Week 2: Introduction to Spark and Resilient Distributed Datasets

Spark Core

- basic Spark functionality:
  - task scheduling, memory management, storage interaction, fault recovery
- defines Resilient Distributed Datasets (RDDs)

Relationship between Spark and Hadoop

- Spark does not require Hadoop, but can leverage its storage functionality (HDFS)
- can also leverage Hadoop's YARN to manage compute resources

SparkSession and SparkContext

- every Spark application has a "driver program" that's responsible for running operations on a cluster
- driver programs access Spark via a SparkSession object
- SparkContext is a component of a SparkSession and is what we're going to be using for this week

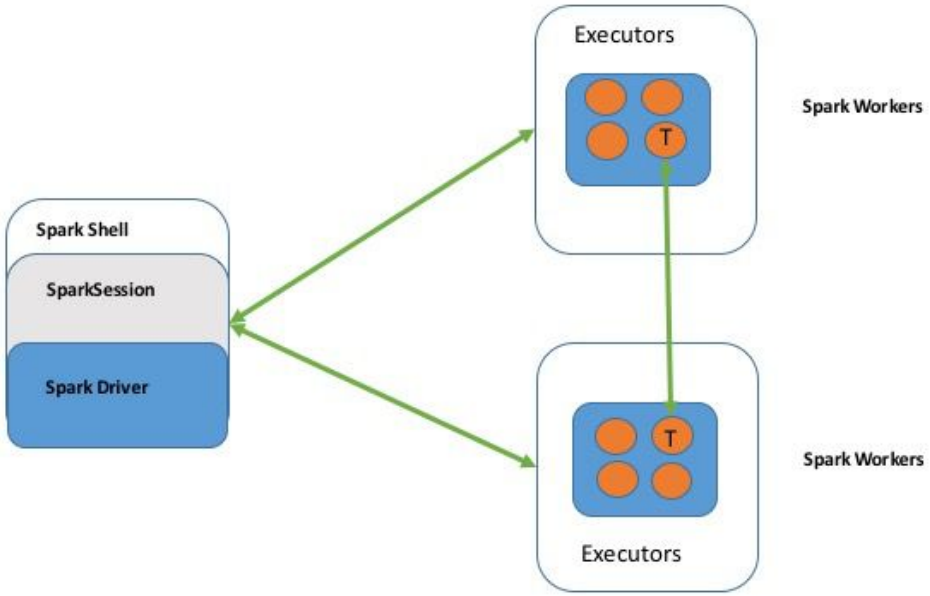
Getting set up:

- in our Spark environment, we will use "boilerplate" code to create our SparkSession
- SparkContext object is automatically created from our SparkSession
- for now, we will focus on SparkContext; we will return to SparkSession next week

Setting up a SparkContext

```
from pyspark.sql import SparkSession
spark = SparkSession \
    .builder \
    .master("local[*]") \
    .appName('My First Spark application') \
    .getOrCreate()

sc = spark.sparkContext
```


Executors
T
Spark Workers
Spark Shell
SparkSession
Spark Driver
T
Spark Workers
Executors