Instructions for using Dumbo, NYU’s Hadoop Cluster

Summer 2017

1. Dumbo - Getting an Account, Finding Info

NYU's Hadoop Cluster, Dumbo

There is an NYU HPC Hadoop cluster (Dumbo) available for homework and projects - this is available to students registered for the course at no charge.

The NYU HPC IT team provides support for Dumbo - you can reach them at hpc@nyu.edu for assistance with the cluster; you can also use our class Forum on NYU Classes to get help.

To get an account, follow these instructions (you can select Suzanne McIntosh for sponsor):
https://wikis.nyu.edu/display/NYUHPC/Getting+or+renewing+an+HPC+account

You can read about Dumbo here:
https://wikis.nyu.edu/display/NYUHPC/Clusters+-+Dumbo

Once you have an account, instructions for logging in are here:
https://wikis.nyu.edu/display/NYUHPC/Clusters+-+Dumbo#Clusters-Dumbo-LOGGING_INLoggingIn

2. Dumbo - Compiling and Running MapReduce Programs

Instructions for compiling and running Java MapReduce programs on Dumbo

If you want to try Dumbo, here are steps I've used to compile and run on Dumbo. Use the Forum if you encounter any difficulties.

```
// Execute these two steps to log into Dumbo, remember to replace 'yourNetID' with your own net ID.
// Log into Dumbo - 2 steps
1. ssh -Y yourNetID@hpc.nyu.edu
2. ssh -Y yourNetID@dumbo.es.its.nyu.edu

// Write your driver source code using a text editor like vi (or emacs):
vi MaxTemperature.java
    // Note: The Dumbo cluster defaults to multiple reducers, normally it would be 1.
    //      You can assign the number of reducers in your driver code by adding this line:
    //      job.setNumReduceTasks(1); // 1 Reduce task
    //      If you add this line, your output result will be in HDFS in file part-r-00000.
    //      If you keep the default, your output will be scattered across 16 files from
    //      part-r-00000 through part-r-00015.

// Write your mapper and reducer source code:
vi MaxTemperatureMapper.java
vi MaxTemperatureReducer.java

// Compile your Java code:
java -version
yarn classpath
javac -classpath `yarn classpath` -d . MaxTemperatureMapper.java
javac -classpath `yarn classpath` -d . MaxTemperatureReducer.java
javac -classpath `yarn classpath`:. -d . MaxTemperature.java
    // Note: It's important to use the correct quotes in the commands above.
    //         If the above did not work, try substituting `yarn classpath` with: "$(yarn classpath)"
    //         Notice that the quotes are different in the two options show in the preceding line.

// Create your jar file
jar -cvf maxTemp.jar *.class

// Create your input data file on the local file system
vi temperatureInputs.txt

// Put your input data file into HDFS
hdfs dfs -ls /
hdfs dfs -ls /user
hdfs dfs -ls /user/yourNetID
hdfs dfs -mkdir /user/yourNetID/class1
hdfs dfs -put temperatureInputs.txt /user/yourNetID/class1
hdfs dfs -cat /user/yourNetID/class1/temperatureInputs.txt

// Run your MapReduce program
// Example: hadoop jar jarfile.jar className pathToInput/myInput.txt pathToOutputDir
hadoop jar maxTemp.jar MaxTemperature /user/yourNetID/class1/temperatureInputs.txt /user/yourNetID/class1/output

// Verify that the program ran and the results are correct
hdfs dfs -ls /user/yourNetID/class1/output
hdfs dfs -cat /user/yourNetID/class1/output/part-r-00000
```