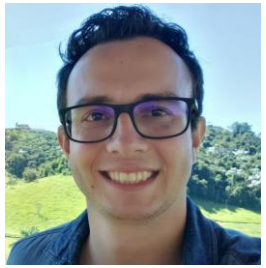
Spark – Big Data Processing

Aula 7

Semantix®

All about data

Quem sou eu?



Rodrigo Augusto Rebouças

Engenheiro de dados da Semantix
Instrutor do Semantix Academy

Contatos
rodrigo.augusto@semantix.com.br
linkedin.com/in/rodrigo-reboucas

RODRIGO.A.REBOUCAS

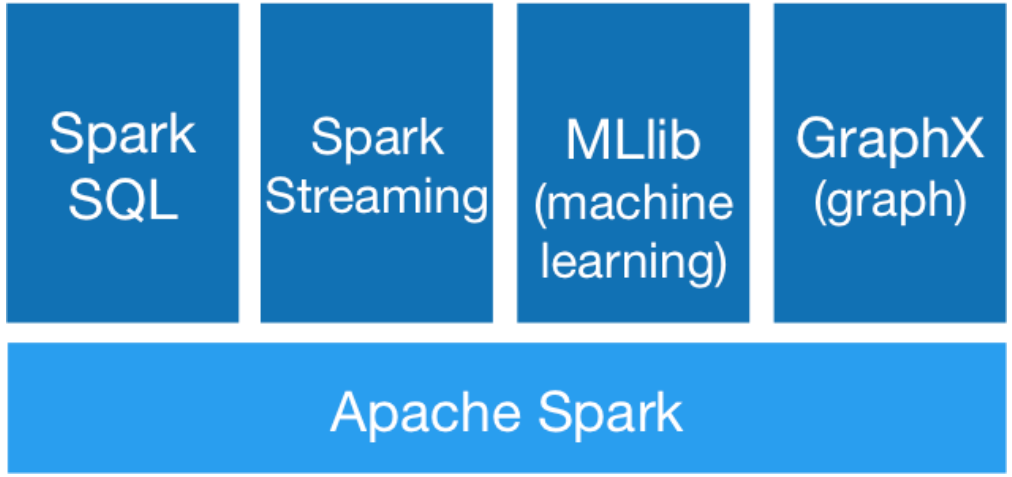
Semantix®
All about data

Spark Streaming

Semantix®
All about data

Ferramentas

- **Spark**
  - ETL e processamento em batch
- **Spark SQL**
  - Consultas em dados estruturados
- **Spark Streaming**
  - Processamento de stream
- Spark MLib
  - Machine Learning
- Spark GraphX
  - Processamento de grafos

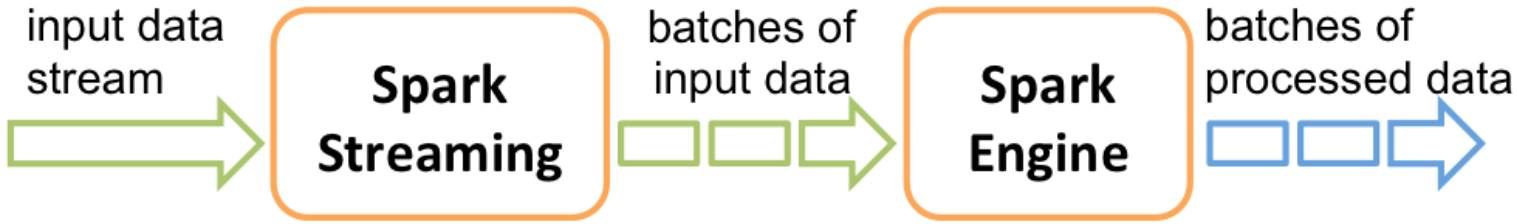


Spark Streaming

- Abstração de alto nível
  - Dstreams (Discretized Streaming)
  - Representa um Stream contínuo de dados
- Extensão da API core do Spark
  - Processamento escalonável
  - Alta taxa de transferência
  - Tolerante a falhas de stream de dados

Spark Streaming

- Recebe fluxos de dados de entrada e divide os dados em lotes
  - Processados pela engine do Spark para gerar o stream final de resultados em bath
- DStream é representado como uma sequência de RDDs



Spark Streaming – Leitura de Dados

Semantix®
All about data

Dstream - Leitura Básica

Scala

Python

- Criar um Contexto com Intervalo de 2 segundos

```
import org.apache.spark._
import org.apache.spark.streaming._
val conf = new SparkConf().setMaster("local")
val sc = new SparkContext(conf)
val ssc = new StreamingContext(sc, Seconds(2))
```

```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
conf = SparkConf().setMaster("local")
sc = SparkContext.getOrCreate(conf)
ssc = StreamingContext(sc,2)
```

- Criar um Dstream para captura dos dados relativos a sessão da porta 9999

```
val dstr = ssc.socketTextStream("localhost", 9999)
```

```
dstr = ssc.socketTextStream("localhost", 9999)
```

Dstream - Ex. Leitura e Exibição de uma Porta

- Exemplo de leitura na porta 9999 no localhost

```
from pyspark.streaming import StreamingContext
ssc = StreamingContext(sc,2)
readStr = ssc.socketTextStream("localhost",9999)
readStr.pprint()
ssc.start()
```

- Usar Netcat para enviar dados na porta 9999

```
$ nc -lp 9999
```

Exercícios – Dstream Leitura

1. Instalar o NetCat no container do spark
   - apt update
   - apt install netcat
2. Criar uma aplicação para ler os dados da porta 9999 e exibir no console

Spark Streaming – Operações

Semantix®
All about data

Spark Streaming – Operações

- Ação: Retorna um valor
  - Count
  - CountByValue
  - Reduce
  - Print
  - ForeachRDD
- Transformação: Retorna um DStream
  - Map
  - Filter
  - FlatMap
  - ReduceByKey

Spark Streaming

- Flatmap

```
from pyspark.streaming import StreamingContext
ssc = StreamingContext(sc,2)
readStr = ssc.socketTextStream("localhost",9999)
palavras = readStr.flatMap(lambda linha: linha.split(" "))
palavras.saveAsTextFiles(“hdfs://localhost/linha”)
ssc.start()
```

Spark Streaming

- Transformações de Map

```
pMinuscula = palavras.map(lambda palavra: palavra.lower())

pMaiuscula = palavras.map(lambda palavra: palavra.upper())
```

Spark Streaming

- Filtrar dados

```
filtro_a = palavras.filter(lambda palavra: palavra.startswith("a"))

filtro_tamanho = palavras.filter(lambda palavra: len(palavra)>5)

num_par = numeros.filter(lambda numero: numero % 2 == 0)
```

Spark Streaming – Contar Palavras

- Exemplo

Semantix®
All about data

Spark Streaming – Contar Palavras

- Exemplo de contar palavras dos dados na porta 9999 no localhost

```
import org.apache.spark.streaming._
ssc = StreamingContext(sc, 1)
readStr = ssc.socketTextStream("localhost“,9999)
palavras = readStr.flatMap(lambda linha: linha.split(" "))
pMinuscula = palavras.map(lambda palavra: palavra.lower())
pChaveValor = pMinuscula.map(lambda palavra: (palavra,1)
pReduce = pChaveValor.reduceByKey(lambda key1,key2: key1+key2)
pReduce.pprint()
ssc.start()
```

Exercícios – Dstream Word Count

1. Criar o diretório no hdfs “/user/rodrigo/stream”
2. Criar uma aplicação para contar palavras a cada 10 segundos da porta 9998 e exibir no console durante 50 segundos
3. Criar uma aplicação para contar palavras a cada 10 segundos da porta 9998 e salvar os dados no namenode no diretório “hdfs://namenode/user/rodrigo/stream/word_count” durante 50 segundos

Semantix®

All about data

contato@semantix.com.br

www.semantix.com.br