THE FUNCTIONAL ARCHITECTURE STUDIO

APACHE Spark™

Spark y sombras del big data

Habla Computing
info@hablapps.com
@hablapps

About me

- Senior data engineer at Habla Computing
- 13+ years experience, 4+ using Apache Spark
- Experience with other Scala technologies
  - Akka
  - Cats
- Domain Specific Language development

What will be discussed at this talks

- The tale of big data
- What are Apache Spark strengths?
- Notebook session
- Conclusions

Once upon a time in big data engineering...


hadoop
Map Reduce
sqoop
mahout
HIVE
FLUME
hadoop
HDFS
OOZIE
MOVIECLIPS.COM

Apache Hadoop Ecosystem

Ambari
Provisioning, Managing and Monitoring Hadoop Clusters

Sqoop Data Exchange

Oozie Workflow

Pig Scripting

Mahout Machine Learning

R Connectors Statistics

Hive SQL Query

Flume Log Collector

YARN Map Reduce v2
Distributed Processing Framework

HDFS
Hadoop Distributed File System

and then a hero came to save us...

he built a tool to rule the big data kingdoms...

APACHE Spark™

and the big data community lived happily ever after...

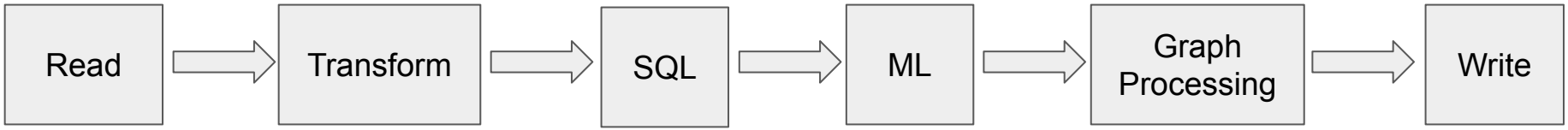

APACHE
Spark™
IWDRM

...or at least things became simpler.

What are Spark strengths?

- Unified programming model
- Better abstractions (RDD, DataFrame, Dataset)
- Declarative
  - Optimizations
  - Push down filters
  - Read Schema

Unified programming model



Better abstractions

```
function map(String name, String document):
  for each word w in document:
    emit (w, 1)

function reduce(String word, Iterator partialCounts):
  sum = 0
  for each pc in partialCounts:
    sum += pc
  emit (word, sum)
```

Better abstractions

```
textFile.flatMap(line => line.split(" "))
    .map(word => (word, 1))
    .reduceByKey(_ + _)
```

Declarative

Declare what to do

- The driver declares what needs to be done.
- Declarativity
  - Transformations
  - Actions
- Output: Logical plan

Declarative

Declare what to do

- The driver declares what needs to be done.
- Declarativity
  - Transformations
  - Actions
- Output: Logical plan

Optimization phase

- Catalyst optimizer
  - ReadSchema
  - Pushdown filters
- Input: Logical plan
- Output: Physical plan

Declarative

Declare what to do	Optimization phase	Distributed execution

● The driver declares what needs to be done.
● Declarativity
  ○ Transformations
  ○ Actions
● Output: Logical plan
	● Catalyst optimizer
	  ○ ReadSchema
	  ○ Pushdown filters
	● Input: Logical plan
	● Output: Physical plan
		● Tasks assigned to executors
		● Optimized serialization
		● Input: Physical plan
		● Output: Side effects

Dude, are you trying to sell me a motorbike?

Recap

Thanks

See you at the course