Apache Spark in CSB120 Installation Guide

Computer Science Department
Colorado State University

Download latest Apache Spark 3.0.3 binary (pre-build for Hadoop 2.7 and later): [Link] Go through its official documentation: [Link]

STEP 0: Your Hadoop cluster should be installed and running. To install Hadoop, please visit https://colostate.instructure.com/files/22375158/download?download_frd=1 and follow the instruction and video clips infospaces.

STEP 1: Downloading Spark

Log into your CS account. Download Spark release 3.0.3, Pre-built for Hadoop 2.7 using the following command.

```
wget https://mirrors.sonic.net/apache/spark/spark-3.0.3/spark-3.0.3-bin-hadoop2.7.tgz
```

Unzip downloaded archive to your home directory. Change directory (cd) to folder containing downloaded archive (~/spark-3.0.3-bin-hadoop2.7). Run the following command to unzip the archive and copy to your home directory.

```
tar -xvf  spark-3.0.3-bin-hadoop2.7.tgz -C ~
```

STEP 2: Environmental variables

Set SPARK_HOME environment variable to point above unzipped

Open .bashrc file, which will be inside your home directory. Add the following line, if it doesn't exist.

```
export SPARK_HOME=${HOME}/spark-3.0.3-bin-hadoop2.7
```

To reflect changes made in .bashrc file, run the following command.

```
source ~/.bashrc
```

Check the SPARK_HOME environment variable if it is correctly specified.

```
echo $SPARK_HOME
```

Note: Placeholders are denoted using <> in this document. You need to replace those (including <>) with appropriate values.

STEP 3: Configuration

Changes to configure Spark need to be done in the $SPARK_HOME/conf folder, which is inside the unzipped folder (spark-3.0.3-bin-hadoop2.7). Templates for every configuration file are already given in this folder. We just need to make changes in required templates and save them with correct names.

slaves

- This file stores the machines list/worker nodes (each machine name on a new line) your Spark environment uses to execute code.
- Save slaves.template as slaves.
- Remove localhost from the list and copy all the worker machines used in Hadoop (${HADOOP_CONF_DIR}/workers) and paste in slaves.
    - This will maximize the benefit of data locality as the same nodes are working as datanodes(HDFS) and workernodes(Spark).
- Save the slaves file.

spark-env.sh

- This file stores primary configuration options for the Spark environment.
- Spark provides several options for deployment, but we are configuring for Spark Standalone Deploy Mode. For more information, refer http://spark.apache.org/docs/latest
- Save spark-env.sh.template as spark-env.sh.
- Update following options (search for standalone) with appropriate values in spark-env.sh and don't forget to export them. You can refer to the sample below this table.

Environment variable	Description
SPARK_MASTER_IP	Bind the master to a specific hostname or IP address.
SPARK_MASTER_PORT	Start the master on a different port (default 7077). You need to select a set of available ports from the non-privileged port range. To reduce possible port conflicts, you will be assigned a port range.
SPARK_MASTER_WEBUI_PORT	Port for the master webUI (default: 8080). Follow port selection instructions mentioned in SPARK_MASTER_PORT.

SPARK_WORKER_INSTANCES	To set the number of worker processes per node (e.g. 2, 4)
SPARK_WORKER_CORES	Total number of cores to allow Spark applications to use per worker instance (e.g. 1, 2) (default: all available cores)
SPARK_WORKER_MEMORY	Total amount of memory to allow Spark applications to use per worker instance, e.g. 1000m , 2g (default: total memory minus 1GB)

Sample environment variables are given below. Put appropriate values in placeholders.

```
export SPARK_MASTER_IP=<hostname in CS120>
export SPARK_MASTER_PORT=<port number>
export  SPARK_MASTER_WEBUI_PORT=<port number>
export SPARK_WORKER_CORES=3
export SPARK_WORKER_MEMORY=1g
export SPARK_WORKER_INSTANCES=4
```

Save the spark_env.sh file.

Spark-defaults.conf

- This file is used to set default properties included when running spark-submit. This is useful for setting default environment settings.
- Each line in this file consists of a key and a value separated by whitespace.
- Save spark-defaults.conf.template as spark-defaults.conf.
- Create a directory named "spark_log" in the root of Hadoop dfs
- Update following options with appropriate values in spark-defaults.conf. You can refer to the sample below.

  Sample key-value pairs are given below. Put appropriate values in placeholders.

```
spark.master spark://<SPARK_MASTER_IP>:<SPARK_MASTER_PORT>
spark.eventLog.enabled true
spark.eventLog.dir hdfs://<Namenode_hostname>:<Namenode_port>/spark_log
spark.serializer org.apache.spark.serializer.KryoSerializer
spark.logConf true
spark.driver.memory 1g
spark.executor.extraJavaOptions  -XX:+PrintGCDetails -Dkey=value -Dnumbers="one two three"
spark.kryoserializer.buffer.max 128m
```

Save the spark-defaults.conf file.

STEP 4: Launching Spark Cluster

- To be safe, start a Hadoop cluster before launching Spark cluster.
- To start the cluster;
    - Login to Spark Masternode, specified in SPARK_MASTER_IP.
    - Run the following script to launch Master as well as slaves.

```
$SPARK_HOME/sbin/start-all.sh
```

- Check master WebUI using <SPARK_MASTER_IP>:<SPARK_MASTER_WEBUI_PORT> (Please read https://colostate.instructure.com/files/22375155/download?download_frd=1 to learn more about opening web user interfaces remotely)

- To stop the cluster;
    - Login to Spark Masternode, specified in SPARK_MASTER_IP.
    - Run the following script to launch Master as well as slaves.

```
$SPARK_HOME/sbin/stop-all.sh
```

- You can individually start and stop master as well as slave instances. Refer http://spark.apache.org/docs/latest/spark-standalone.html#cluster-launch-scripts for additional information.

STEP 5: Launching Spark Applications

- Spark applications can be launched using spark-submit script.
- Change directory to your project folder.
- Run the following command with appropriate values.

```
$SPARK_HOME/bin/spark-submit --class <your Class> --deploy-mode cluster --supervise <yourJar>
<any_arguments>
```

You can refer http://spark.apache.org/docs/latest/submitting-applications.html for more information.