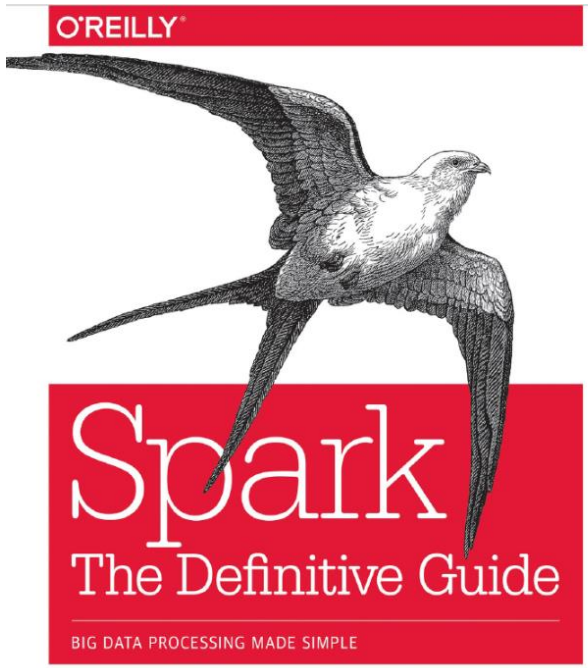
Introdução ao Apache Spark

Arquitetura do Spark - Parte 1



Capítulos Abordados

2. A Gentle Introduction do Spark

- **Master/Worker Architeture**
- **The SparkSession**
- **Spark´s Languages APIs**
- **Sparks API´s**
- **Starting Spark**

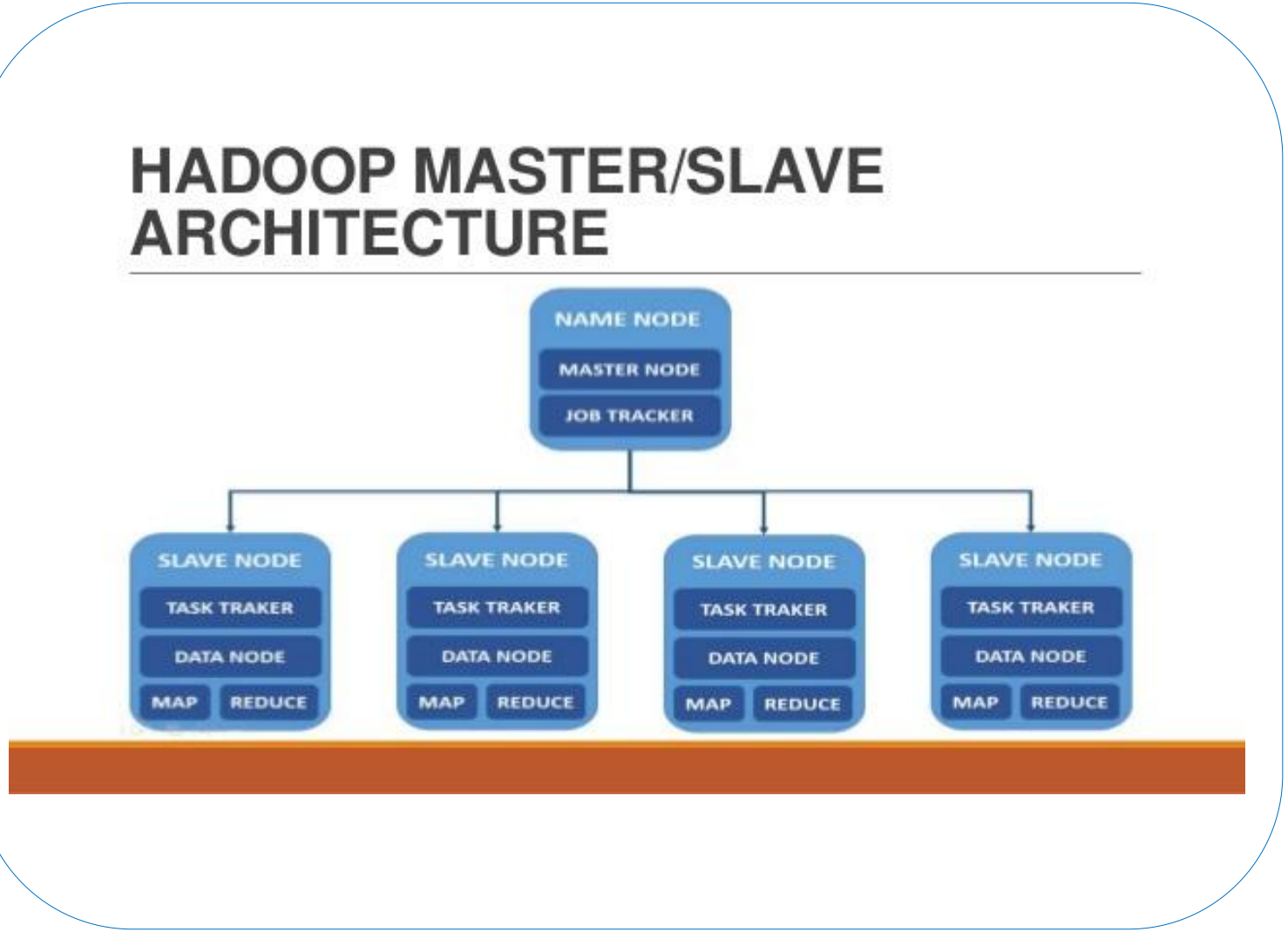
<Master/Worker Architecture/>

Master:

- Coração do Cluster
- Aloca recursos, agenda e monitora tarefas executadas pelos Workers
- Mantém a integridade do Sistema

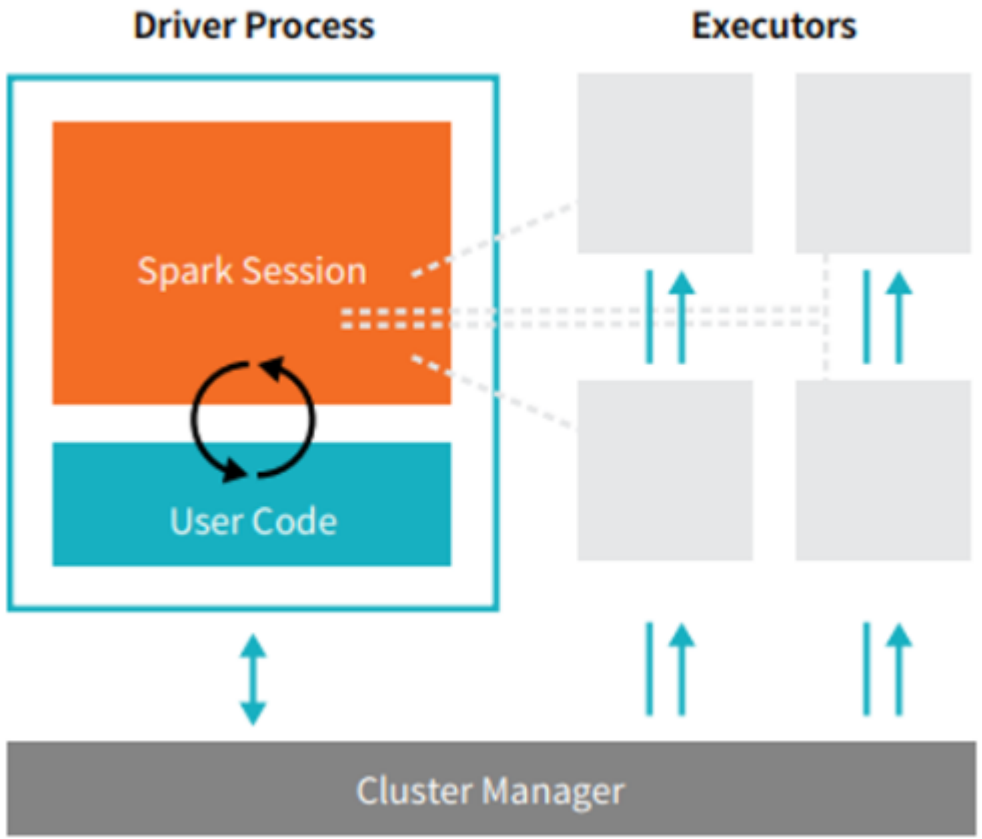
Worker/Slave:

- Executa tarefas
- Persiste dados



<Master/Worker Architecture/>

Aplicação Spark



Driver (1) –> Coração da Aplicação!

Mantém as informações da aplicação;
Responde ao código desenvolvido pelo usuário;
Analisa, distribui e agenda tarefas para os executores.

Executores (n)

Processa dados (executa os códigos designado a ele);
Reporta o resultado da computação ao driver.

Cluster Manager

Controla as máquinas físicas e aloca recursos à Aplicação Spark;
Pode ser: Standalone, YARN, Mesos e Local Mode.

Pode ter mais de 1 Aplicação Spark rodando no mesmo Cluster!

<Languages APIs/>

Scala

Spark foi primeiramente escrito em Scala;

Linguagem Padrão do Framework.

```scala
val spark = new SparkContext()

val lines    = spark.textFile("hdfs://docs/")    // RDD[String]
val nonEmpty = lines.filter(l => l.nonEmpty())   // RDD[String]

val count = nonEmpty.count
```

Python

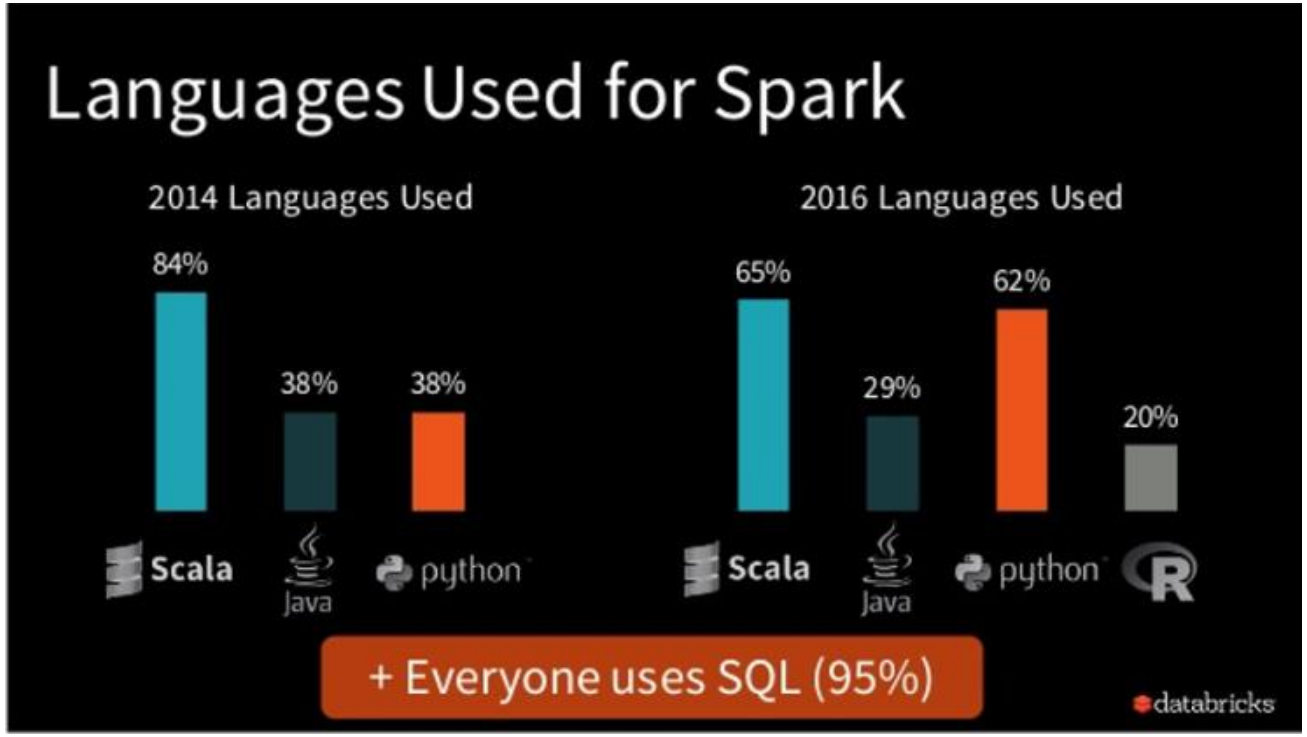
Possui quase todas as bibliotecas presentes em Scala.

Amplamente utilizada por ser fácil de aprender!

```python
spark = SparkContext()

lines = spark.textFile("hdfs://docs/")
nonEmpty = lines.filter(lambda line: len(line) > 0)

count = nonEmpty.count()
```



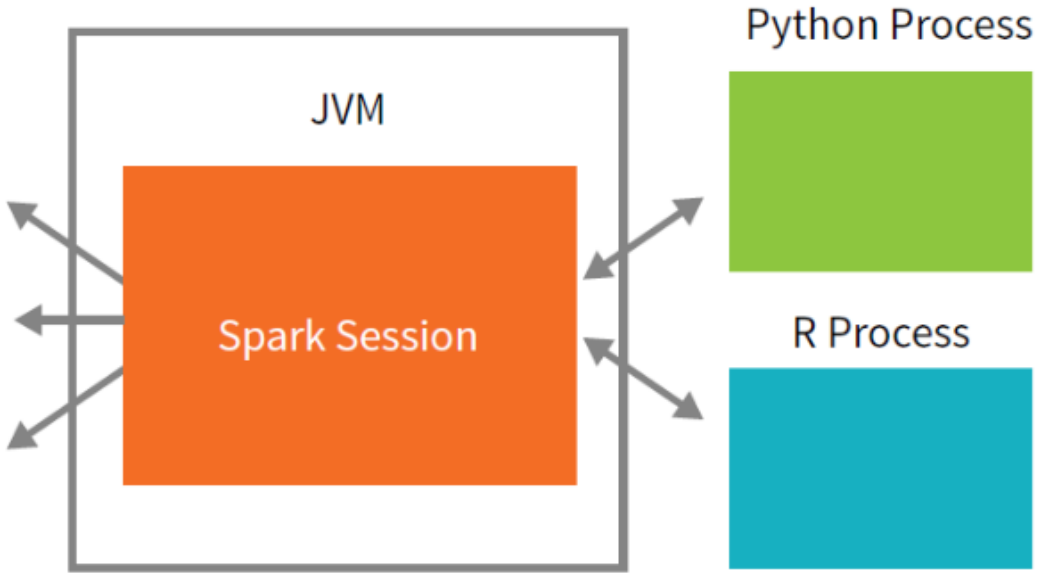
SQL, Java, Scala, R and Python (coming soon: .NET)

<Languages APIs/>

SparkSession: ponte entre a linguagem e a aplicação Spark

Todas as linguagens mantem o mesmo conceito: o código escrito na linguagem escolhida será traduzido para códigos que irão rodar nas JVMs (Java Virtual Machines).

Executors
(Também possuem JVM)



PySpark: Construído em cima da Spark´s Java API.

O driver utiliza o Py4J para iniciar uma JVM e criar o JavaSparkContext.

Próximos Vídeos: UDF in Python