pros:
  - Scalability and runs on commodity hardware
- Cons:
  - Not usable for non-programmers
  - Even non-procedural programmers struggle with the functional nature of Map/Reduce
  - Not everything is petabyte scale…

- Dataflow systems such as Spark and Flink improve the usability quite a bit another approach available is **HIVE** as 'SQL on MapReduce'
  - But still targets petabyte scale problems

Product Thinking

Define a Problem before the Solution

First define the problem…

User problem: What problem do we solve?

Target audience: For whom are we doing this?

Vision: Why are we doing this?

Strategy: How are we doing this?
Goals: What do we want to achieve?

Only then does it make sense to think about the solution

Relationship of Data Scientist to Product

Model 1: Data scientist as an **owner**

Model 2: Data scientist as a **service**

Model 3: Data scientist as a **partner**

Data Scientist as Owner of Product

Operates in a “hacky way” (early stage companies)

Key steps in business rely heavily on data

- Recommender
- Relevance
- Matching
- Scoring

Mostly backend, relatively stand-alone features

Data Scientist as a Service

Engagement is “on-demand”, project-based

Examples:

- some of the BI roles
- strategy roles (consultancy)
- data API for product
- modeling for specific purposes: propensity to {x}, where x:

  {buy, attrite, convert, etc}

Data Scientist as a Partner

- Plays active role in every stage of Product Life Cycle
- Shares the ultimate goal of product success
- Often requires an embedded engagement model

The Ethical Data Scientist

Why is it a Data Scientist's Job?

Consider:

- User behaviour data forms the foundation of data products
- Products assist users but may also influence their behaviour
- e.g., ranking algorithms, recommendation systems, friend suggestions

Models/algorithms not only predict but affect the future

- This is both incredibly exciting and absolutely terrifying

Example 1: Preventative Policing

- Chicago police used predictive modelling to create a heat list
  - Given social network from arrest records, geographic, temporal data
  - Pre-emptively approach:

    Predict whether a person is likely to be involved in violent crime

Questions of Ethics: Preventative Policing

- How avoid perpetrating potentially unfair or damaging stereotypes/profiling present in the data?
- How use information positively and manage potential prediction mistakes?

Example 2: User Tweaking

Inducing emotional states

- A 2014 study explored whether user mood is contagious on Facebook
- Manipulated feeds to include fewer positive or negative posts
- Discuss risks: How should users be protected?

Machine Learning and Human Bias

Scenario: Optimise services to homeless families

- Imagine our goal is to match homeless families with the most appropriate services
- We have historical data with various characteristics:

  number and age of children and parents, zip code, number and length of previous stays in homeless services, race

Which characteristics should we use?

What about Race?

- Now imagine we find that including race makes the model more accurate
- Should we use it?
- Remember algorithm output will be used to help pair families with services

- Using historical data means that we are “training our model” on data that is surely biased, given a history of racism
- An algorithm cannot see the difference between patterns that are based on injustice and patterns that are based on traffic
- So choosing race as a characteristic in our model would be unethical