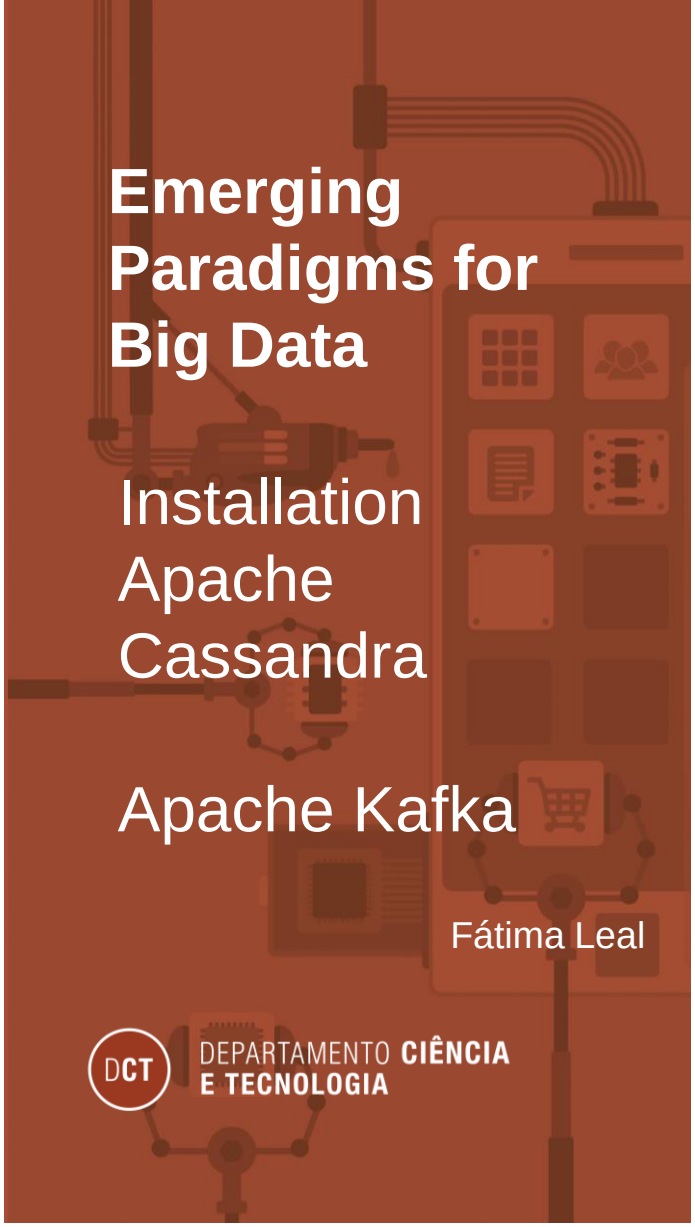

UNIVERSIDADE PORTUCALENSE

Content

- Installation Apache Cassandra
- Installation Apache Kafka
- Hands-on Activity

Big Data Solutions

- **Apache Kafka** is a real-time publish-subscribe solution messaging system: open source, distributed, partitioned, replicated, commit-log based with a publish-subscribe schema.

- **Apache Cassandra** is a NoSQL database management system designed to handle large amounts of data across many servers, providing high availability with no failures.

- **Apache Spark** is an infrastructure engine can be attached to powerful tools like Apache Kafka and Apache Cassandra to produce data science pipelines. Simultaneously, it is a data science laboratory because it represents an engine for machine learning in both a laptop and a productive cluster, from a few data kilobytes up to what the hardware capacity allows. Likewise, you can build models based on sample data and then apply them in larger datasets.

Big Data Solutions

Installation of Apache Cassandra

Installation of Apache Cassandra

- Installation of the Java Development Kit 1.8 – update 251
- Environment variables Definition
  - set JAVA_HOME = path of JDK
- Installation of Python 2.7
- Environment variables Definition
  - Path = path of Python2.7
- MacOS (ensure that you have the versions of JDK and python 2.7 in your mac)
  - Install brew
  - brew update
  - brew install cassandra

Installation of Apache Cassandra

- Download Apache Cassandra 3.11

- Installation Folder – Copy the Apache Cassandra unzip folder to
  - ”C:/Cassandra/”

- Environment variables Definition
  - CASSANDRA_HOME = path of Cassandra

- Open two terminals for the server and client
- Put both in the directory `cd %CASSANDRA_HOME%/bin`

- To run the server type `cassandra`
- To run the client type `cqlsh`

Check the version of java and python

-- java version "1.8.0_251“
-- Python 2.7.18

Cassandra Example

- In the client use the following commands:

```
create keyspace course with replication =
{'class':'SimpleStrategy','replication_factor':1};

describe keyspaces;

use courses;

create table student (id int primary key, name text);

select * from student;
```

- What are we creating?

Big Data Solutions

Installation of Apache Kafka

Installation of Apache Kafka

- **Java Development Kit 1.8 DONE**
- **Environment variables Definition DONE**
  - set JAVA_HOME = path of JDK
- **Download Apache Kafka 2.7**
- **Installation Folder**
  - Copy the Apache Kafka unzip folder to “C:/Kafka/“
- **Environment variables Definition**
  - KAFKA_HOME = path of Kafka

MacOS
brew install kafka

Installation of Apache Kafka

- **Create folders for Logs**
  - Create a folder "data" inside the Kafka folder
  - Inside the data folder create two folders "kafka" and "zookeeper“
- **Edit the configuration files**
  - Open the ”zookeeper.properties” with notepad
  - change ”dataDir=” to the path of the folder ”…/data/zookeeper” (introduce the entire path to avoid mistakes)
  - Open the ”server.properties” with notepad
  - change ”log.dirs=” to the path of the folder ”…/data/kafka” (introduce the entire path to avoid mistakes)

Installation of Apache Kafka

- **In Anaconda prompt type:**
  - `conda activate pyspark_env`
  - `pip install kafka-python`

- **Launch Zookeeper**
  - open a command line terminal and type:
    - `cd %KAFKA_HOME%/bin/windows`
    - `zookeeper-server-start.bat ../../config/zookeeper.properties`

- **Launch Kafka**
  - open a command line terminal and type:
    - `cd %KAFKA_HOME%/bin/windows`
    - `kafka-server-start.bat ../../config/server.properties`

Apache Kafka example Producer

```
from json import dumps
from time import sleep
from kafka import KafkaProducer

producer = KafkaProducer ( bootstrap_servers =['localhost:9092'],
value_serializer = lambda x : dumps ( x ).encode ( 'utf-8 '))

for e in range (1000) :
    data = {'number' : e }
    print(data)
    producer.send ('numtest' , value = data )
    sleep (5)
```

Apache Kafka example Consumer

```
from json import loads
from kafka import KafkaConsumer
print("a")
consumer = KafkaConsumer ( 'numtest' ,
bootstrap_servers=['localhost:9092'] , auto_offset_reset = 'earliest',
                          enable_auto_commit = True, group_id = 'my-
group',
                          value_deserializer = lambda x : loads (
x.decode ( 'utf-8') ) )
for message in consumer :
    message = message.value
    print ( message )
```

Apache Kafka example

- To see the data received by the consumer in the server:
  - open a command line terminal and type:
  - cd %KAFKA_HOME%/bin/windows
  - bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic numtest --from-beginning

Next Class

- Apache Kafka architecture
- Apache Kafka exercises

UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.