Laboratory 10: Spark Programming

Instructor: Young H. Cho T.A.: Arthur Win

Due Nov 9 at 11:59 PM (Report), Nov 11 at 11:59 PM (Video)

Apache Spark is a powerful distributed computing framework for efficiently processing very large datasets. You will explore its libraries using the PySpark API in both a standalone and distributed environment.

1. PySpark

We will first practice using Spark in local mode on a single-node machine. The following instructions only cover installation on an EC2 instance. If you are already familiar with using Docker containers, you may do this part on your own laptop or computer.

Create an EC2 instance using the Amazon Linux 2 AMI and select c5.large for your instance type. Make sure to modify your instance's security group inbound rules so that port 8888 is accessible by your IP address.

Run the following commands to install Docker onto your instance and logout of your SSH session.

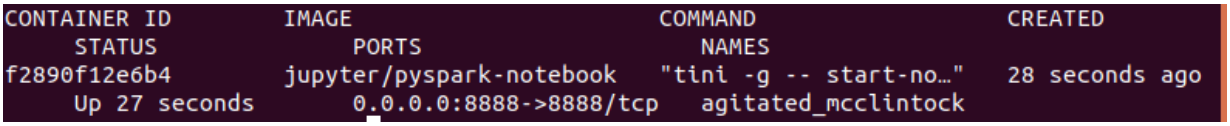
```
sudo yum update -y
sudo yum install -y docker
sudo service docker start
sudo usermod -a -G docker ec2-user
exit
```

SSH tunnel into your instance with the -L flag as shown below and run the subsequent commands to install the Docker image containing PySpark and Jupyter Notebook.

```
## Connect localhost port 8888 to instance port 8888
ssh -i myKey.pem ec2-user@XXXX.compute.amazonaws.com -L 8888:127.0.0.1:8888

## Install Docker image with PySpark Notebook
docker run -v ~/work:/home/jovyan/work -d -p 8888:8888 jupyter/pyspark-notebook

## Allow preserving Jupyter notebook
sudo chown 1000 ~/work

## Install tree to see our working directory next
sudo yum install -y tree
```

Run the following command to get the name of your container.

```
docker ps
```



In this case, my container's name is "agitated_mcclintock". Use the Docker logs to get the link to access your Jupyter Notebook running PySpark by running the following command.

```
docker logs "name of container with no quotes"
```

Copy the displayed link into your browser and create a new Python 3 notebook.

Follow the tutorial linked below starting from section "Spark Context" to get familiar with the PySpark API. The dataset that is used has been altered since the tutorial's writing, changing several features and breaking certain functions. Additionally, there are several typos that you will need to fix. When you're finished, download your notebook and make sure to **TERMINATE** your instance to avoid additional costs.

https://www.guru99.com/pyspark-tutorial.html

For a better fundamental understanding of how Spark and its API work, it is recommended that you create an edX account and watch all videos from Week 2, Lecture 2b of the BerkeleyX course "Scalable Machine Learning".

https://courses.edx.org/courses/BerkeleyX/CS190.1x/1T2015/course/

After completing the tutorial, download the "Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set" from the UCI Machine Learning Repository. This dataset contains accelerometer and gyroscope data from smartphones of people doing various activities. You can read more about the dataset and download it from the below link.

https://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions

Complete the following tasks on the dataset:

- Create a machine learning model using any algorithm not used in the tutorial from Spark's MLLib
- Report the training accuracy, test accuracy, and test F1-score for your model.

2. AWS Elastic MapReduce

When you run Spark in local mode, it will allocate tasks to local threads instead of worker nodes. Spark is a distributed computing framework and was not primarily designed to be used on a single-node machine. Here, we will practice running a Spark job on a real cluster using AWS Elastic MapReduce.

First, upload the smartphone data from the previous part into an S3 bucket.

Go to EMR in the AWS console and click on the "Notebook" tab. Create a new notebook with a two-node cluster of instance type c5.xlarge. EMR will create several EC2 instances of your selected type and automatically configure your cluster manager.

Wait for your cluster to start and your notebook to attach to the cluster. This can take anywhere from 5-10 minutes. Running a cluster in EMR is very expensive, so do not leave it idle and start using your notebook immediately when it becomes ready.

Open your notebook in JupyterLab, select your created notebook, and select PySpark for your kernel. Run the code from model you made in the previous part. In EMR, your SparkContext object is already created so you don't have to declare it. You can reference it using "sc" as usual. You will also have to reference files from S3 instead of a local file directory.

When you're finished, **TERMINATE** your cluster in the "Clusters" tab to avoid extra charges. You will certainly accumulate huge charges if this is not done properly. Your notebook is saved in an S3 bucket, which you can download and delete later if you choose.

Answer the following questions:

Some of your code may have run slower in a cluster than in local mode. Why is that? When does it make sense to use Spark for an application? Answer with respect to your instance type and dataset.

It is ever useful to use Spark in local mode? Why might you choose to use PySpark over some library like Pandas in combination with a different ML framework in local mode? When does it make sense to use one over the other?

Sources

https://www.guru99.com/pyspark-tutorial.html
https://courses.edx.org/courses/BerkeleyX/CS190.1x/1T2015/course/
https://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions