PySpark

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia https://twitter.com/tutorialspoint

About the Tutorial

Apache Spark is written in Scala programming language. To support Python with Spark, Apache Spark community released a tool, PySpark. Using PySpark, you can work with RDDs in Python programming language also. It is because of a library called Py4j that they are able to achieve this.

This is an introductory tutorial, which covers the basics of Data-Driven Documents and explains how to deal with its various components and sub-components.

Audience

This tutorial is prepared for those professionals who are aspiring to make a career in programming language and real-time processing framework. This tutorial is intended to make the readers comfortable in getting started with PySpark along with its various modules and submodules.

Prerequisites

Before proceeding with the various concepts given in this tutorial, it is being assumed that the readers are already aware about what a programming language and a framework is. In addition to this, it will be very helpful, if the readers have a sound knowledge of Apache Spark, Apache Hadoop, Scala Programming Language, Hadoop Distributed File System (HDFS) and Python.

Copyright and Disclaimer

Table of Contents

1.PySpark – Introduction

In this chapter, we will get ourselves acquainted with what Apache Spark is and how was PySpark developed.

Spark – Overview

Apache Spark is a lightning fast real-time processing framework. It does in-memory computations to analyze data in real-time. It came into picture as **Apache Hadoop MapReduce** was performing batch processing only and lacked a real-time processing feature. Hence, Apache Spark was introduced as it can perform stream processing in real-time and can also take care of batch processing.

Apart from real-time and batch processing, Apache Spark supports interactive queries and iterative algorithms also. Apache Spark has its own cluster manager, where it can host its application. It leverages Apache Hadoop for both storage and processing. It uses **HDFS** (Hadoop Distributed File system) for storage and it can run Spark applications on **YARN** as well.

PySpark – Overview

Apache Spark is written in **Scala programming language**. To support Python with Spark, Apache Spark Community released a tool, PySpark. Using PySpark, you can work with **RDDs** in Python programming language also. It is because of a library called **Py4j** that they are able to achieve this.

PySpark offers **PySpark Shell** which links the Python API to the spark core and initializes the Spark context. Majority of data scientists and analytics experts today use Python because of its rich library set. Integrating Python with Spark is a boon to them.

2.PySpark – Environment Setup

In this chapter, we will understand the environment setup of PySpark.

Note: This is considering that you have Java and Scala installed on your computer.

Let us now download and set up PySpark with the following steps.

Step 1: Go to the official Apache Spark download page and download the latest version of Apache Spark available there. In this tutorial, we are using **spark-2.1.0-bin-hadoop2.7**.

Step 2: Now, extract the downloaded Spark tar file. By default, it will get downloaded in Downloads directory.

```
# tar -xvf Downloads/spark-2.1.0-bin-hadoop2.7.tgz
```

It will create a directory **spark-2.1.0-bin-hadoop2.7**. Before starting PySpark, you need to set the following environments to set the Spark path and the **Py4j path**.

```
export SPARK_HOME=/home/hadoop/spark-2.1.0-bin-hadoop2.7
export PATH=$PATH:/home/hadoop/spark-2.1.0-bin-hadoop2.7/bin


export PYTHONPATH=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.4-
src.zip:$PYTHONPATH
export PATH=$SPARK_HOME/python:$PATH
```

Or, to set the above environments globally, put them in the **.bashrc file**. Then run the following command for the environments to work.

```
# source .bashrc
```

Now that we have all the environments set, let us go to Spark directory and invoke PySpark shell by running the following command:

```
# ./bin/pyspark
```

This will start your PySpark shell.

```
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
```

```
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0
      /_/

Using Python version 2.7.12 (default, Nov 19 2016 06:48:10)
SparkSession available as 'spark'.
>>>
```

End of ebook preview

If you liked what you saw…

Buy it from our store @ **https://store.tutorialspoint.com**