Spark Streaming

Let’s learn something!

PIERIAN DATA

Python and Spark

- Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams.

Python and Spark

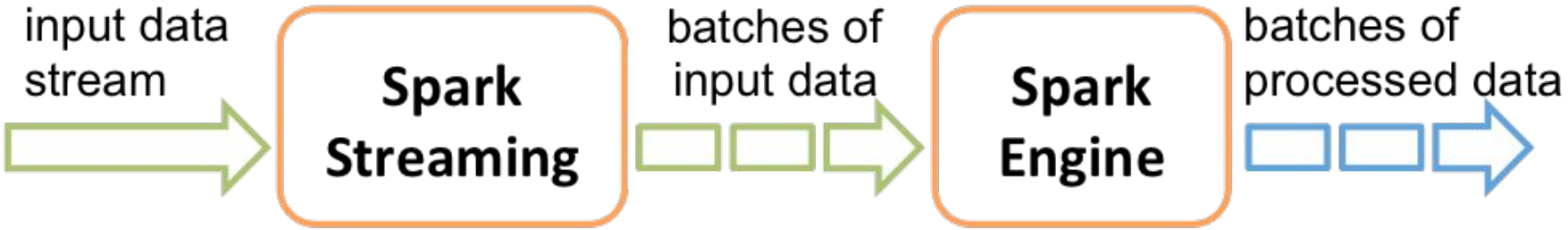
- Data can be ingested from many sources like Kafka, Flume, Kinesis, or TCP sockets, and can be processed using complex algorithms expressed with high-level functions like map, reduce, join and window.

Python and Spark

Python and Spark

- Internally, Spark Streaming receives live input data streams and divides the data into batches, which are then processed by the Spark engine to generate the final stream of results in batches.

Python and Spark



Python and Spark

- For this course we will first work through a simple streaming example.
- You will need to simultaneously use jupyter notebook and a terminal for this.
- This is easiest to follow through a local installation using Virtual Box.

Python and Spark

- After working through that example we will finish off the course with a Twitter Analysis Project.
- To follow along with this project you'll need to install a few visualization libraries and set up a Twitter Developer Account.

Python and Spark

- We'll walk you through all these steps when the time comes during the lectures.

Python and Spark

- The various possible data sources (Kafka, Flume,Kinesis, etc...) can not realistically be shown in a single computer setting.
- If your place of work necessitates use of one of these sources, Spark provides full integration guides.

Python and Spark

- Keep in mind not every source version is available with the Python API.
- Let’s jump to the documentation to show you where you can find additional information on Spark Streaming!

Spark Streaming Example Code Along

PIERIAN DATA

Python and Spark

- Because we will be using Spark Streaming and not structured streaming (still experimental and in Alpha) we need to use some older “RDD” syntax.
- This stems from using a SparkContext instead of a SparkSession.

Python and Spark

- We will be building a very simple application that connects to a local stream of data (an open terminal) through a socket connection.
- It will then count the words for each line that we type in.

Python and Spark

- The steps for streaming will be:
  - Create a SparkContext
  - Create a StreamingContext
  - Create a Socket Text Stream
  - Read in the lines as a “DStream”

Python and Spark

- The steps for working with the data:
  - Split the input line into a list of words
  - Map each word to a tuple: (word,1)
  - Then group (**reduce)** the tuples by the word (**key)** and sum up the second argument (the number one)

Python and Spark

- That will then provide us with a word count in the form **(‘hello’,3)** for each line.
- As a quick note, the RDD syntax relies heavily on lambda expressions, which are just quick anonymous functions.

Python and Spark

- Fortunately, all the lambda expressions used here are quite simple and basic.
- Let’s get started with this simple example!

Spark Streaming Twitter Project Part One

PIERIAN DATA

Python and Spark

- Now it is time for your final project!
- We'll create a simple application that plots out the popularity of tags associated with incoming tweets streamed live from Twitter.

Python and Spark

- We first need to create a Twitter Developer Account to get our access codes.
- Then you'll need to install the tweepy library as well as matplotlib and seaborn.
- Let's get started by going to:
  - **apps.twitter.com**

Spark Streaming Twitter Project Part Two

PIERIAN DATA

Python and Spark

- Our next task is to write a script that will connect to Twitter for streaming.
- This will be a .py file that we will call later on.
- Let’s get started!

Spark Streaming Twitter Project Part 3

PIERIAN DATA

Python and Spark

- Now we will finish off our project by setting up a Jupyter Notebook with Spark that will connect to the socket connection created by TweetRead.py!
- Remember to fill in your credentials for everything to work!

Python and Spark

- For this lecture I **highly** recommend that you avoid typing out the instructions on your own and instead use the provided notebook:
  - Introduction to Spark Streaming.ipynb

Python and Spark

- There are many places where a simple typo can mess everything up, and the nature of SQLContexts and the sockets will require you to restart your VirtualBox to get going again.

Python and Spark

- Depending on how many times you run through this exercise, the port you assigned in TweetRead.py may already be in use, we'll go over how to change that port.

Python and Spark

- We also need to install one more library called pandas.
  - **pip3 install pandas**
- Install this at the terminal and then restart you VirtualBox Ubuntu Machine before following along with this lecture.

Python and Spark

- Here are the steps we'll take:
  - Quickly review TweetRead.py
  - Open up the provided notebook file
  - Scroll down to the Twitter section
  - Run the .py files and cells in the correct order as we explain what is happening line by line.

Python and Spark

- There are many steps that must be done in the **exact** order for this entire process to work!
- If you do even a single step out of order, you'll need to restart the VirtualBox and start over again.

Python and Spark

- Using the provided TweetRead.py and Introduction to Spark Streaming.ipynb will save you a lot of time and frustration!
- Let’s get started