UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Apache Spark: Concept Review

Ana Rocha, Artur Rodrigues, Jean Morelli, Ricardo Correia, Rúben Cruz

1

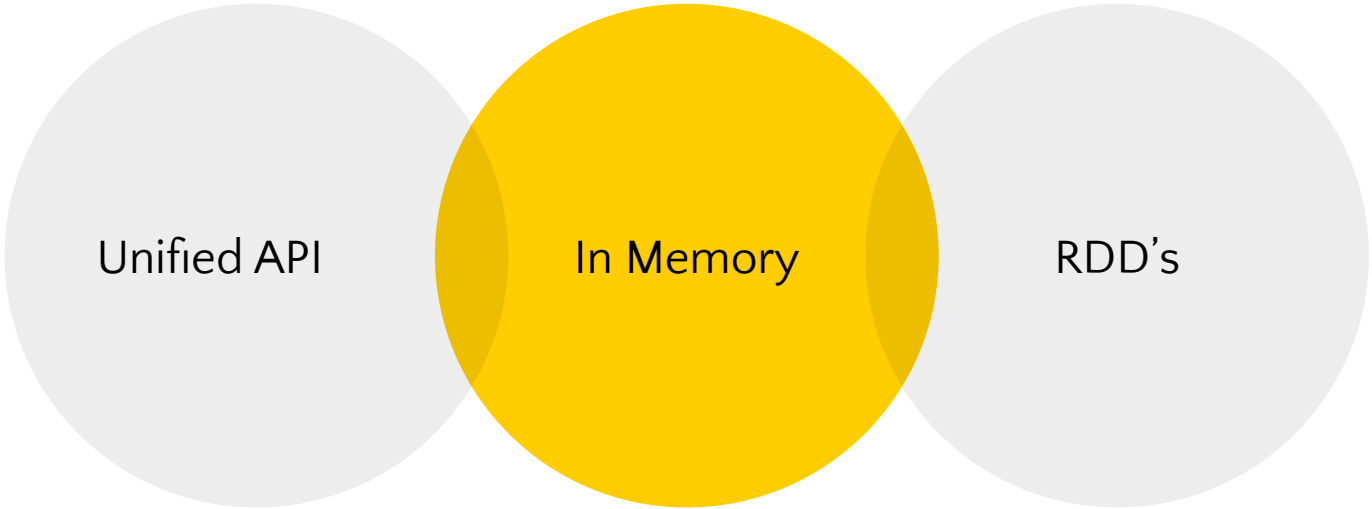
Spark: Overview

Story, characteristics and architecture

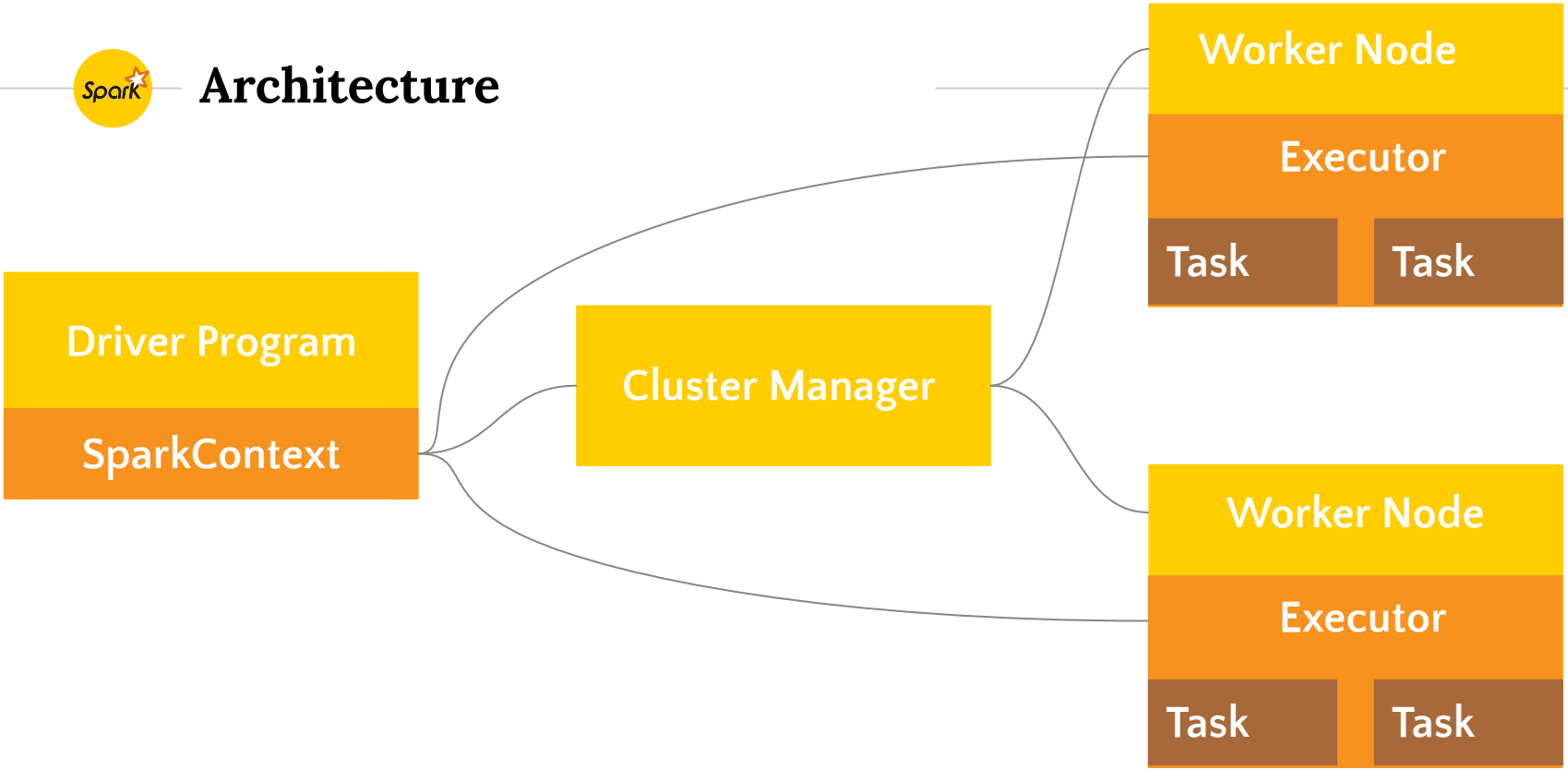
Unified analytics engine for large-scale data processing.

- Started in 2009 at UC Berkeley
- Open Source
- Data Processing

Characteristics



Architecture

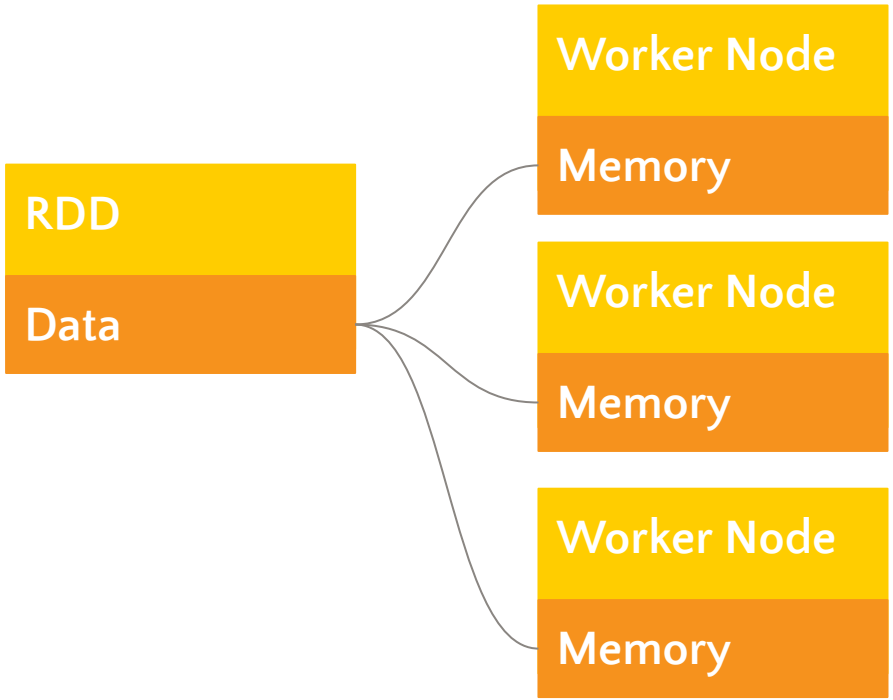


2

Spark: RDD

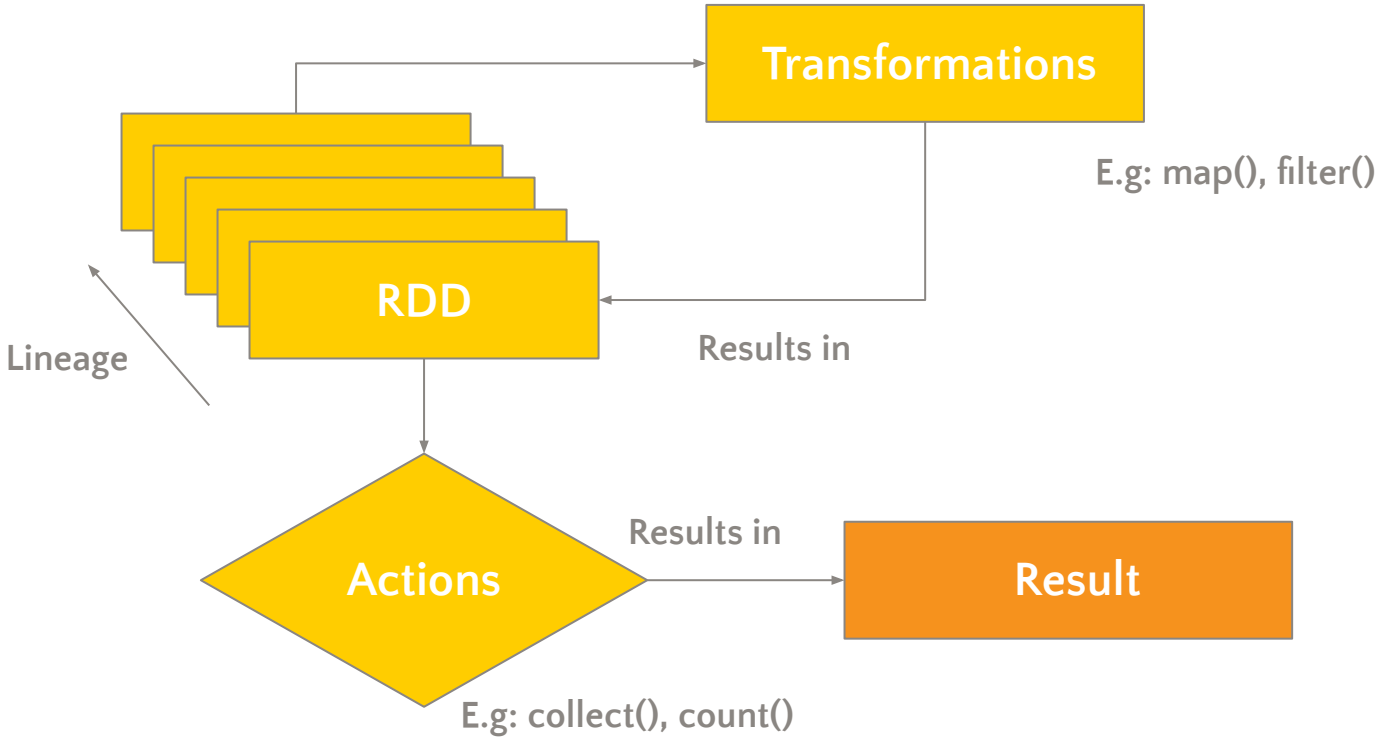
Spark's most important concept

Resilient Distributed Datasets (RDD)



- Spark's primary form of **abstraction**
- **Collection** of objects **partitioned in a cluster**
- Allows for great **performance** and **generality**
- Exposed in a **API** written in **several languages**
- **Fault tolerant**
- Allow for **two types of operations** by users

RDD operations

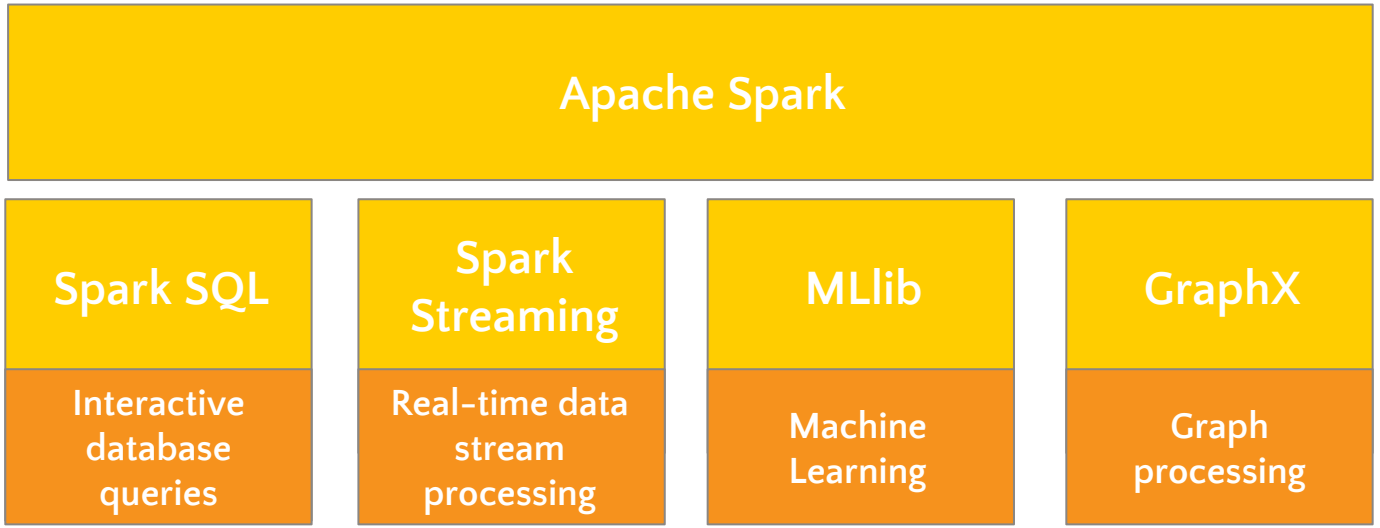


3

Spark: Libraries

Higher-level libraries, targeting the use cases of specialized computing engines

Higher-Level Libraries



4

Spark : Applications

Batch Processing | Interactive queries | Stream Processing | Scientific Applications

Applications: Batch processing

- Processing of **large datasets**
- **Structuring** of raw data
- Training of **machine learning** models
- **Real world examples:**

Applications: Interactive queries

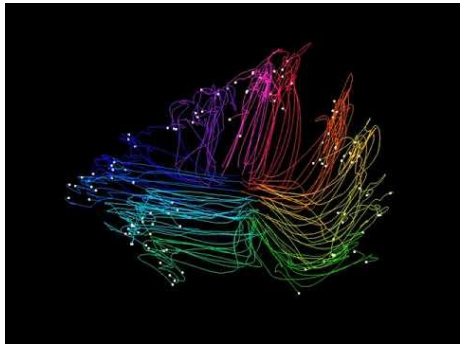
- **Relational queries** (Spark SQL)
- Use of **shells** (Spark API)
- **Domain specific interactive** applications
- **Real world examples:**

Applications: Stream Processing

- **Real time data stream processing**
- Often combined with **batch processing** and **interactive queries**
- **Real world examples:**

Applications: Scientific Applications

- Large Scale **Spam detection**
- **Image** processing
- **Genomic** data processing
- **Real world examples:**



Thunder. Credit: Jeremy Freeman, Nikita Vladimirov, Takashi Kawashima, Yu Mu, Nicholas Sofroniew, Davis Bennett, Joshua Rosen, Chao-Tsung Yang, Loren Looger, Philipp Keller, Misha Ahrens

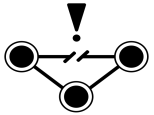

APACHE
Spark
VS
hadoop
MapReduce

Apache Spark VS Hadoop MapReduce

Cost

Performance

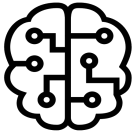
Fault Tolerance



Ease of use

Security

Machine Learning

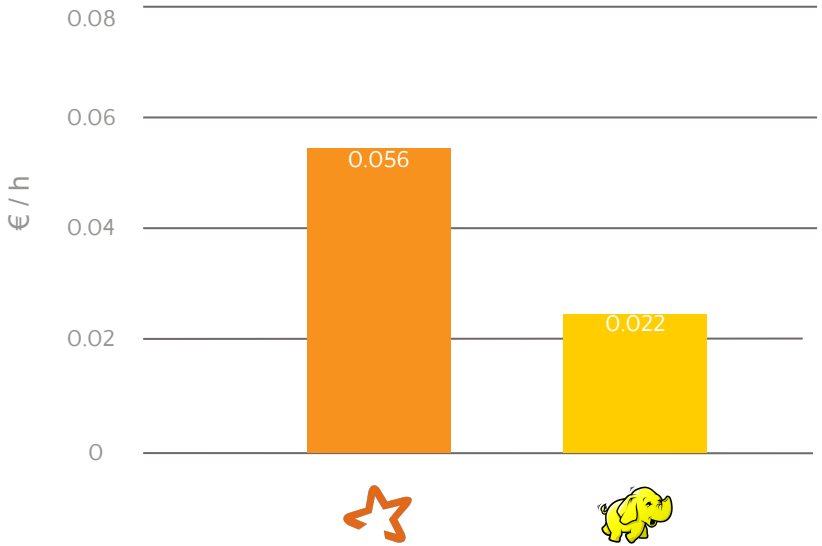


Scheduling and Resource Management

Data Processing

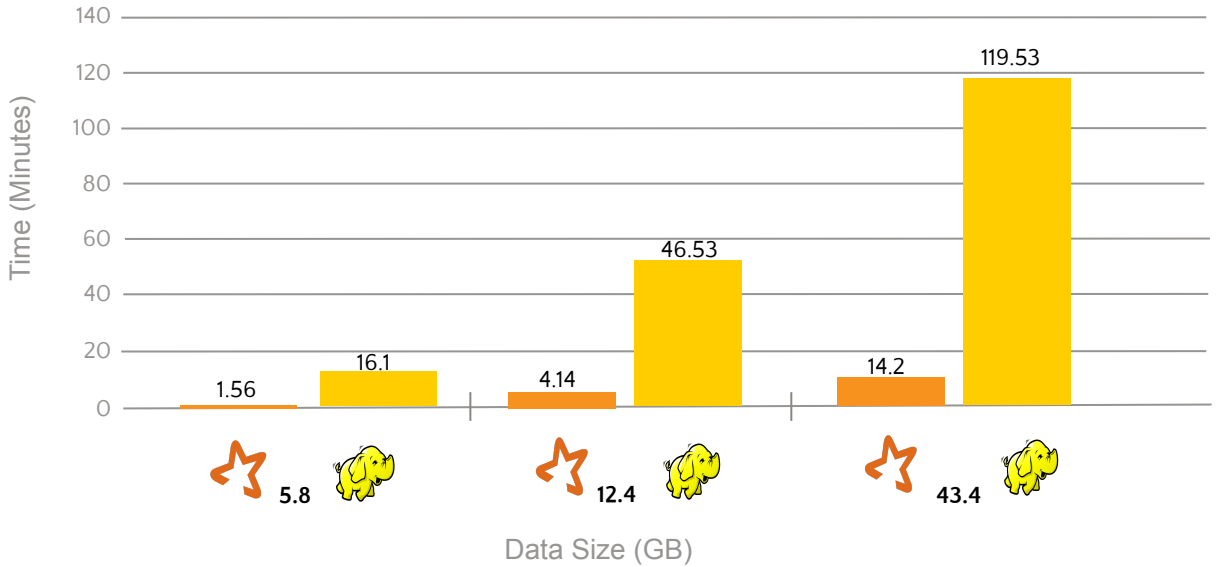
Scalability

Cost

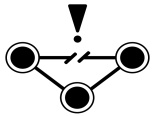


Spark VS Hadoop MapReduce

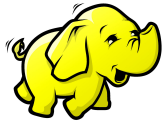
Performance



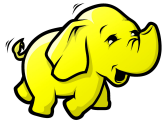
Fault Tolerance



- Lineage mechanism
- Directed acyclic graph



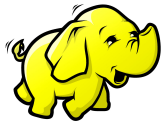
- Data replication



- Master node and Slave node

Ease of use

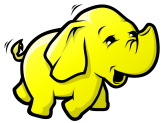
- Java,Python, Scala, R, Spark SQL
- Interactive mode
- Code reusability



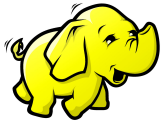
Java,Python

Security

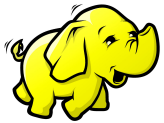
- Authentication via secret
- Authentication via event logging
- HDFS or YARN integration



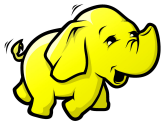
- Kerberus



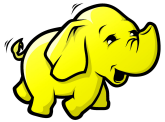
- Apache Ranger



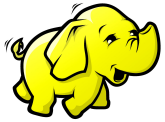
- ACL's



- Inter-node encryption

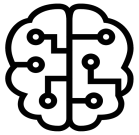


- HDFS file permissions

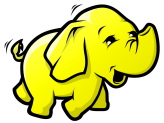


- Service level authorization

Machine Learning

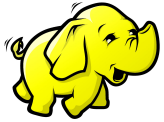


MLLib



Mahout

More than 50 common algorithms for distributed model training



Susceptible to serious I/O performance issues

Scheduling and Resource Management

- Functions are built-in
- DAG Scheduler
- Spark Scheduler and Block Manager

- Relies on external solutions to deal with resource management and scheduling issues
- YARN or third-party plugins like capacityScheduler and FairScheduler.

Apache Spark VS Hadoop MapReduce

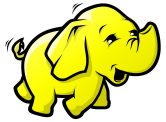
Data Processing

- In-memory data processing using RDD's
- RDD's too big are split into partitions and moved to the closest nodes
- Perfect for real-time processing

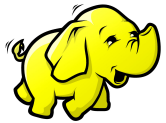
- Stores data in disk memory
- Splits data into batches and processes parallelly with Map-Reduce
- Perfect for batch processing

Scalability

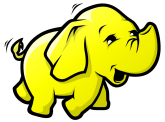
- No native filesystem, so when data grows too large it relies on a external file system implementation.
- Incredibly reliant on RAM memory, which is expensive
- Able to deal with data in the order of the petabytes



- As the data grows, it inherently scales to accommodate the demands



- Relies on disk-memory, which is somewhat cheap nowadays



- Able to deal with data in the order of the exabytes

Final Thoughts

- Leading framework for big data.
- In memory and RDD's
- Ideal scenario – Size of datasets smaller than the amount of available RAM

Thanks!

Any questions ?

- a63971@ualg.pt | a64591@ualg.pt
- a64592@ualg.pt | a64007@ualg.pt
- a64014@ualg.pt