Apache Spark on EC2

(Hopefully DataBricks in Jul/Aug)

Data Science Indy

History

Spark was initially started by Matei Zaharia at UC Berkeley AMPLab in 2009, and open sourced in 2010 under a BSD license.

In 2013, the project was donated to the Apache Software Foundation and switched its license to Apache 2.0. In February 2014, Spark became an Apache Top-Level Project.

Arch / Component Overview

Spark Core and Resilient Distributed Datasets

- Memory biased distributed tasks, local resources, distributed datasets - cached in memory as much as possible

Spark SQL

- On top of Spark Core provides effecient SQL access to SchemaRDD for structured / semi-structured RDDs.

Spark Streaming

- Provides ability to apply batch analytics to streaming RDDs.

Components Overview (cont.)

MLlib Machine Learning Library

- summary statistics, correlations, stratified sampling, hypothesis testing, random data generation
- classification and regression: SVMs, logistic regression, linear regression, decision trees, naive Bayes
- collaborative filtering: alternating least squares (ALS)
- clustering: k-means
- dimensionality reduction: singular value decomposition (SVD), principal component analysis (PCA)
- feature extraction and transformation
- optimization primitives: stochastic gradient descent, limited-memory BFGS (L-BFGS)

GraphX - Distributed Graph Computation framework

Cluster on EC2 w/ Spark

Master + N Slave Machines

- Specify Number of Slave Machines
- Specify Machine Class
- Specify .pem file downloaded from ec2 console (MAKE SURE 0600 permissions before)
- Specify name of Key Pair managed in ec2

Set Access Keys and Download Spark

Set AWS keys

export AWS_ACCESS_KEY_ID=<access key>

export AWS_SECRET_ACCESS_KEY=<secret key>

Download Spark Distribution

wget http://apache.mirrors.hoobly.com/spark/spark-1.3.1/spark-1.3.1-bin-hadoop2.6.tgz

Launch Cluster on EC2

Launch Cluster

./spark-ec2

--key-pair=**instance_spark_demo_key_pair**
--identity-file=**/home/ec2-user/awsfiles/instance_spark_demo_key_pair.pem**
--ganglia
--region=us-east-1
--zone=us-east-1a
--instance-type=**m3.large**
--ebs-vol-size=**10**
--slaves=**3**
--copy-aws-credentials
launch **spark-demo-cluster**

Other spark-ec2 command actions:

launch, destroy, login, stop, start, get-master, reboot-slaves

Connect to Spark Cluster

Spark Control Panel:

http://<master node IP/DNS>:8080/

Ganglia Interface:

1st fix libphp module (libphp 5.6 not 5.5)
sudo vi /etc/httpd/conf/httpd.conf
sudo /etc/init.d/httpd start

http://<master node IP/DSN>/ganglia

IPython Notebook Setup / Config

Create pyspark profile on Spark Master Node:

ipython profile create pyspark

Make notebook support external access:

vi ~/.ipython/profile_pyspark/ipython_notebook_config.py

uncomment c.NotebookApp.ip and set to ‘*’ instead of localhost

c.NotebookApp.ip = '*'

Create ~/.ipython/profile_pyspark/startup/pyspark_setup.py

pyspark_setup.py content

```
import os
import sys

spark_home = os.environ.get('SPARK_HOME', None)

# check if it exists
if not spark_home:
    raise ValueError('SPARK_HOME environment variable is not set')

# check if it is a directory
if not os.path.isdir(spark_home):
    raise ValueError('SPARK_HOME environment variable is not a directory')

#check if we can find the python sub-directory
if not os.path.isdir(os.path.join(spark_home, 'python')):
    raise ValueError('SPARK_HOME directory does not contain python')

sys.path.insert(0, os.path.join(spark_home, 'python'))

#check if we can find the py4j zip file
if not os.path.exists(os.path.join(spark_home, 'python/lib/py4j-0.8.2.1-src.zip')):
    raise ValueError('Could not find the py4j library - \
        maybe your version number is different?(Looking for 0.8.2.1)')

sys.path.insert(0, os.path.join(spark_home, 'python/lib/py4j-0.8.2.1-src.zip'))

with open(os.path.join(spark_home, 'python/pyspark/shell.py')) as f:
    code = compile(f.read(), os.path.join(spark_home, 'python/pyspark/shell.py'), 'exec')
    exec(code)
```

IPython Notebook Setup / Config (cont.)

Run IPython Notebook using pyspark profile:

ipython notebook --profile=pyspark

Open in browser:

add port 8888 to master's Security Group

http://<cluster's master node IP/DNS>:8888