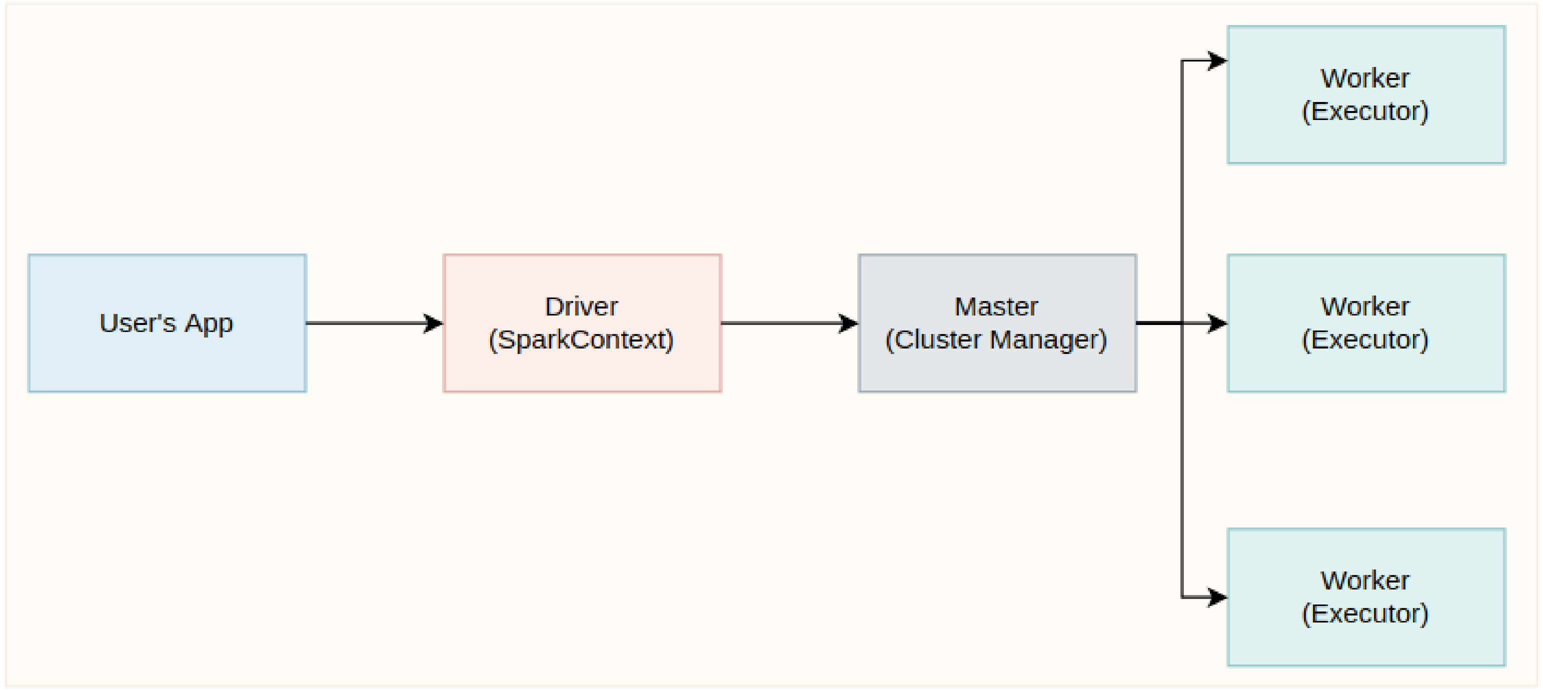
Optimization Spark Application Using Scala

Kodjo Klouvi | kodjo@osekoo.com

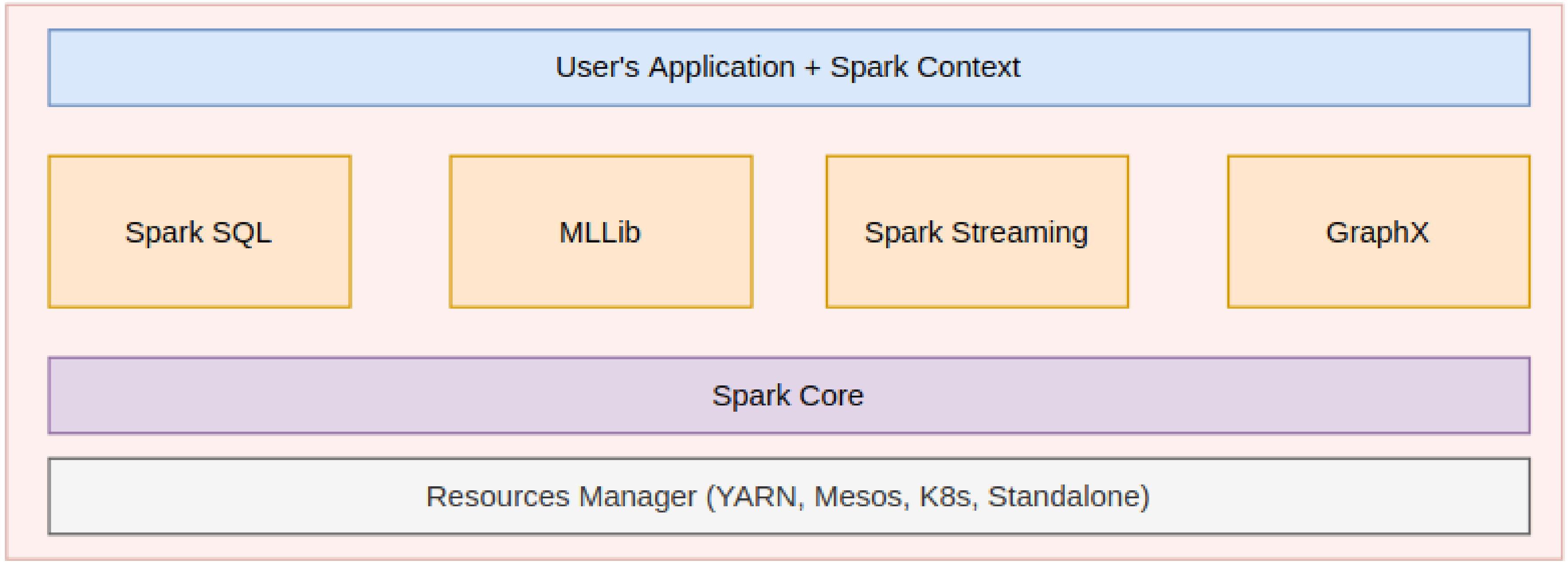
Apache Spark

- Big Data processing Engine (batch and streaming)
- In-memory data pipeline (ETL)
- Parallel execution
- Machine Learning

Spark Architecture



Spark Framework


User's Application + Spark Context
Spark SQL
MLLib
Spark Streaming
GraphX
Spark Core
Resources Manager (YARN, Mesos, K8s, Standalone)

Spark Optimization

- Make a difference between Transformations and Actions
- Use appropriate data format
- Clean data before Spark operations
- Filter data at the beginning of the ETL
- Avoid as much as possible Shuffle operations
- Apply appropriate partitioning

Spark Optimization

- Control the join tasks
- Use of broadcast join
- Cache data in memory or on disk
- Review logical and physical plans
- Be aware of DAGs
- Prefer Scala to Python
    - Python doesn't handle correctly UDFs
- Persist ML models' state

Transformations vs Actions

- Tranformations = Lazy Evaluation
    - GroupBy, select, orderBy, join, filter, read
    - Only the logical plan is computed
    - Few computing resources are used at this stage
- Actions = Execution
    - Collect, show, count, save, take
    - Optimized and physical plans are computed
    - Transformations are computed
    - Very high consumption of resources

Data format

- Store/persist data in parquet format (column-oriented structure)
    - Suitable for big data tasks
    - For future tasks
    - After shuffle tasks (to avoid re-shuffle)
    - For machine learning tasks

Pre-cleaning Data

- Remove unnecessary data before start spark operations
- Drop columns
- Review the dataset and keep only useful data

Filtering data at the beginning

- Apply filter, groupByKey, reduceByKey, join as soon as possible
- Cache or persist the data in memory or on the disk depending on the data size

Shuffle or not…

- The most expensive action in Spark!
- Moving data over the cluster
- Use the appropriate strategy when filtering or joining dataset
    - Parallelize tasks (partitioning)
    - Broadcast
    - hash

Partitioning

- Partition data to parallelize tasks
    - Repartition, partionBy, coalesce
- Partition data before saving it to disc

Review computation plans

- Spark computes 3 plans before execution
    - Logical plan
    - Optimized plan
    - Physical plans
- Explain(extend=true) displays all plans + DAGs
- Review the physical plan nad DAGs to detect any redundancy or unnecessary operations

Scala

- Object-oriented programming language
- Coupled with Java (Java bytecode/Gateway)
- Statically typed (String, Boolean, Int, Long, Float)
- SuperTypes **Any** and **AnyRef**

Scala

- Multithreading / Parallel run
- Supports optional parameters, named parameters, etc.
- Widely used by big companies (APple, Twitter, Google,…)
- Spark is built in Scala

Scala

- Case-sensitive: **DataFrame** is different from **Dataframe**
- Classes and interfaces name are camelcase with first capital letter (**HousePriceEstimator**)
- Methods and variables name are camelcase with first small letter (**getPrice()**)

Scala

- Use ***val*** keyword to create an **immutable** variable
    - val lastUpdate = "2020-11-23"
    - val level : Int = 17
- Use ***var*** keyword to create a **mutable** variable
    - var gender = "F"
    - var strike: Float= 85.9

Scala

- Example and comment at https://github.com/osekoo/hands-on-spark-scala/blob/develop/get-started/src/main/scala/WordCount.scala

Hands-On

- https://github.com/osekoo/hands-on-spark-scala
- Preparing dev environment
- Writing code
- Building/testing
- Packaging and submitting tasks

IntelliJ

- Integrated Development Environment
- Multiple modules available for Scala, SBT, Big Data, etc.
- Easy to debug
- Profiler available

Docker

- Platform as a service (PaaS)
- OS abstraction
- Self-contained applications running in containers
- Simplify applications delivery/distribution
- Enable (auto)scaling

Lab Session

- Sync Spark, Hadoop, Scala, Java and SBT versions
- We use here Spark 3.0.2, Hadoop 2.7, Scala 2.12.x, Java 8 and SBT 1.x
- Other details on the Github page

Troubleshooting

- Raise any question during the session or via email kodjo@osekoo.com