Apache Spark

Защитная зона для интеграции видео спикера



Аналитик-разработчик, Яндекс



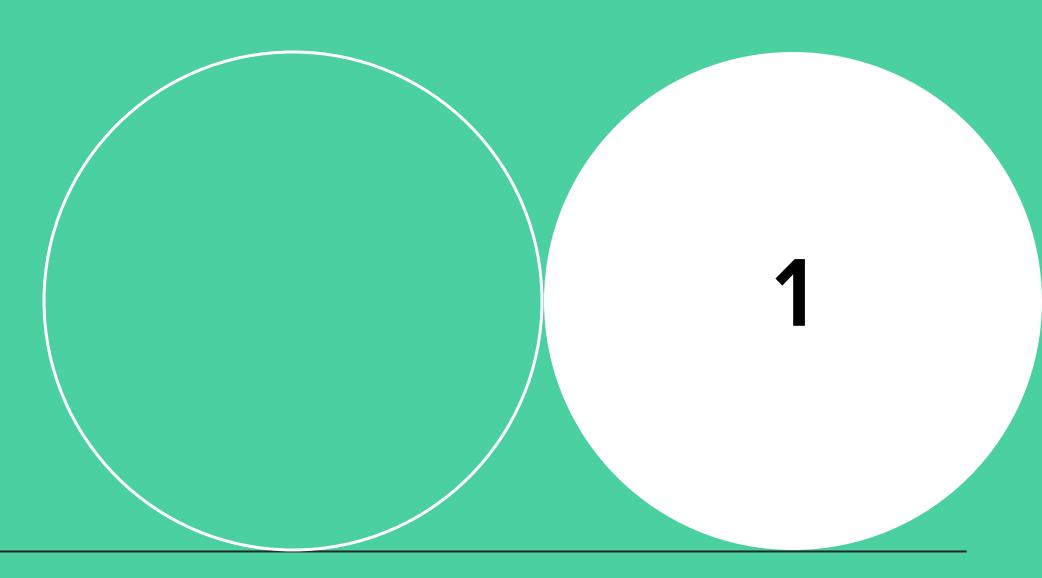


Содержание

- 0 Еще один инструмент ура
- 1 Установка
- 2 RDD и ленивые вычисления
- 3 Основные команды



Зачем нам еще один инструмент





Что имеем

- **★** Mapreduce
- **★** Hive
- **★** Pig
- **★** HBase
- **★** Cassandra



Нужно больше!

- Марreduce на десятки сложных операций
- Чтобы сразу со сложными функциями
- ❖ SQL или команды как в pandas
- Использовать машинное обучение на больших данных
- Обработка потоков данных
- Работа с графами



- YARN, Mesos, Kubernetes, Amazon
- Чтобы локально можно было гонять данные
- Нужно разные языки использовать
- Данные бывают из текстовых файлов
- Внешние плагины тоже не помешают



Apache Spark

- ★ Сложный Mapreduce RDD
- ★ SQL и аналоги pandas Spark SQL
- ★ Потоки данных Spark Streaming
- ★ Машинное обучение MLlib
- ★ Графы GraphX
- ★ Внешние проекты Third-Party Projects



Mapreduce

```
Защитная зона
для интеграции
видео спикера
```

```
data
.map(lambda x: (x[0].split(' '), x[1]))
.flatMap(lambda x: [(i, x[1]) for i in x[0]])
.saveAsTextFile('/opt/bitnami/spark/output/flatmap')
)
```



SQL & dataframes

```
#Showing the data
df.show()
```

```
|Company| Person|Sales|
          Sam|200.0|
   GOOG |
   G00G|Charlie|120.0|
   G00G| Frank|340.0|
          Tina|600.0|
   MSFT|
   MSFT|
          Amy | 124.0|
   MSFT|Vanessa|243.0|
          Carl|870.0|
     FB|
          Sarah|350.0|
           John | 250.0 |
   APPL|
          Linda|130.0|
   APPL
           Mike|750.0|
   APPL |
   APPL|
          Chris | 350.0|
```

```
# Max
df.groupBy('Company').max().show()
```

```
+----+
|Company|max(Sales)|
+----+
| APPL| 750.0|
| G00G| 340.0|
| FB| 870.0|
| MSFT| 600.0|
+-----+
```



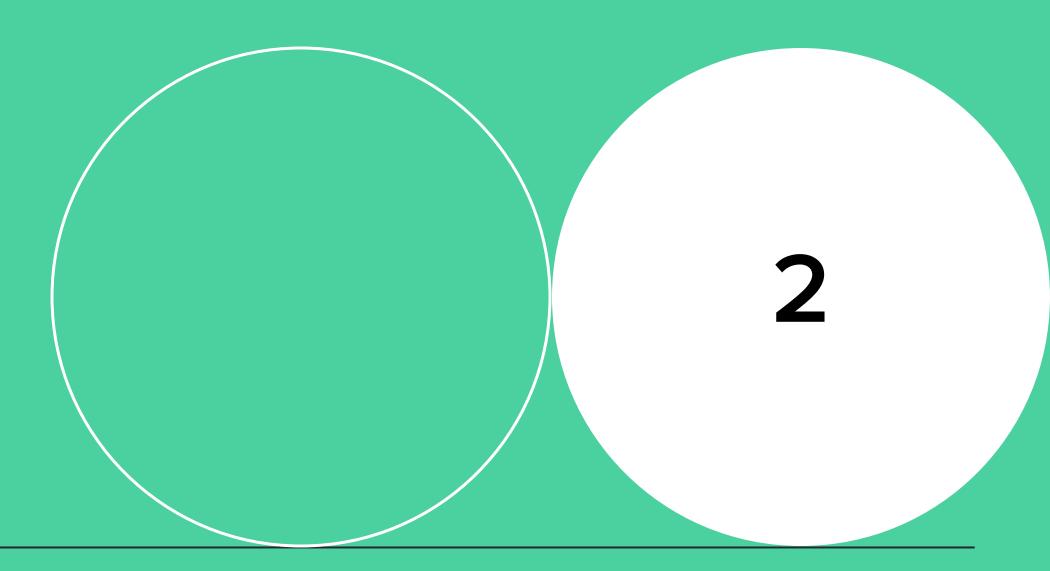
ML Pipelines

```
from pyspark.ml.linalg import Vectors
from pyspark.ml.classification import LogisticRegression
# Prepare training data from a list of (label, features) tuples.
training = spark.createDataFrame([
    (1.0, Vectors.dense([0.0, 1.1, 0.1])),
    (0.0, Vectors.dense([2.0, 1.0, -1.0])),
    (0.0, Vectors.dense([2.0, 1.3, 1.0])),
    (1.0, Vectors.dense([0.0, 1.2, -0.5]))], ["label", "features"])
# Create a LogisticRegression instance. This instance is an Estimator.
lr = LogisticRegression(maxIter=10, regParam=0.01)
# Print out the parameters, documentation, and any default values.
print("LogisticRegression parameters:\n" + lr.explainParams() + "\n")
# Learn a LogisticRegression model. This uses the parameters stored in lr.
model1 = lr.fit(training)
```





Про установку



Установка

Защитная зона для интеграции видео спикера

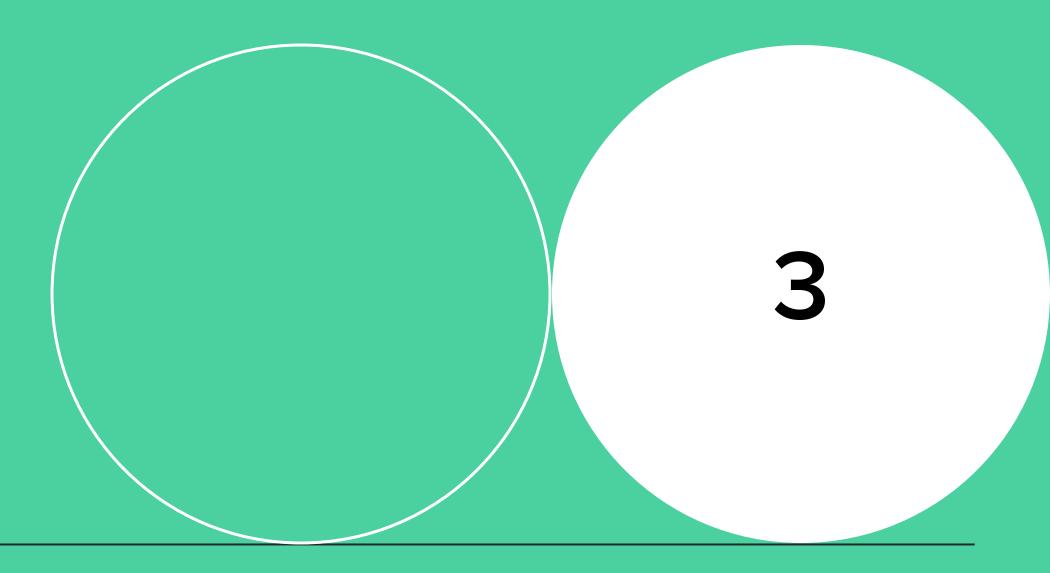
Используем образ <u>bitnami/spark</u>

Особенности установки:

- Нужна Java
- Настройка переменных окружения
- ❖ Если машин несколько, то нужна настройка master-worker
- ❖ По умолчанию будет Scala, используйте pyspark
- Pyspark это часть кластера



Mapreduce Ha Spark





RDD



Resilient Distributed Datasets

- Набор элементов из источника данных (file, HDFS, HBase)
- Устойчив к падениям процесса расчета
- Обрабатывается параллельно
- Используется концепция ленивых вычислений (!)



RDD из памяти

```
Защитная зона
для интеграции
видео спикера
```

```
data = [1, 2, 3]
data_rdd = sc.parallelize(data)

data_rdd.count()
# 3
```



Ленивые вычисления



Начинаются только когда очень попросят. Пример в python:

```
data = [11, 22, 33, 44, 55]

even_odd = map(lambda x: x % 2, data)

for result in even_odd:
    print(result)

1
0
1
0
1
```

Ленивые вычисления



Пример вычисления с ошибкой

```
In [4]: wrong_data = [11, 22, '33', 44, 55]
even_odd = map(lambda x: x % 2, wrong_data)
```

Но ячейка выполнилась без ошибки



Ленивые вычисления

Защитная зона для интеграции видео спикера

Ошибка будет только в самом процессе

```
for result in even odd:
    print(result)
                                          Traceback (most recent call last)
TypeError
<ipython-input-5-c71fd2e75322> in <module>
----> 1 for result in even odd:
         print(result)
<ipython-input-4-250809b68714> in <lambda>(x)
      1 wrong data = [11, 22, '33', 44, 55]
---> 2 even odd = map(lambda x: x % 2, wrong_data)
TypeError: not all arguments converted during string formatting
```



B RDD аналогично

```
Защитная зона
для интеграции
видео спикера
```

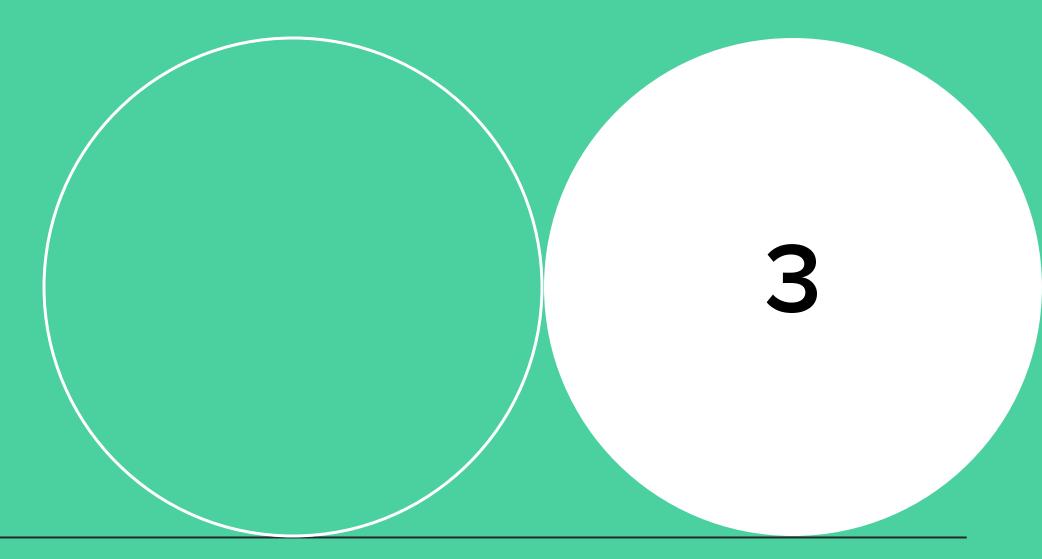
```
data = [11, 22, 33, 44, 55]
data_rdd = sc.parallelize(data)
data_rdd
# ParallelCollectionRDD[15] at readRDDFromFile at PythonRDD.scala:262
```

```
even_odd = data_rdd.map(lambda x: x % 2)
even_odd
# PythonRDD[16] at RDD at PythonRDD.scala:53
```

even_odd.take(5) # [1, 0, 1, 0, 1]



Основные команды





Для начала

Защитная зона для интеграции видео спикера

sc.textFile - сформировать RDD из текстового файла take(5) - посмотреть первые 5 элементов результата collect - возвращает результат вычислений в память (!)

count() - подсчет числа строк map - построчная обработка (маппер) flatMap - разворачивание списка в столбец

filter - фильтрация строк функцией reduce - попарные действия с элементами

