

Data Science Bootcamp



Host Oficial
sãojudas
universidade



MÓDULO #7.0

Data Engineering - Spark

Cinthia M. Tanaka



Cinthia

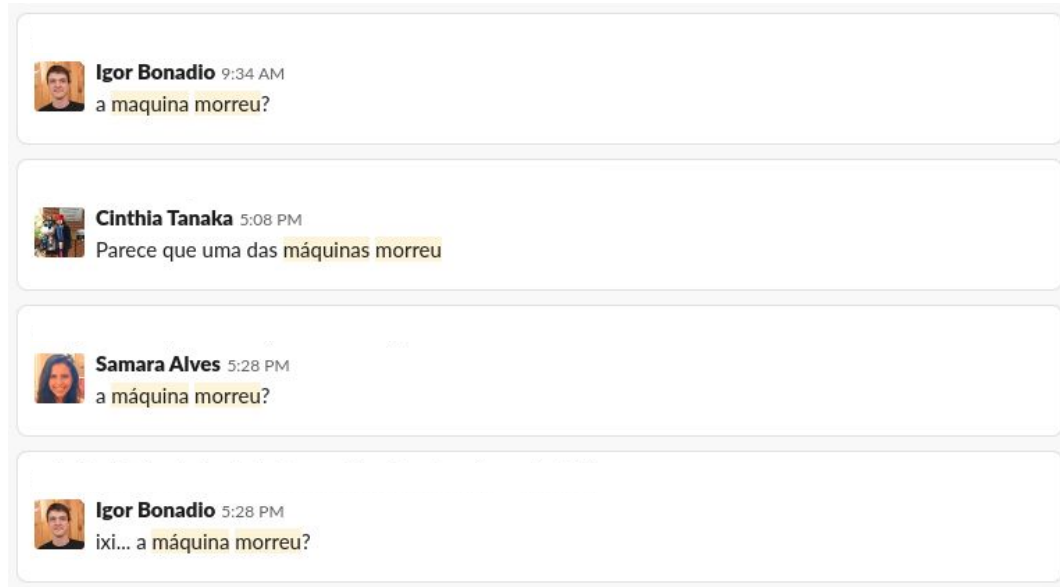
Data scientist no **Elo7**

 @cimarieta

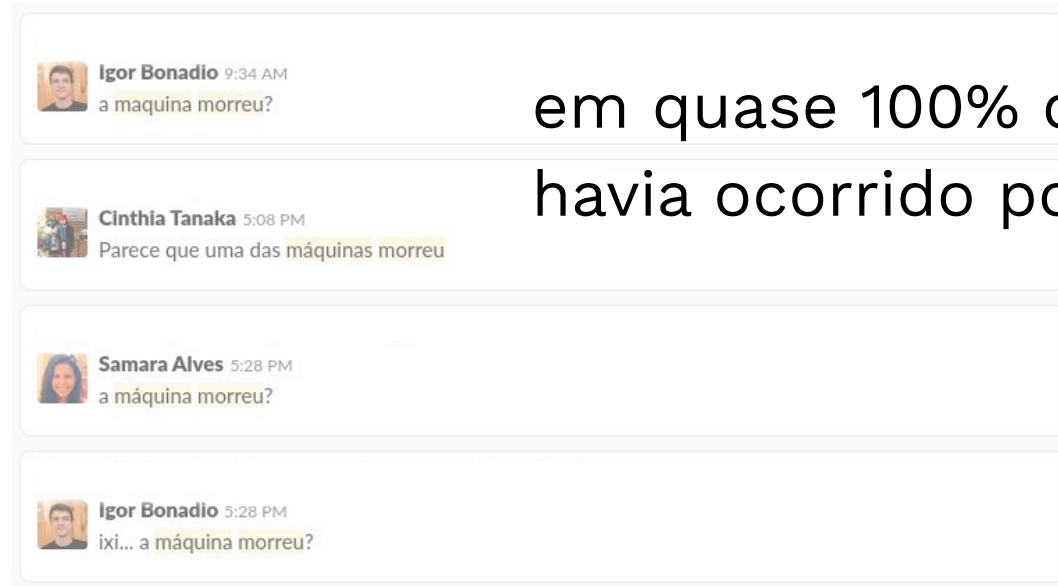
 cinthia-tanaka



Conversas cotidianas...

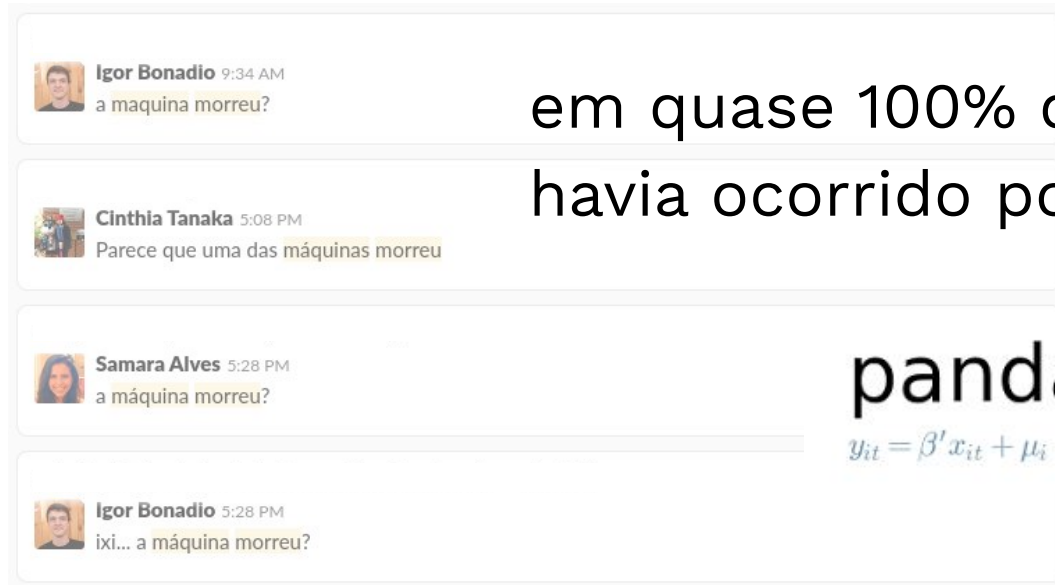


Conversas cotidianas...



em quase 100% das vezes, isso
havia ocorrido por causa do...

Conversas cotidianas...



em quase 100% das vezes, isso
havia ocorrido por causa do...

pandas
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



Mas... o ruim não era o Pandas

E sim, nossa escolha de usá-lo
com um dataset **gigante**



O que estava faltando era o...

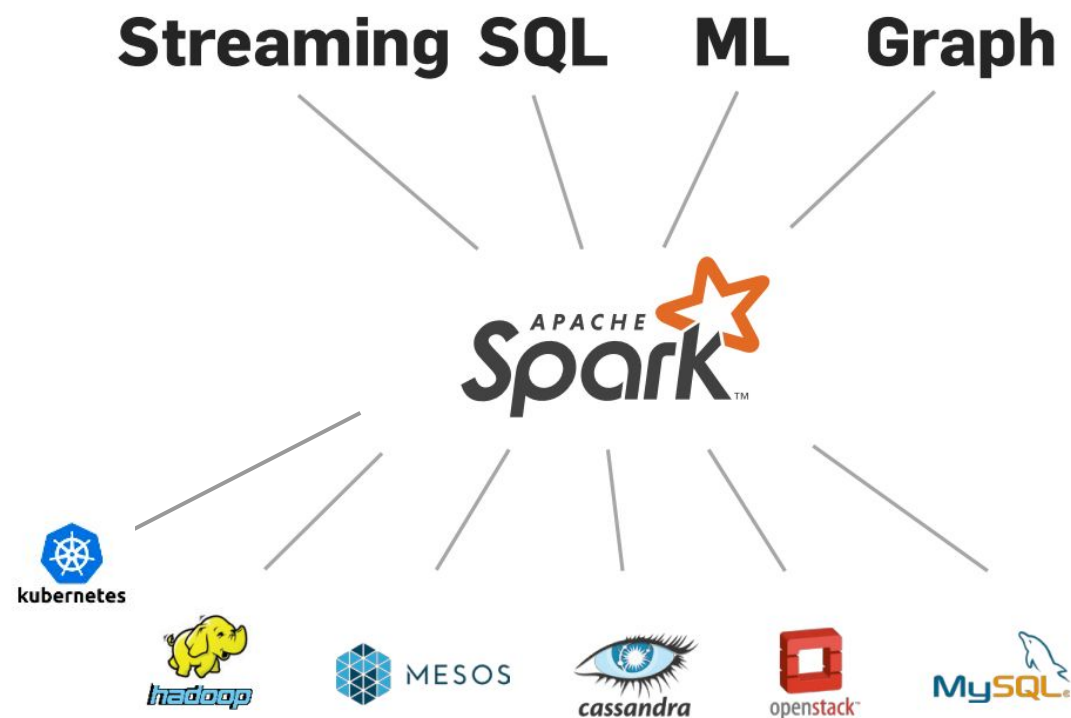


O que é Spark?

Apache Spark™ is a **unified** analytics engine
for **large-scale data** processing.



O que é Spark?



Saiba mais: Zaharia, Matei, et al. "Apache spark: a unified engine for big data processing." Communications of the ACM 59.11 (2016): 56-65. [\[link\]](#)






















Vantagens do Spark - analogia



Vantagens do Spark

- ▷ API unificada

0	1	2	3	4	5	6	7	8	9
〇	一	二	三	四	五	六	七	八	九
-	I	II	III	IV	V	VI	VII	VIII	IX
o	১	২	৩	৪	৫	৬	৭	৮	৯
									
-									



0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---



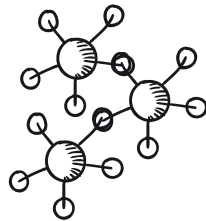
Vantagens do Spark

- ▷ API unificada
- ▷ facilidade de trabalhar com os dados (em memória)



Vantagens do Spark

- ▷ API unificada
- ▷ facilidade de trabalhar com os dados (em memória)
- ▷ mais possibilidades
 - queries interativas em grafos
 - modelos de machine learning com streaming [\[veja mais\]](#)



**Qual a mágica
do Spark?**

Qual a mágica do Spark?

MapReduce + **Resilient Distributed Datasets** + **Lazy Evaluation**



Entenda mais: Why Spark DataFrame, lazy evaluation models outpace MapReduce [\[link\]](#)

Qual a mágica do Spark?



MapReduce



**Resilient
Distributed
Datasets**



**Lazy
Evaluation**



Entenda mais: Why Spark DataFrame, lazy evaluation models outpace MapReduce [\[link\]](#)

Qual a mágica do Spark?

compartilhamento de
dados em operações de
processamento distribuído



**Lazy
Evaluation**



Entenda mais: Why Spark DataFrame, lazy evaluation models outpace MapReduce [\[link\]](#)

Qual a mágica do Spark?

compartilhamento de
dados em operações de
processamento distribuído

+

**Lazy
Evaluation**



Entenda mais: Why Spark DataFrame, lazy evaluation models outpace MapReduce [\[link\]](#)

Qual a mágica do Spark?

compartilhamento de
dados em operações de
processamento distribuído

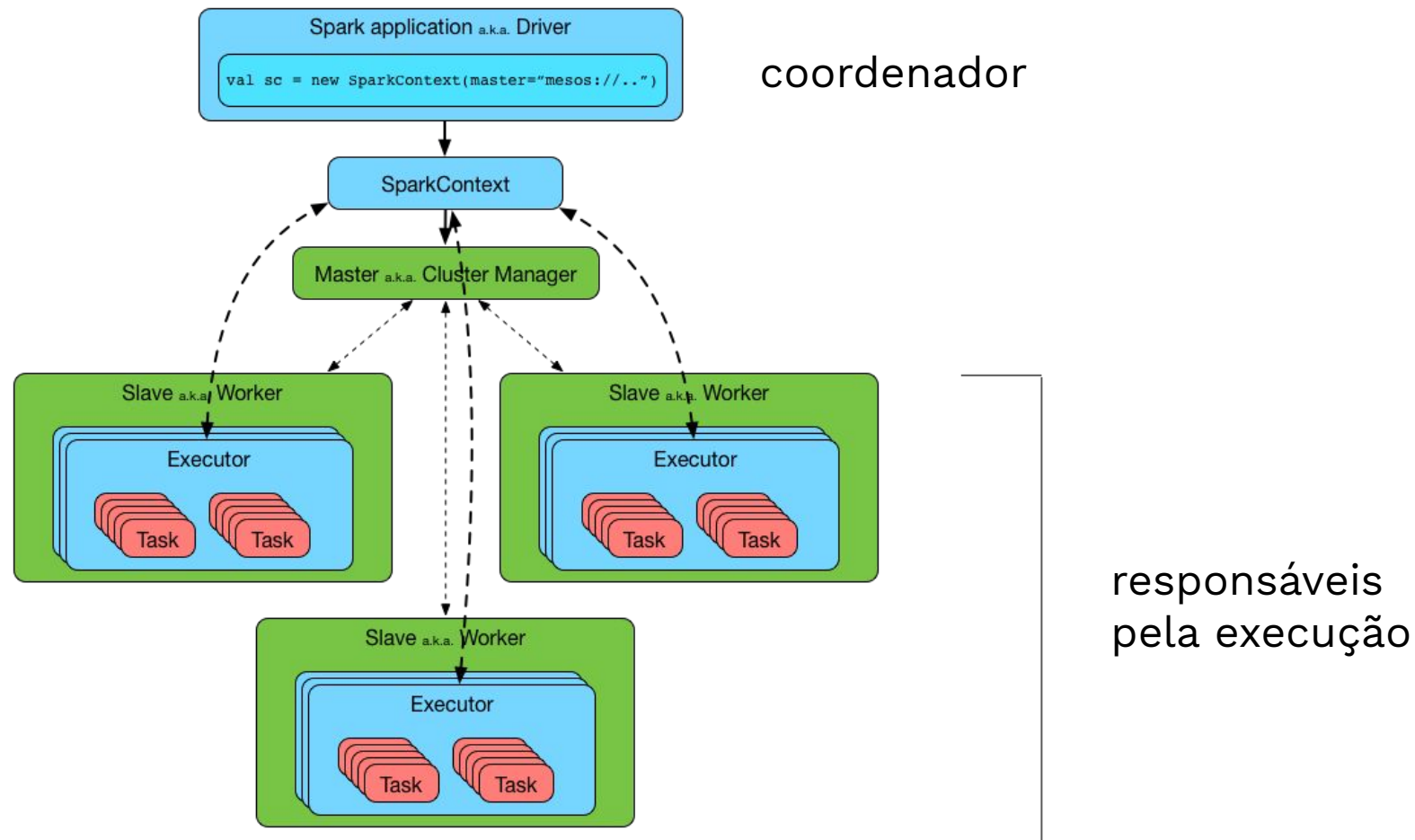


otimização
da execução



Entenda mais: Why Spark DataFrame, lazy evaluation models outpace MapReduce [\[link\]](#)

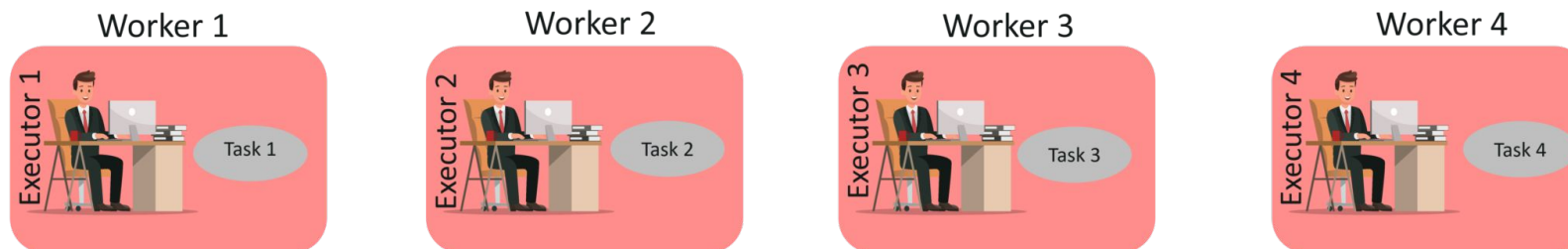
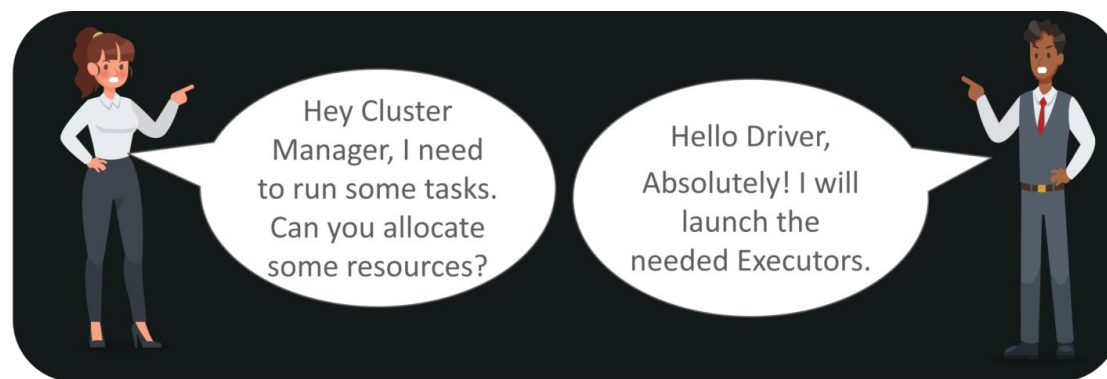
Como o Spark funciona?



Saiba mais: The internals of Apache Spark - Spark Architecture [\[link\]](#)



Como o Spark funciona?



Quando usar o Spark?

Spark deve ser usado para processamento de **grandes** volumes dados



Características de datasets grandes: Salganik, Matthew. Bit by bit: Social research in the digital age. Princeton University Press, 2019. [\[link para versão antiga\]](#)

```
+-----+-----+
|           nome|idade|
+-----+-----+
|fulanoA de tal|   15|
|fulanoB de tal|   20|
|fulanoC de mal|   12|
+-----+-----+
```

```
[8]: 15.666666666666666
```



```
[1]: import pandas as pd

[2]: data = {'nome': ['fulanoA de tal', 'fulanoB de tal', 'fulanoC de mal'], 'idade': [15, 20, 12]}

[3]: df = pd.DataFrame(data)
```

```
[4]: df
```

	nome	idade
0	fulanoA de tal	15
1	fulanoB de tal	20
2	fulanoC de mal	12

```
[5]: spark_df = spark.createDataFrame(df)

[6]: spark_df.show()
```

```
+-----+-----+
|      nome|idade|
+-----+-----+
|fulanoA de tal|  15|
|fulanoB de tal|  20|
|fulanoC de mal|  12|
+-----+-----+
```

Comparação de tempo para cálculo da média de idade

```
[7]: %%time
df['idade'].mean()

CPU times: user 176 µs, sys: 92 µs, total: 268 µs
Wall time: 273 µs

[7]: 15.666666666666666
```

```
[8]: %%time
spark_df.select('idade').groupBy().mean().collect()[0][0]

CPU times: user 7.07 ms, sys: 35 µs, total: 7.11 ms
Wall time: 330 ms

[8]: 15.666666666666666
```

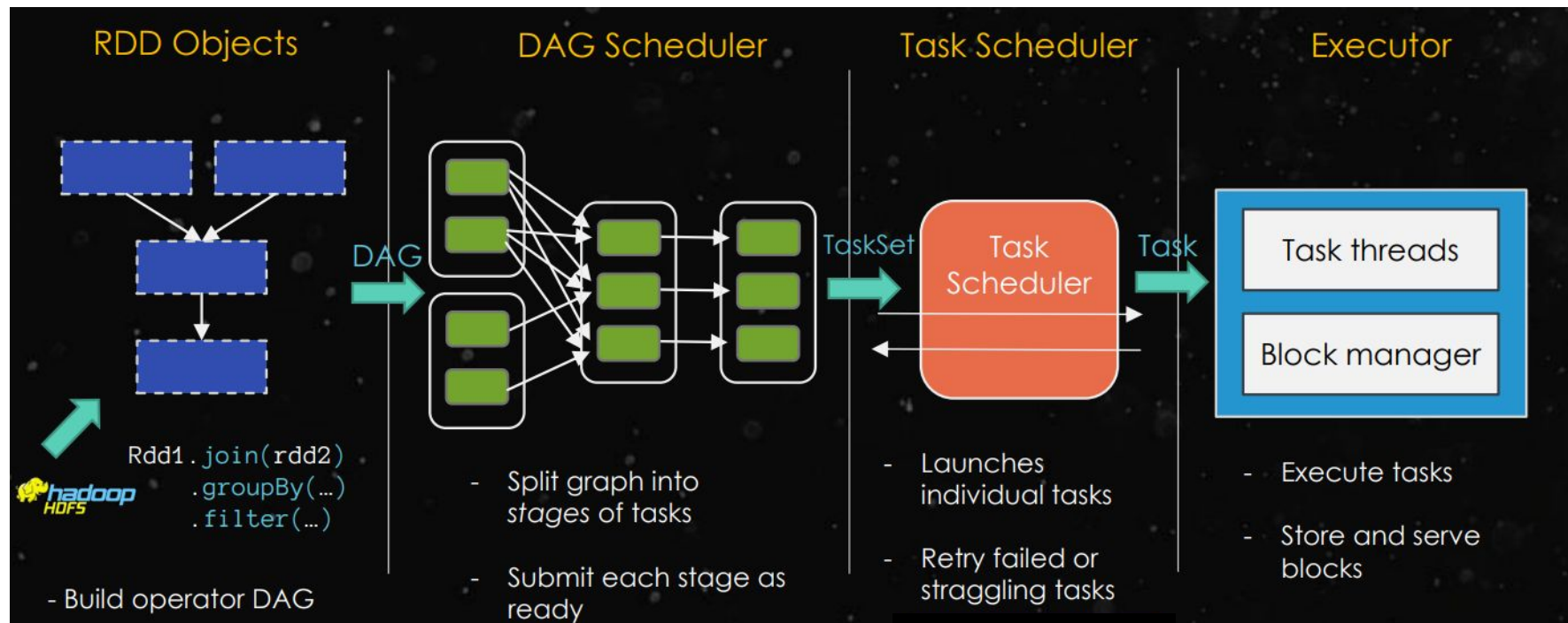
Resultado:

Pandas: $273 \times 10^{-6} \text{ s}$

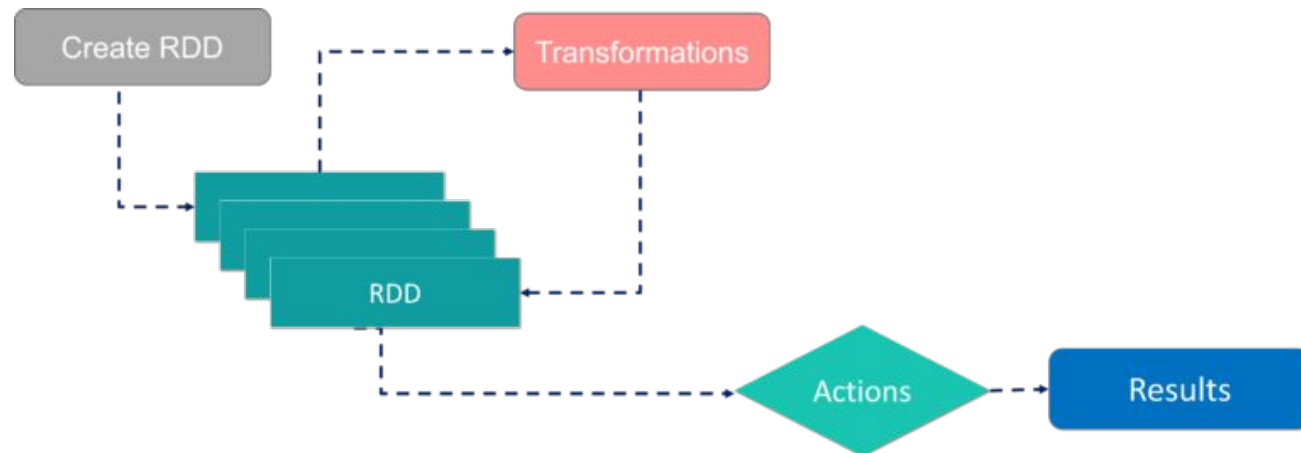
vs

Spark: $330 \times 10^{-3} \text{ s}$

Como o Spark funciona?



Como o Spark funciona?



Ações e transformações

Ações são operações com RDDs como input, mas que não retornam RDDs

Transformações são operações que operam com RDDs e retornam RDDs



Ações e transformações

Ações são avaliadas no momento em que são chamadas (strict evaluation)


Transformações *não* são avaliadas no momento em que foram chamadas (lazy evaluation)



Para ficar de olho no Spark...



Spark UI

 2.4.3

Jobs | Stages | Storage | Environment | Executors | SQL | pyspark-shell application UI

Details for Job 6

Status: SUCCEEDED
Completed Stages: 1

- ▶ Event Timeline
- ▶ DAG Visualization

▼ Completed Stages (1)

Stage Id ▼	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
6	collect at <ipython-input-15-448831e7b971>:1 <small>+details</small>	2019/11/01 00:55:45	82 ms	8/8				

Em geral, está acessível na porta **4040**

A dimly lit office scene. In the foreground, a woman with long dark hair is seated at a desk, viewed from the side. She is resting her chin on her hand and has her other hand on the trackpad of a silver laptop. To her left is a large black computer monitor. On the desk, there are some papers and a red pen. In the background, another person with long hair is seated at a similar desk, working on a laptop. The overall atmosphere is quiet and professional.

PARTE PRÁTICA

Onde e como rodar? Spark-shell


```
cina@positivamente ~$ pyspark
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
19/11/02 23:01:19 WARN Utils: Your hostname, positivamente resolves to a loopback address: 127.0.1.1; using 192.168.0.123 instead (on interface wlp58s0)
19/11/02 23:01:19 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
19/11/02 23:01:19 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
19/11/02 23:01:21 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Welcome to

      /--\
     /__ \
    /__ \|
   /__ \|
  /__ \|
 /__ \|
/_ \|
 version 2.4.4

Using Python version 3.7.4 (default, Aug 13 2019 20:35:49)
SparkSession available as 'spark'.
>>>
```


Parte 1

Onde e como rodar? Jupyter



The screenshot shows a Jupyter Notebook window titled "demo_jupyter_kernel_pytho". The interface includes a toolbar with icons for saving, adding, deleting, and running code, along with a "Code" dropdown menu. The Python version is indicated as "Python 3".

Below the toolbar, there is a text instruction: "Descomente a linha abaixo e coloque o caminho onde o Spark foi instalado, caso você não tenha setado a variável de ambiente \$SPARK_HOME".

The notebook contains four code cells:

```
[ ]: # import os
    # os.environ['SPARK_HOME'] = '/opt/spark-2.4.4-bin-hadoop2.7'
```

```
[2]: from pyspark.sql import SparkSession
```

```
[3]: spark = SparkSession \
    .builder \
    .appName("MyFirstSparkSession") \
    .getOrCreate()
```

```
[4]: spark
```

The output of the fourth cell is displayed below the code:

```
[4]: SparkSession - in-memory

SparkContext

Spark UI

Version
  v2.4.4

Master
  local[*]

AppName
  MyFirstSparkSession
```

Parte 1

Onde e como rodar? **spark-submit**

```
> spark-submit exemplo_spark_submit/demo_script.py
```

```
+-----+-----+
|      nome|idade|
+-----+-----+
|fulanoA de tal| 15|
|fulanoB de tal| 20|
|fulanoC de mal| 12|
+-----+-----+
```

```
19/11/02 23:28:05 INFO SparkContext: Invoking stop() from shutdown hook
19/11/02 23:28:05 INFO SparkUI: Stopped Spark web UI at http://192.168.0.123:4042
19/11/02 23:28:05 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
19/11/02 23:28:05 INFO MemoryStore: MemoryStore cleared
19/11/02 23:28:05 INFO BlockManager: BlockManager stopped
19/11/02 23:28:05 INFO BlockManagerMaster: BlockManagerMaster stopped
19/11/02 23:28:05 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
19/11/02 23:28:05 INFO SparkContext: Successfully stopped SparkContext
19/11/02 23:28:05 INFO ShutdownHookManager: Shutdown hook called
19/11/02 23:28:05 INFO ShutdownHookManager: Deleting directory /tmp/spark-5fb83489-bfb0-4d43-aa00-375eef02a1b5
19/11/02 23:28:05 INFO ShutdownHookManager: Deleting directory /tmp/spark-5fb83489-bfb0-4d43-aa00-375eef02a1b5/pyspark-56f49b32-b6d7-4c7e-8fb7-6303f7ba48cd
19/11/02 23:28:05 INFO ShutdownHookManager: Deleting directory /tmp/spark-4ec42b35-fe50-4d51-b851-88bdfdc9faaa
```

Parte 2

Configurações da SparkSession

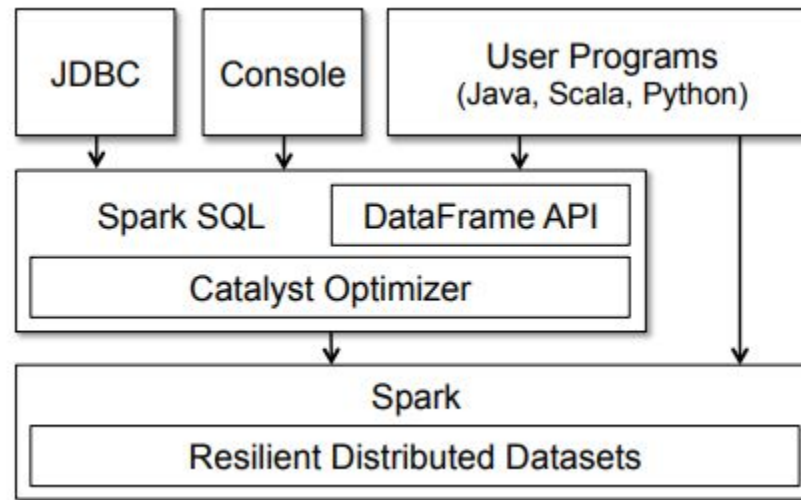
- ▷ ver exemplo no [README] dessa aula
- ▷ adição de packages [\[maven-central\]](#)



Veja mais opções: [\[documentação\]](#)

Parte 3

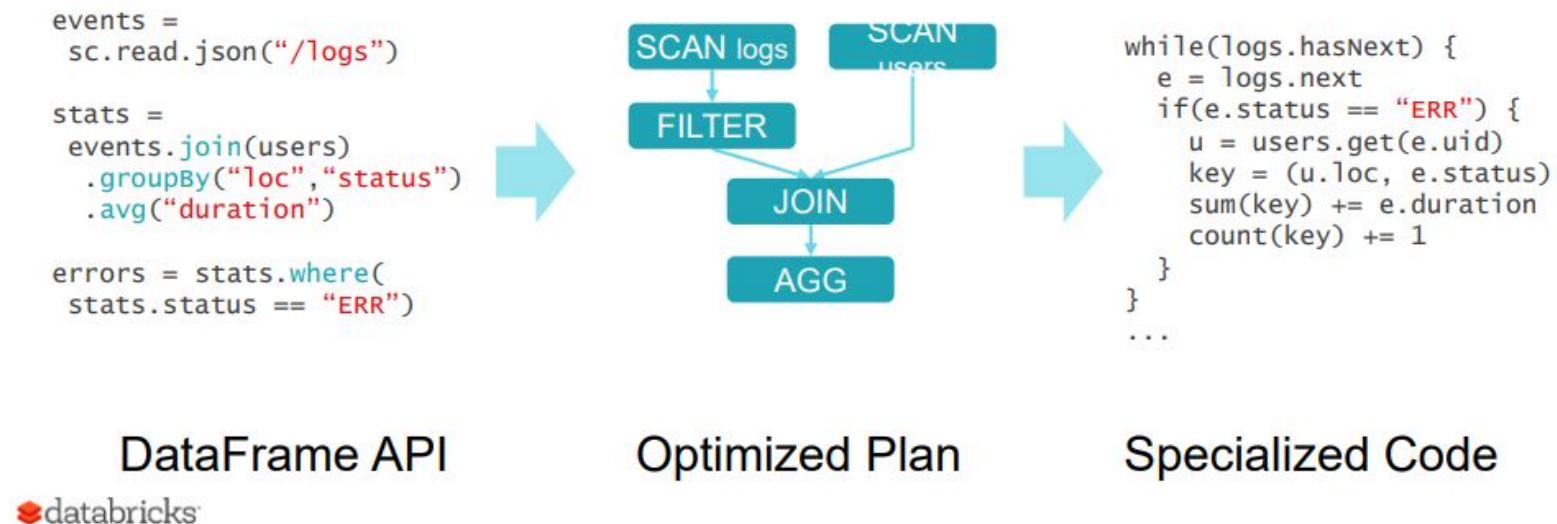
Spark SQL vs. DataFrame API



Entenda mais: SQL at Scale with Apache Spark SQL and DataFrames [\[link\]](#)

Parte 3

SparkSQL vs. DataFrame API




Entenda mais: SQL at Scale with Apache Spark SQL and DataFrames [\[link\]](#)

Parte 4

User Defined Functions

Para definir funções customizadas

- ▷ retorno da função deve ser um tipo nativo (ex.: tipo do *numpy* não vale)
- ▷  use com moderação



Entenda mais: User-Defined Functions - Python [\[link\]](#)

Parte 5

Particionamento

Tutorial em [\[datanoon\]](#)



Entenda mais sobre otimização no Spark:

Apache Spark Core—Deep Dive—Proper Optimization [\[link\]](#)

Extra: fontes de dados

- ▷ leitura de dados de um banco MySql ([exemplo em Python](#))
- ▷ leitura e escrita de dados no [Hive](#)
- ▷ streaming de dados de/para tópico do [Kafka](#)

Alternativa ao Spark

- ▷ [Dask](#): Scalable analytics in Python



Veja uma comparação entre dask e Spark:

Adapting from Spark to Dask: what to expect [\[link\]](#)

Obrigada!

Cynthia

Dúvidas?

 cinthia-tanaka

cimarie@gmail.com