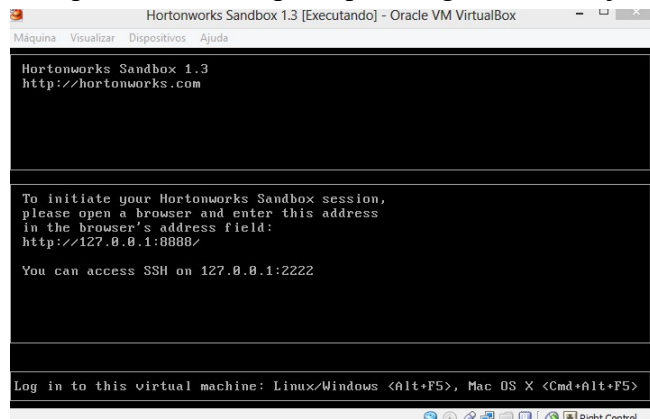


Professor: Cláudio Lúcio
Atividade Verificando processos e RDD's

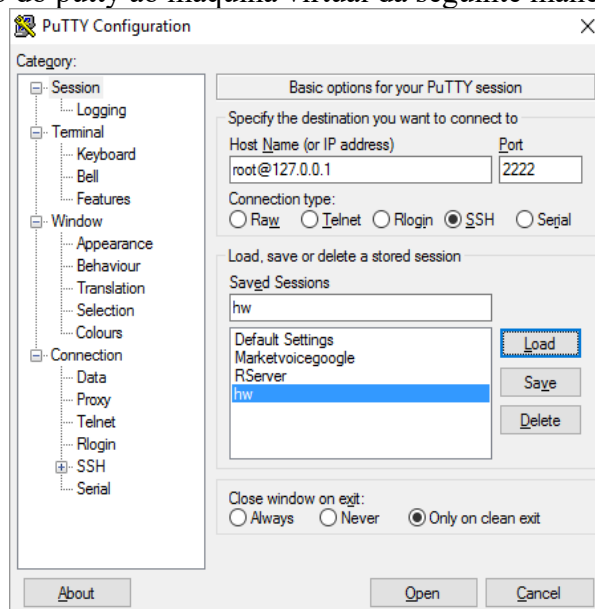
- 1) Para acesso ao Shell do Spark vamos usar a versão do Spark que vem na máquina virtual do HortonWorks:

Esta é a versão 2.3.2 do produto e pode ser obtida em:
<http://hortonworks.com/products/ Hortonworks-Sandbox/>

- 2) Inicialize a máquina virtual. Espere que a seguinte tela seja exibida:



- 3) A recomendação é usar um cliente para acesso SSH ao servidor do spark. Baixe a ferramenta putty.exe;
- 4) Configure o acesso do putty ao máquina virtual da seguinte maneira:



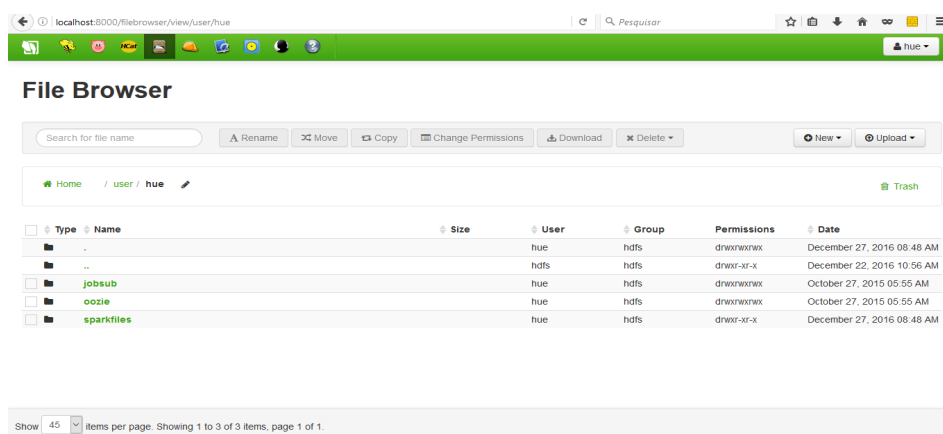
- 5) Clique em “Open” e então digite a senha do Root: hadoop

```
root@sandbox:~  
Using username "root".  
root@127.0.0.1's password:  
Last login: Fri Dec 23 16:56:09 2016 from 10.0.2.2  
[root@sandbox ~]#
```

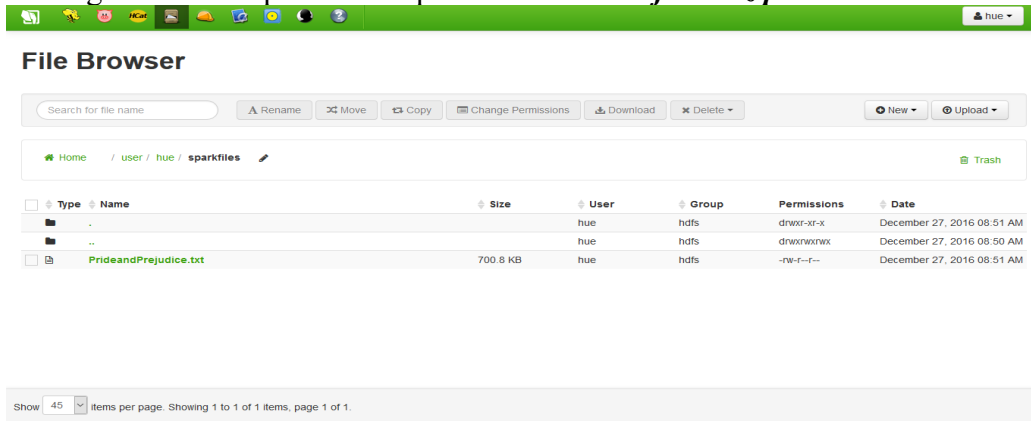
- 6) Primeiramente vamos interagir com a interface python para o Spark, para tal vamos acessar o bin/pyspark:
- 7) Digite os seguintes comandos:
 - `cd /usr/hdp/2.3.2.0-2950/spark/bin`
 - `pyspark`

```
root@sandbox:/usr/hdp/2.3.2.0-2950/spark/bin  
16/12/26 17:25:39 WARN SparkConf: The configuration key 'spark.yarn.applicationMaster.waitTries' has been deprecated as of Spark 1.3 and and may be removed in the future. Please use the new key 'spark.yarn.am.waitTime' instead.  
16/12/26 17:25:39 WARN SparkConf: The configuration key 'spark.yarn.applicationMaster.waitTries' has been deprecated as of Spark 1.3 and and may be removed in the future. Please use the new key 'spark.yarn.am.waitTime' instead.  
16/12/26 17:25:39 INFO Executor: Starting executor ID driver on host localhost  
16/12/26 17:25:40 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 53908.  
16/12/26 17:25:40 INFO NettyBlockTransferService: Server created on 53908  
16/12/26 17:25:40 INFO BlockManagerMaster: Trying to register BlockManager  
16/12/26 17:25:40 INFO BlockManagerMasterEndpoint: Registering block manager localhost:53908 with 265.4 MB RAM, BlockManagerId(driver, localhost, 53908)  
16/12/26 17:25:40 INFO BlockManagerMaster: Registered BlockManager  
Welcome to  
  
██████████ version 1.4.1  
  
Using Python version 2.6.6 (r266:84292, Jul 23 2015 15:22:56)  
SparkContext available as sc, HiveContext available as sqlContext.  
>>>
```

- 8) Veja que estamos com a versão 1.4.1 do Spark;
- 9) Vamos agora fazer o upload de arquivo para o HDFS e posteriormente vamos processá-lo utilizando um programa Spark. Crie uma pasta chamada *sparkfiles* no diretório do usuário Hue:



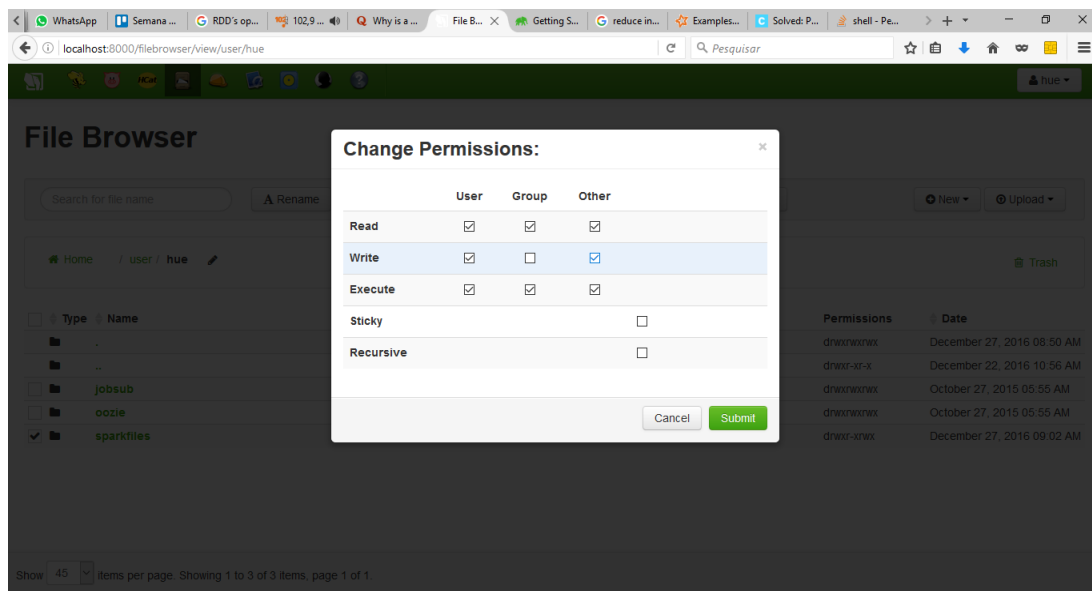
10) Vamos agora fazer o upload do arquivo *PrideandPrejudice.zip*



11) Adicione o direito de outros usuários escreverem neste diretório.

12) Marque o checkbox do diretório e clique no botão 'Change permissions'

13) Marque o checkbox da linha 'Write' e da coluna 'Other' e clique no botão 'submit'



14) Vamos agora entrar no pySpark e trabalhar com este arquivo

15) Veja o programa abaixo:

```
tokenized = sc.textFile('/user/hue/sparkfiles/PrideandPrejudice.txt').flatMap(lambda line:
line.split(" "))
```

```
allStopWords={'about':1, 'above':1, 'after':1, 'again':1, 'against':1, 'all':1, 'am':1,
'an':1, 'and':1, 'any':1, 'are':1, 'arent':1, 'as':1, 'at':1, 'be':1, 'because':1, 'been':1,
'before':1, 'being':1, 'below':1, 'between':1, 'both':1, 'but':1, 'by':1, 'cant':1,
'cannot':1, 'could':1, 'couldnt':1, 'did':1, 'didnt':1, 'do':1, 'does':1, 'doesnt':1,
'doing':1, 'dont':1, 'down':1, 'during':1, 'each':1, 'few':1, 'for':1, 'from':1, 'further':1,
'had':1, 'hadnt':1, 'has':1, 'hasnt':1, 'have':1, 'havent':1, 'having':1, 'he':1, 'hed':1,
'hell':1, 'hes':1, 'her':1, 'here':1, 'heres':1, 'hers':1, 'herself':1, 'him':1, 'himself':1,
'his':1, 'how':1, 'hows':1, 'i':1, 'id':1, 'ill':1, 'im':1, 'ive':1, 'if':1, 'in':1,
'into':1, 'is':1, 'isnt':1, 'it':1, 'its':1, 'its':1, 'itself':1, 'lets':1, 'me':1, 'more':1,
'most':1, 'mustnt':1, 'my':1, 'myself':1, 'no':1, 'nor':1, 'not':1, 'of':1, 'off':1, 'on':1,
'once':1, 'only':1, 'or':1, 'other':1, 'ought':1, 'our':1, 'ours':1, 'ourselves':1, 'out':1,
'over':1, 'own':1, 'same':1, 'shant':1, 'she':1, 'shed':1, 'shell':1, 'shes':1, 'should':1,
'shouldnt':1, 'so':1, 'some':1, 'such':1, 'than':1, 'that':1, 'thats':1, 'the':1, 'their':1,
'theirs':1, 'them':1, 'themselves':1, 'then':1, 'there':1, 'theres':1, 'these':1, 'they':1,
'theyd':1, 'theyll':1, 'theyre':1, 'theyve':1, 'this':1, 'those':1, 'through':1, 'to':1,
'too':1, 'under':1, 'until':1, 'up':1, 'very':1, 'was':1, 'wasnt':1, 'we':1, 'wed':1,
'well':1, 'were':1, 'weve':1, 'were':1, 'werent':1, 'what':1, 'whats':1, 'when':1, 'whens':1,
'where':1, 'wheres':1, 'which':1, 'while':1, 'who':1, 'whos':1, 'whom':1, 'why':1, 'whys':1,
'with':1, 'wont':1, 'would':1, 'wouldnt':1, 'you':1, 'youd':1, 'youll':1, 'youre':1,
'youve':1, 'your':1, 'yours':1, 'yourself':1, 'yourselves':1}
```

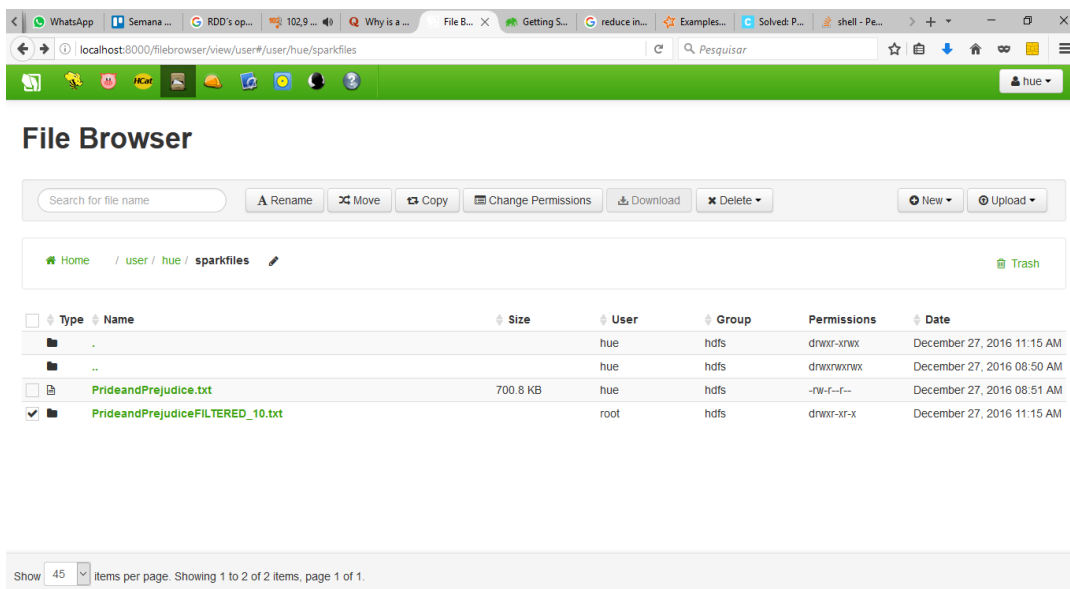
```
# contar a ocorrencia das palavras com alguns tratamentos
wordCounts = tokenized.map( lambda x: x.replace(',',' ').replace('.', ' ').replace('-', ' ')
.lower()) \
    .flatMap(lambda x: x.split()) \
    .filter(lambda x: x not in allStopWords ) \
    .map(lambda x: (x, 1)) \
    .reduceByKey(lambda x,y:x+y) \
    .map(lambda x:(x[1],x[0])) \
    .sortByKey(False)

wordCounts.cache()
wordCounts.count()
wordCounts.take(10)

threshold = 10
filtered = wordCounts.filter(lambda pair:pair[0] >= threshold)
filtered.count()
filtered.first()

filtered.saveAsTextFile("/user/hue/sparkfiles/PrideandPrejudiceFILTERED_"+str(threshold)
+".txt")
```

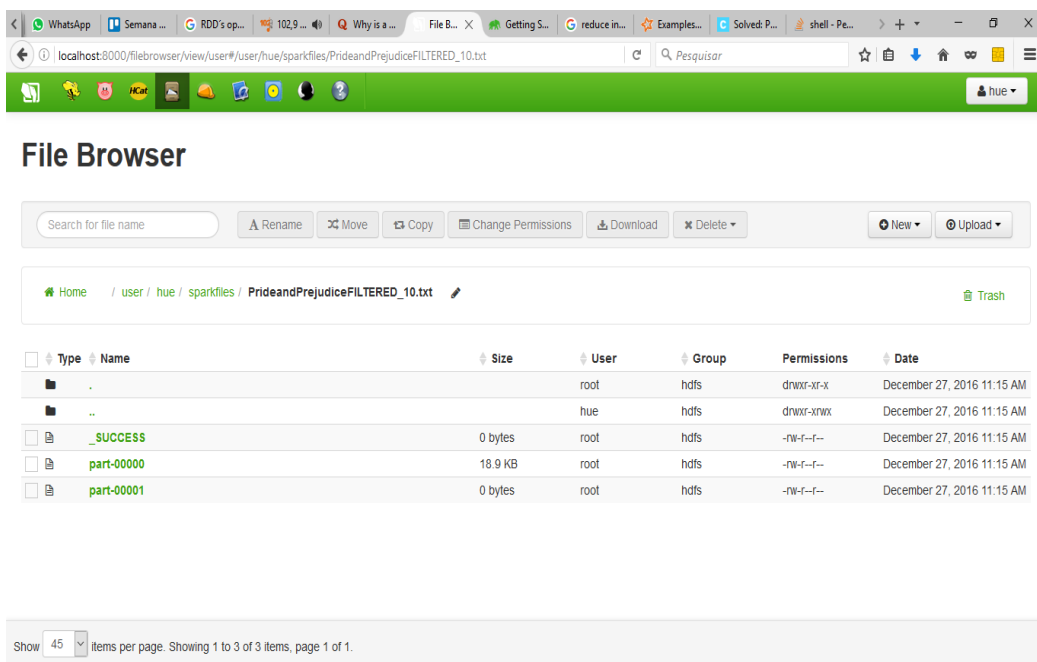
16) Execute o código acima e veja se o arquivo final será criado:



The screenshot shows the Hue File Browser interface. The breadcrumb path is `Home / user / hue / sparkfiles`. The file list shows:

Type	Name	Size	User	Group	Permissions	Date
Folder	.		hue	hdfs	drwxr-xrwx	December 27, 2016 11:15 AM
Folder	..		hue	hdfs	drwxrwxrwx	December 27, 2016 08:50 AM
File	PrideandPrejudice.txt	700.8 KB	hue	hdfs	-rw-r--r--	December 27, 2016 08:51 AM
File	PrideandPrejudiceFILTERED_10.txt		root	hdfs	drwxr-xr-x	December 27, 2016 11:15 AM

At the bottom, it says "Show 45 items per page. Showing 1 to 2 of 2 items, page 1 of 1."

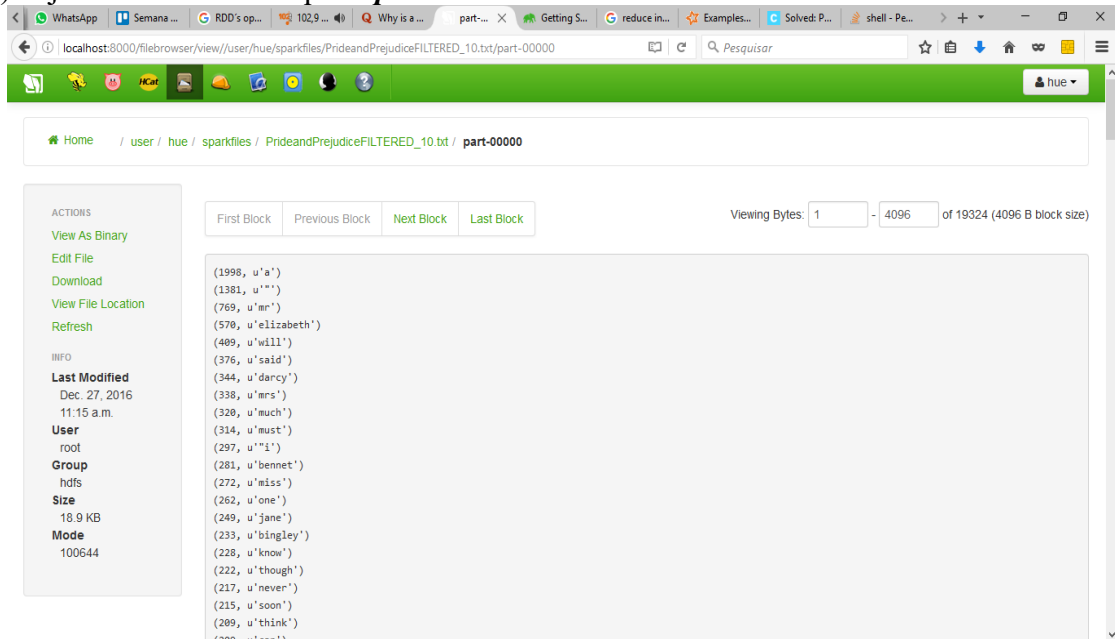


The screenshot shows the Hue File Browser interface with the breadcrumb path `Home / user / hue / sparkfiles / PrideandPrejudiceFILTERED_10.txt`. The file list shows:

Type	Name	Size	User	Group	Permissions	Date
Folder	.		root	hdfs	drwxr-xr-x	December 27, 2016 11:15 AM
Folder	..		hue	hdfs	drwxr-xrwx	December 27, 2016 11:15 AM
File	._SUCCESS	0 bytes	root	hdfs	-rw-r--r--	December 27, 2016 11:15 AM
File	part-00000	18.9 KB	root	hdfs	-rw-r--r--	December 27, 2016 11:15 AM
File	part-00001	0 bytes	root	hdfs	-rw-r--r--	December 27, 2016 11:15 AM

At the bottom, it says "Show 45 items per page. Showing 1 to 3 of 3 items, page 1 of 1."

17) Veja o conteúdo do arquivo *part-0000*



Home / user / hue / sparkfiles / PrideandPrejudiceFILTERED_10.txt / part-0000

ACTIONS

- View As Binary
- Edit File
- Download
- View File Location
- Refresh

INFO

Last Modified: Dec. 27, 2016 11:15 a.m.

User: root

Group: hdfs

Size: 18.9 KB

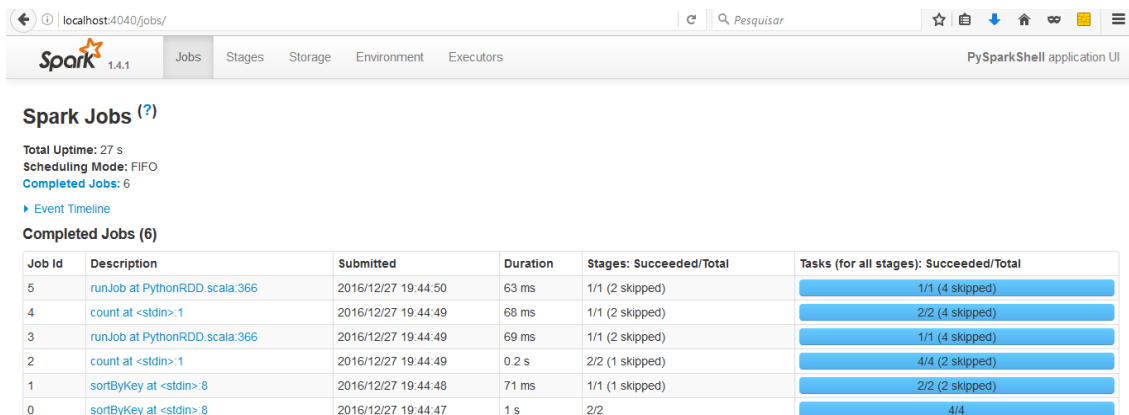
Mode: 100644

First Block Previous Block Next Block Last Block

Viewing Bytes: 1 - 4096 of 19324 (4096 B block size)

```
(1998, u'a')
(1381, u''')
(769, u'mr')
(570, u'elizabeth')
(489, u'will')
(376, u'said')
(344, u'darcy')
(338, u'mrs')
(320, u'much')
(314, u'must')
(297, u'i')
(281, u'bennet')
(272, u'miss')
(262, u'one')
(249, u'jane')
(233, u'bingley')
(228, u'know')
(222, u'though')
(217, u'never')
(215, u'soon')
(209, u'think')
(200, u'can')
```

18) Quando o pyspark é utilizado, de maneira local, como neste exemplo o Spark irá também apresentar uma interface para visualizar os jobs submetidos. Digite o endereço <http://localhost:4040> no browser:



Spark 1.4.1

Jobs Stages Storage Environment Executors

PySparkShell application UI

Spark Jobs (?)

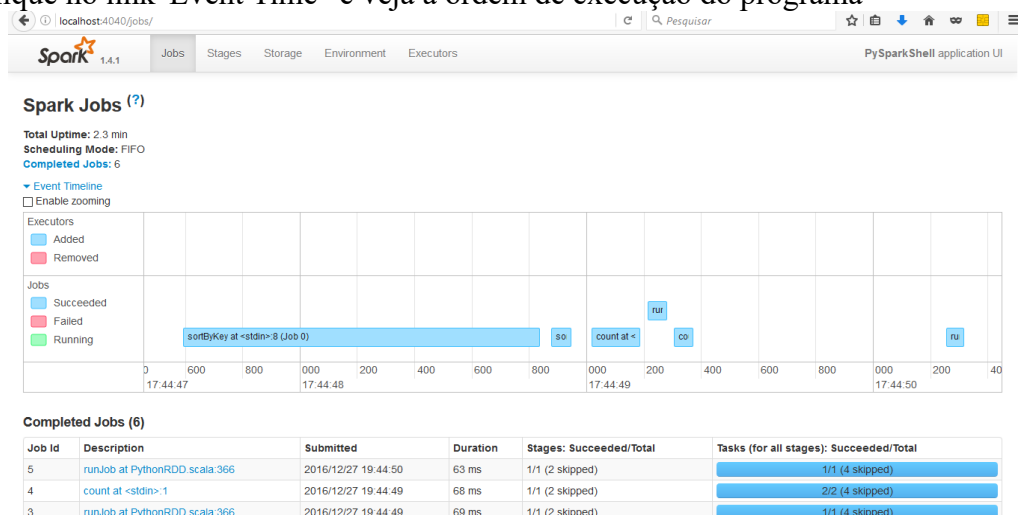
Total Uptime: 27 s
Scheduling Mode: FIFO
Completed Jobs: 6

Event Timeline

Completed Jobs (6)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
5	runJob at PythonRDD scala:366	2016/12/27 19:44:50	63 ms	1/1 (2 skipped)	1/1 (4 skipped)
4	count at <stdin>:1	2016/12/27 19:44:49	68 ms	1/1 (2 skipped)	2/2 (4 skipped)
3	runJob at PythonRDD scala:366	2016/12/27 19:44:49	69 ms	1/1 (2 skipped)	1/1 (4 skipped)
2	count at <stdin>:1	2016/12/27 19:44:49	0.2 s	2/2 (1 skipped)	4/4 (2 skipped)
1	sortByKey at <stdin>:8	2016/12/27 19:44:48	71 ms	1/1 (1 skipped)	2/2 (2 skipped)
0	sortByKey at <stdin>:8	2016/12/27 19:44:47	1 s	2/2	4/4

19) Clique no link 'Event Time ' e veja a ordem de execução do programa



Spark 1.4.1

Jobs Stages Storage Environment Executors

PySparkShell application UI

Spark Jobs (?)

Total Uptime: 2.3 min
Scheduling Mode: FIFO
Completed Jobs: 6

Event Timeline

Enable zooming

Executors

- Added
- Removed

Jobs

- Succeeded
- Failed
- Running

sortByKey at <stdin>:8 (Job 0)

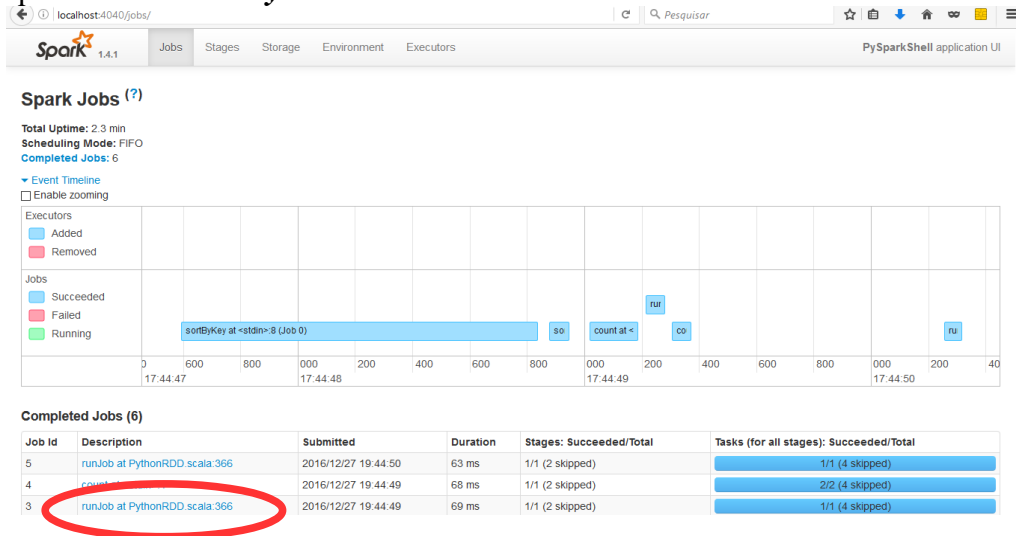
count at <stdin>:1

runJob at PythonRDD scala:366

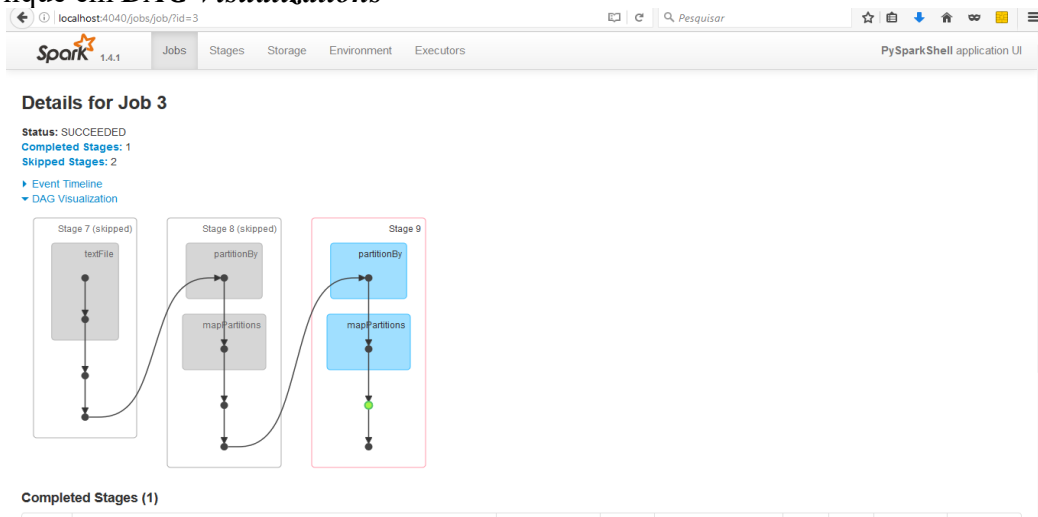
Completed Jobs (6)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
5	runJob at PythonRDD scala:366	2016/12/27 19:44:50	63 ms	1/1 (2 skipped)	1/1 (4 skipped)
4	count at <stdin>:1	2016/12/27 19:44:49	68 ms	1/1 (2 skipped)	2/2 (4 skipped)
3	runJob at PythonRDD scala:366	2016/12/27 19:44:49	69 ms	1/1 (2 skipped)	1/1 (4 skipped)

20) Clique em *runJob at PythonRDD scala:366*...



21) Clique em *DAG Visualizations*



22) Clique na aba 'Stages' em que cada um dos passos é apresentado de forma mais detalhada:

Stages for All Jobs

Completed Stages: 8

Completed Stages (8)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
15	runJob at PythonRDD scala:366	2016/12/27 19:44:50	52 ms	1/1	38.9 KB			
12	count at <stdin>:1	2016/12/27 19:44:49	58 ms	2/2	105.4 KB			
9	runJob at PythonRDD scala:366	2016/12/27 19:44:49	57 ms	1/1	38.9 KB			
6	count at <stdin>:1	2016/12/27 19:44:49	65 ms	2/2			121.2 KB	
5	sortByKey at <stdin>:8	2016/12/27 19:44:49	78 ms	2/2			184.9 KB	121.2 KB
3	sortByKey at <stdin>:8	2016/12/27 19:44:48	61 ms	2/2			184.9 KB	
1	sortByKey at <stdin>:8	2016/12/27 19:44:48	73 ms	2/2			184.9 KB	
0	reduceByKey at <stdin>:6	2016/12/27 19:44:47	1 s	2/2	734.4 KB			184.9 KB

23) Agora veja os detalhes da primeira execução *runJob at PythonRDD Scala...* Clique neste link:

localhost:4040/stages/stage/?id=9&attempt=0

Spark 1.4.1 Jobs Stages Storage Environment Executors PySparkShell application UI

Details for Stage 9 (Attempt 0)

Total Time Across All Tasks: 44 ms
Input Size / Records: 38.9 KB / 12

[DAG Visualization](#)
[Show Additional Metrics](#)
[Event Timeline](#)

Summary Metrics for 1 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	44 ms	44 ms	44 ms	44 ms	44 ms
GC Time	0 ms	0 ms	0 ms	0 ms	0 ms
Input Size / Records	38.9 KB / 12	38.9 KB / 12	38.9 KB / 12	38.9 KB / 12	38.9 KB / 12

Aggregated Metrics by Executor

Executor ID	Address	Task Time	Total Tasks	Failed Tasks	Succeeded Tasks	Input Size / Records
driver	localhost:44973	56 ms	1	0	1	38.9 KB / 12

Tasks

Index	ID	Attempt	Status	Locality Level	Executor ID / Host	Launch Time	Duration	GC Time	Input Size / Records	Errors
0	10	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/12/27 19:44:49	44 ms		38.9 KB (memory) / 12	

24) Clique no item *DAG Visualization*

localhost:4040/stages/stage/?id=9&attempt=0

Spark 1.4.1 Jobs Stages Storage Environment Executors PySparkShell application UI

Details for Stage 9 (Attempt 0)

Total Time Across All Tasks: 44 ms
Input Size / Records: 38.9 KB / 12

[DAG Visualization](#)

[Show Additional Metrics](#)
[Event Timeline](#)

Summary Metrics for 1 Completed Tasks

25) Outra visualização é dos RDD's, clique em '*Storage*'

localhost:4040/storage/

Spark 1.4.1 Jobs Stages Storage Environment Executors PySparkShell application UI

Storage

RDD Name	Storage Level	Cached Partitions	Fraction Cached	Size in Memory	Size in ExternalBlockStore	Size on Disk
PythonRDD	Memory Serialized 1x Replicated	2	100%	105.4 KB	0.0 B	0.0 B

26) Clique em Python RDD para ver as partições:

localhost:4040/storage/rdd/?id=12

Spark 1.4.1 Jobs Stages Storage Environment Executors PySparkShell application UI

RDD Storage Info for PythonRDD

Storage Level: Memory Serialized 1x Replicated
Cached Partitions: 2
Total Partitions: 2
Memory Size: 105.4 KB
Disk Size: 0.0 B

Data Distribution on 1 Executors

Host	Memory Usage	Disk Usage
localhost:44973	105.4 KB (265.3 MB Remaining)	0.0 B

2 Partitions

Block Name	Storage Level	Size in Memory	Size on Disk	Executors
rdd_12_0	Memory Serialized 1x Replicated	38.9 KB	0.0 B	localhost:44973
rdd_12_1	Memory Serialized 1x Replicated	66.5 KB	0.0 B	localhost:44973

27) Por ultimo apague todo o diretório criado, utilizando seguinte linha de comando:

```
hadoop fs -rmr /user/hue/sparkfiles ou  
hadoop fs rm -r /user/hue/sparkfiles
```