Big Data - Spark, Hadoop e afins.

Leonardo Gloria

Aula 03

Instituto Infnet

Google é sinônimo de escalabilidade!

Google InHouse Solution

* Google File System(GFS)
* BigTable
* Map Reduce(MR)

HDFS

Apache Hadoop

* Hadoop Common
* Map Reduce
* HDFS
* Apache Hadoop YARN

Hadoop Ecosystem

Surge o Apache Spark

What is Apache Spark

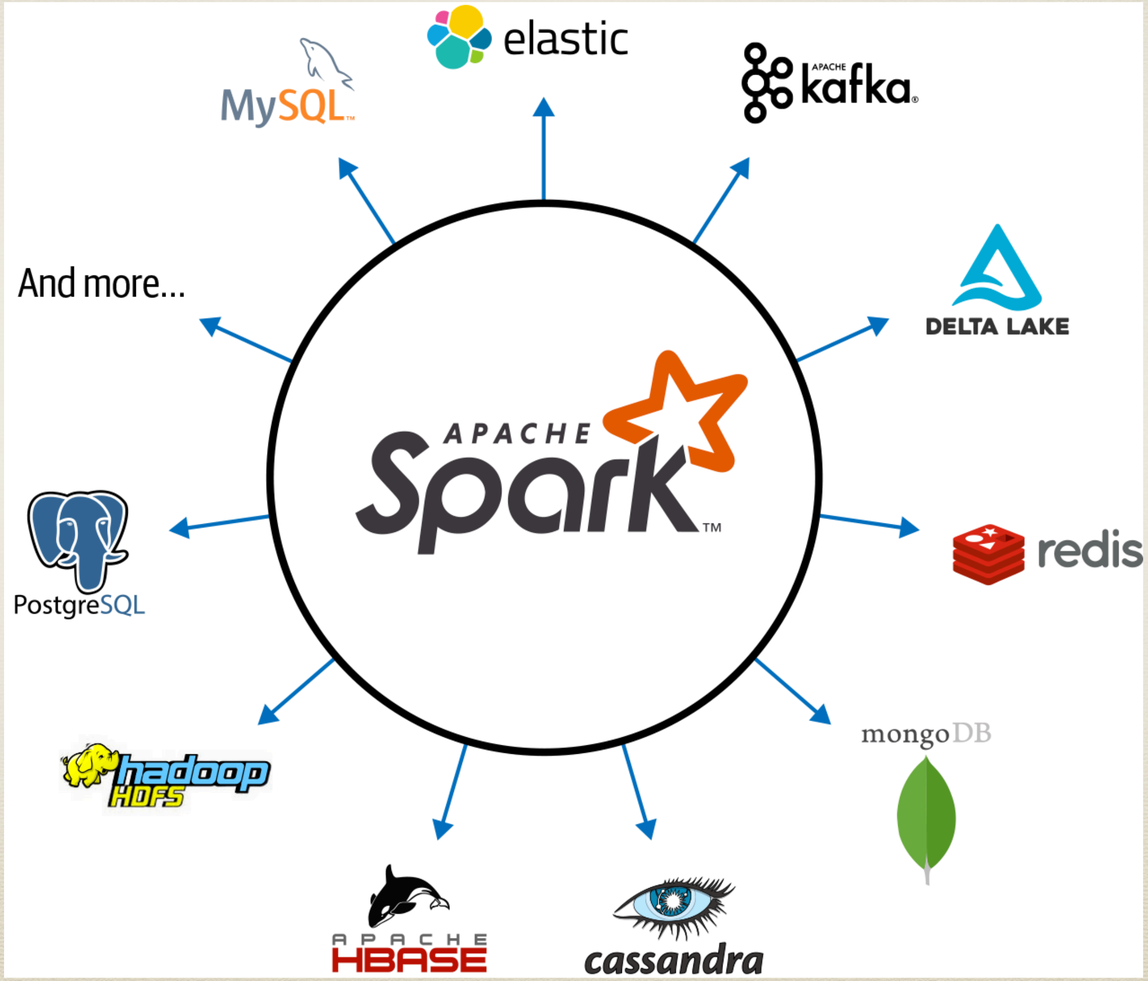
What is Apache Spark

* Engine unificada pensada para processamento de grande quantidades de dados em ambientes on-premises ou em Cloud.
* In Memory Storage
* Apis para Machine Learning - MLLib
* Sql para queries (Spark SQL)
* Processamento de Stream (Structured Streaming)
* Processamento de Grafos (GraphX)

Características chave

* Speed
* Easy of use
* Modularity
* Extensibility

Características chave



Spark Sql

```
spark.read.json("s3://meubucket/commiters.json")
    .createOrReplaceTempView("commiters")

val results = spark.sql("""

select name from commiters where num_commits > 10

  """)
```

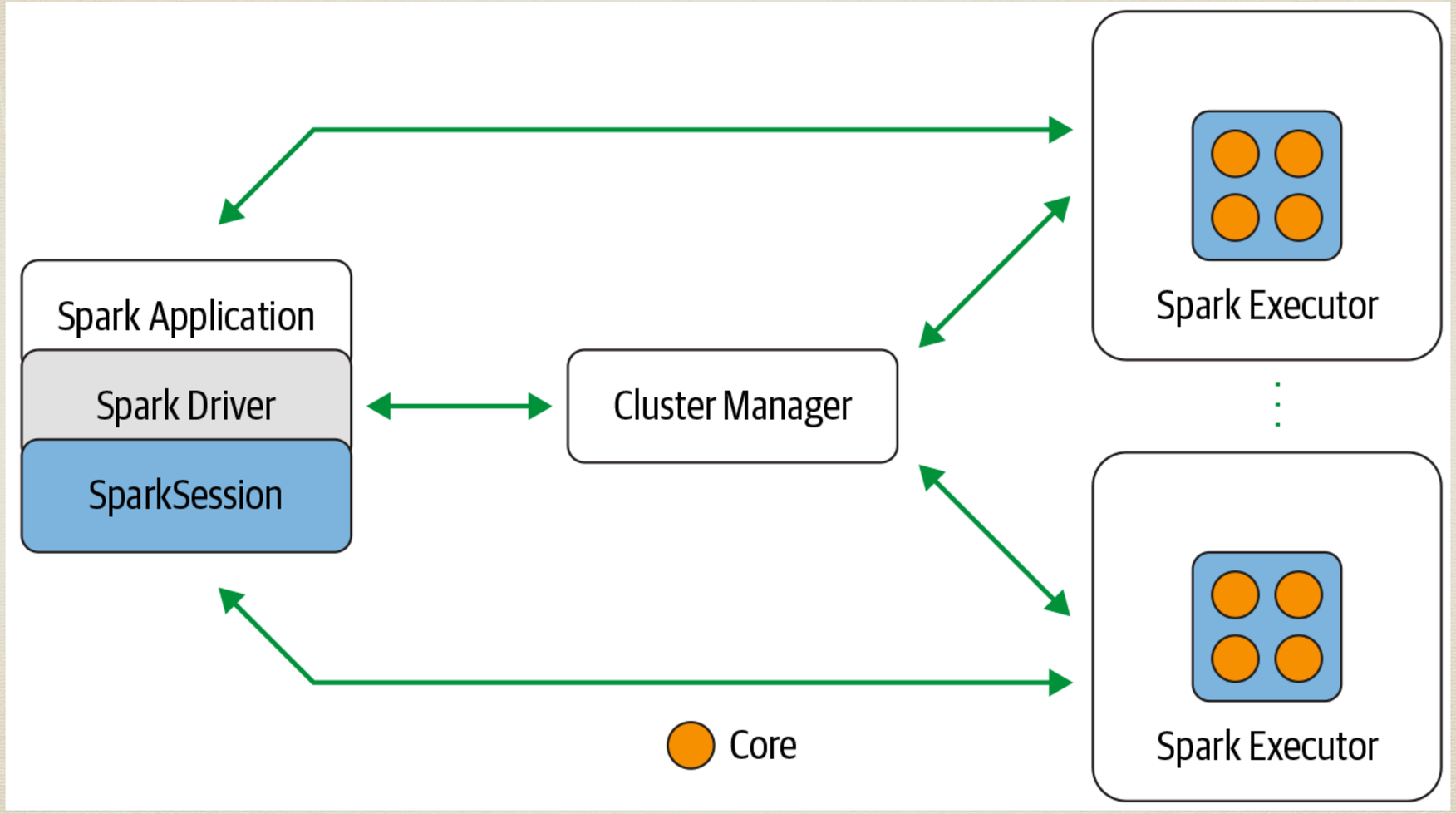
Spark MLLib

```
1 from pyspark.ml.classification import LogisticRegression
2
3 training = spark.read.csv("s3://treino.csv")
4 test = spark.read.csv("s3://teste.csv")
5
6 lr = LogisticRegression(maxIter=10, regParam=0.3,
  elasticNetParam=0.8)
7
8 lr_model = lr.fit(treino)
9 lr_model.transform(test)
```

Graph X

```
1 val graph = Graph(vertex,edges)
2 messages = spark.textFile(hdfs://messages.txt)
3 val graph2 = graph.joinVertices(messages){
4     (id,vertex,message) => ....
5 }
6
```

Mecanismo de Execução distribuída.


Spark Application
Spark Driver
SparkSession
Cluster Manager
Spark Executor
Spark Executor
Core

```
1  import org.apache.spark.sql.SparkSession
2  val spark = SparkSession.builder
3              .appName("Hello World")
4              .config("spark.sql.shuffle.partitions" 6)
5              .getOrCreate()
6
7  val cars = spark.read.json( "... ")
8  val resultDF = spark.sql("select * from cars")
9
10
```

Cluster Manager

* Responsável por gerenciar e alocar recursos dos nós do cluster onde sua Spark Apllication roda.
* Spark atualmente Suporta 4 tipos de Cluster Manager
  * Built-in standalone cluster manager
  * Apache Hadoop Yarn
  * Apache MEsos
  * Kubernetes

Spark Executor

* Roda em cada nó do cluster.
* Se comunica com o Driver e é responsável por executar as tarefas.
* Maioria dos Deployments modes só um executor rodando por nó.

Deployments Modes

Mode	Spark Driver	Spark Executor	Cluster Manager
Local	Roda em uma JVM (ex notebook, ou Single onde)	Roda na mesma JVM do Driver	Roda no mesmo host
Standalone	Pode rodar em qualquer nó no cluster	Cada nó no cluster vai rodar seu próprio executor	Pode ser alocado arbitrariamente para em qualquer nó no cluster
YARN Client - Cluster	Roda no client / Master	Yarns Node Manager Container	Yarn Resource Manager aloca os containers nos nós para a execução.
Kubernetes	Roda em um POD Kubernetes	Roda Dentro do seu proprio POD	Kubernetes Master

Who Uses Spark , and for What?

Dúvidas?

“Stay hungry, stay foolish.”

–Steve Jobs