Introduction to Pyspark
with good data engineering practices

by Oliver Willekens,
data engineer and instructor
at Data Minded

dataminded

Course taught online for KBC, Belgium
between 2020-10-12 and 2020-10-23

Introductions

1. **your full name**

   Oliver Willekens

2. **your background (keep it high level) (e.g. “I have a background in social sciences”)**

   Physics engineering.

3. **number of years you've been using Python**

   About 9. Four of those with Spark.

4. **What do you hope to get out of this training? Why are you here?**

   I’m here to help you. Teach tricks. Introduce software engineering practices.

5. **A specific question or problem you would like to see addressed.**

   ~~Finding the sweet spot between the advanced/intermediate users and the starters.~~ → entry tests

   Finding a good way of working for remote teaching with small groups.

Today’s agenda

- Theory
  - Hadoop
  - Spark
    - Spark Stack
    - Spark inter process communication
    - The DataFrame API
- Practice
  - Working with virtual environments and Pycharm

Hadoop is an ecosystem designed to deal with data across cluster nodes. It is built on top of 4 components.

- Hadoop Common
- Hadoop Distributed File System (HDFS)
- Hadoop YARN
- Hadoop MapReduce

“Ecosystem” is pretty apt:

Fun fat: Hadoop got its name from one of the main developers’s son. The two year old had a stuffed animal - a yellow elephant, which he called Hadoop.

Doug Cutting, with “Hadoop”

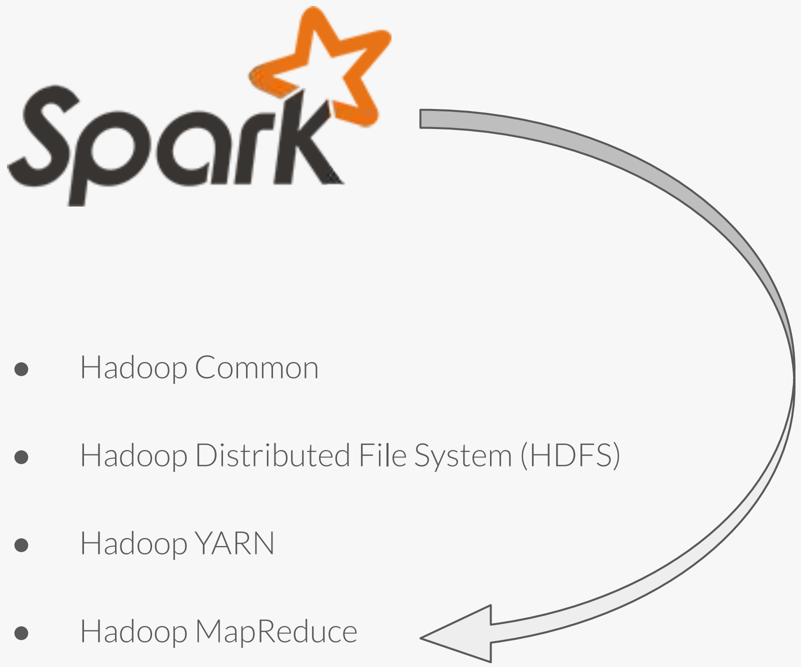
The main concepts behind Hadoop MapReduce can be explained with a deck of cards

Classroom Experiment: need 2 volunteers and a shuffled deck of cards.

Simulate the computation of finding the largest card value per suit, assuming that non-numbered cards are “bad”.

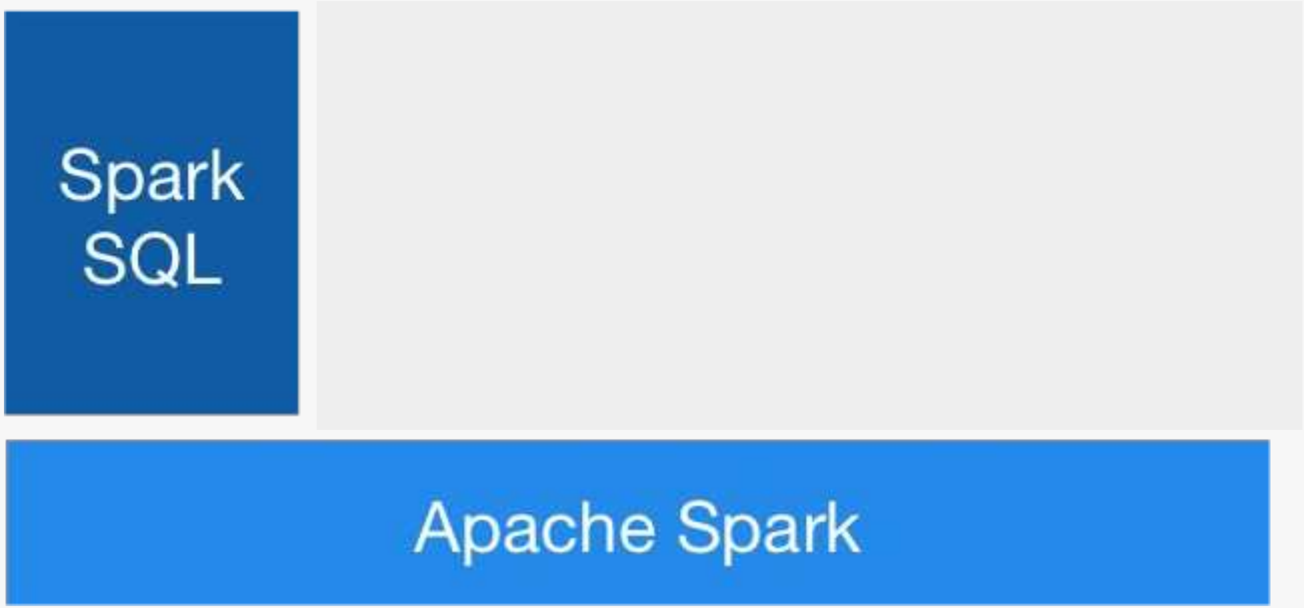
Explain terms like node, process, shuffle, map and reduce. Master/worker.

Apache Spark does not replace all of Hadoop. Instead, it replaces Hadoop MapReduce. It integrates well with YARN and HDFS.



- Hadoop Common
- Hadoop Distributed File System (HDFS)
- Hadoop YARN
- Hadoop MapReduce

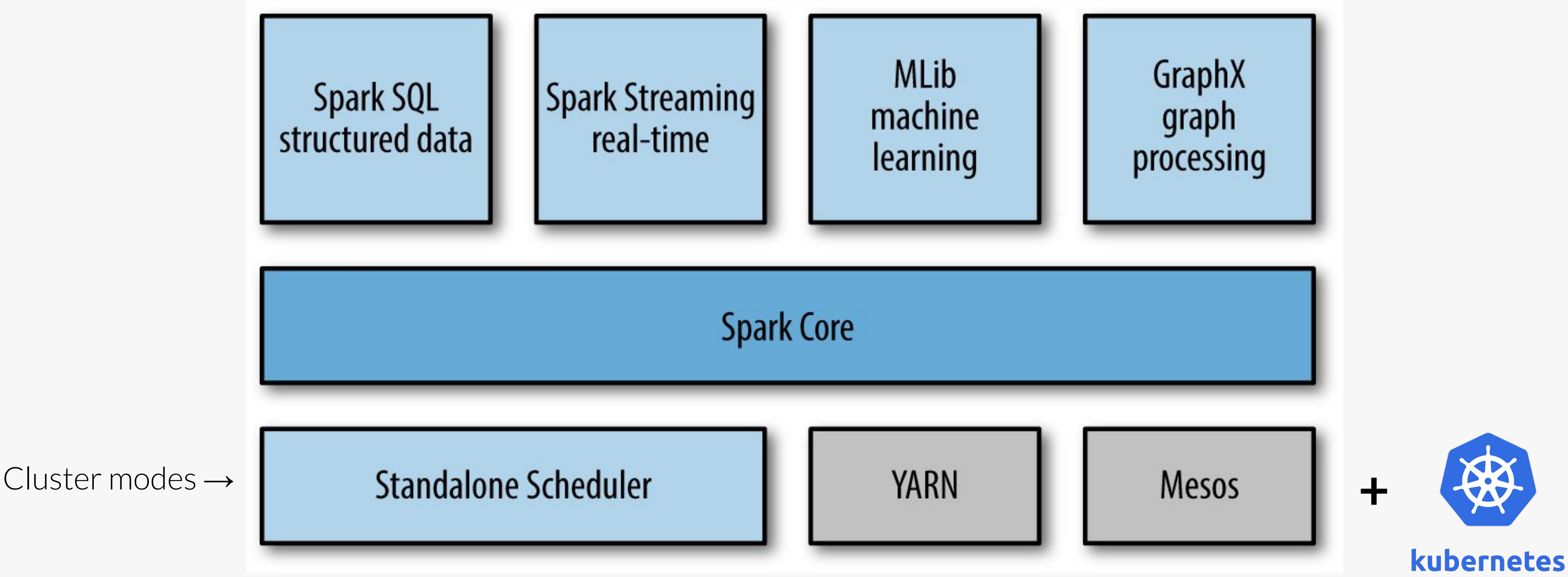
The Spark Stack consists of 4 modules, one common component and a set of operators that allow integrating with resource managers



Spark SQL

Apache Spark

The Spark Stack consists of 4 modules, one common component and a set of operators that allow integrating with resource managers



Spark SQL structured data

Spark Streaming real-time

MLib machine learning

GraphX graph processing

Spark Core

Cluster modes →

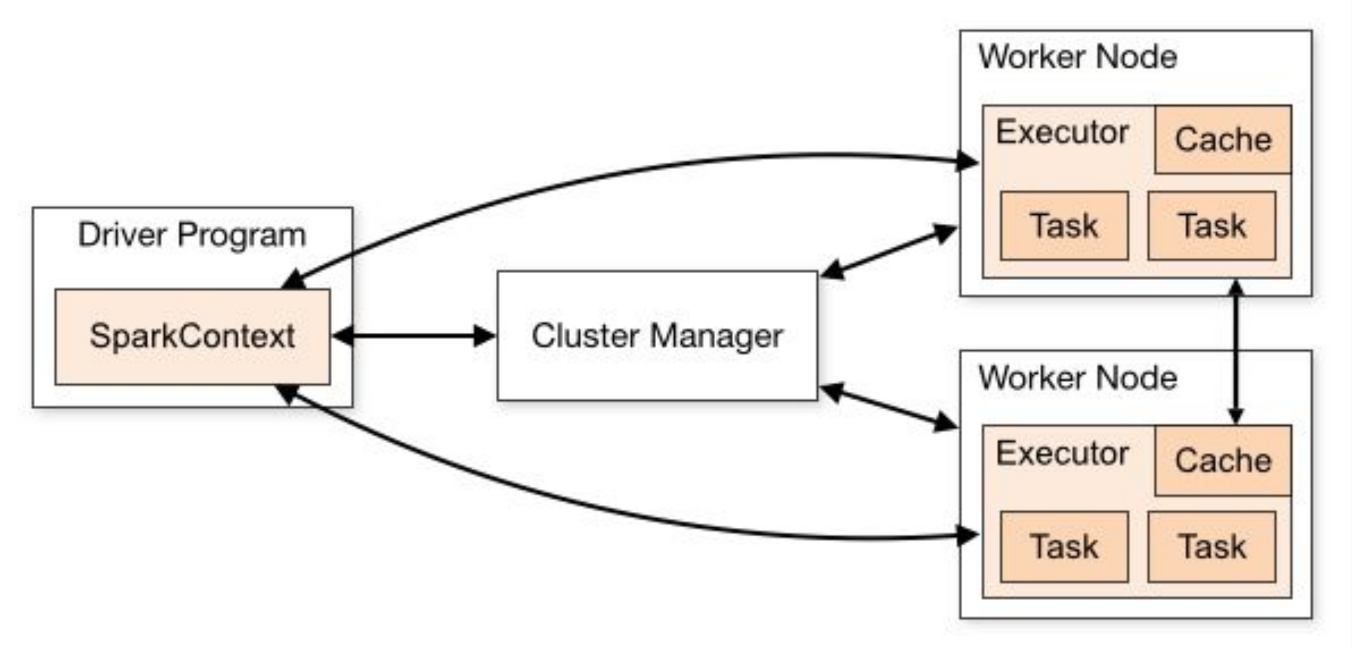
Standalone Scheduler

YARN

Mesos

+ kubernetes

Communication between components in a Spark application happens by all actors



Which edge in this diagram has not been discussed? Can you come up with a reason for its existence?

How does High Performance Computing differ from Spark/MapReduce?

Core concepts of the Spark API

- RDDs
- Datasets
- Row
- Column
- SparkSession

Demos with a pyspark-shell

Production-grade code comes with tests. They allow you to change code in the future, with a feeling of assuredness that stuff still works.

The rationale behind tests:

- Improves chance of code still being correct in the *future*
  - Code likely works now: people have the tendency to test their code (manually) on a small problem
  - Code will change, as requirements and environments change.
  - To prevent introducing breaking changes: write tests and ship these with the code.
- Raises confidence that code is correct now
  - assert that the results match expectations
  - trains you to think about edge cases, which aren't so uncommon as people may believe. Programming is an art about details. This is often times why non-techies do not understand that coding something up properly, can take a while.
- Most up-to-date form of documentation
  - word documents and wikis will grow out of sync with the code.
  - tests usually target a very specific piece of functionality and help you reason about those pieces in the bigger picture

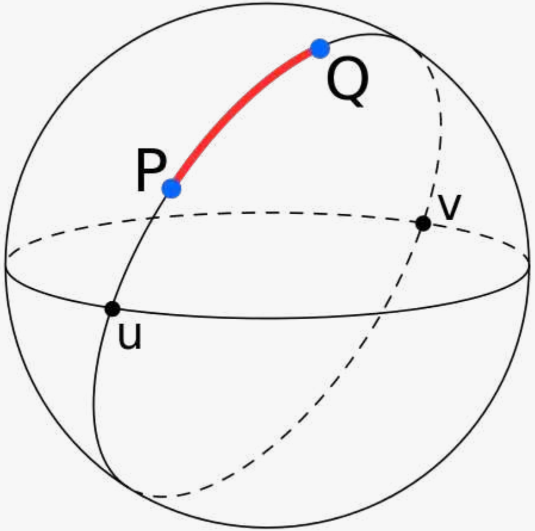
Pytest is one of the most well-known testing libraries in the Python ecosystem

Alternatives: unittest, doctest, nose

A basic test **asserts** something:

statement evaluating to bool	meaning
2 != 3	the numerical value of 2 is not that of 3
len(“hello”) == len(“world”)	the strings “hello” and “world” have the same number of characters
{1, 2, 3}.issubset(range(5))	the former set is a subset of the latter collection

A warm-up to testing PySpark code: let’s write a unit test for the great-circle-distance metric!



The great-circle-distance (gcd) or Haversine distance gives the shortest distance along the surface of a sphere between any two points.

It is a commonly encountered problem in anything related to locations.