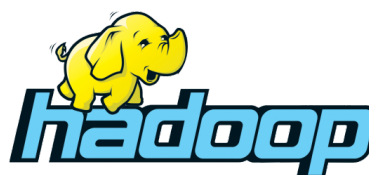


MapReduce on the Rutgers Hadoop Cluster



Introduction

At Rutgers, we have a Hadoop cluster that students can use to run Spark and MapReduce programs. In this guide, we'll be going over how to use Rutgers' Hadoop Cluster to run MapReduce Programs. In order to run MapReduce you will have to do three things: 1.) Place your runnable jar and input files onto the cluster, 2.) Move your input files into HDFS, and 3.) Invoke MapReduce.

Contents

1	Connecting to the Cluster	1
2	Moving Files to the Cluster	2
3	HDFS and Preparing Input	2
4	Executing MapReduce	2
5	Examining MapReduce Output	3

1 Connecting to the Cluster

There are three nodes at which we can connect to the Rutgers Hadoop Cluster: *data1.cs.rutgers.edu*, *data2.cs.rutgers.edu*, and *data3.cs.rutgers.edu*. You can connect to them via Secure Shell (SSH)

```
$ ssh <netid>@<remote>
```

example:

```
$ ssh abc123@data1.cs.rutgers.edu
```



You need to know linux commands: To interact with the cluster you will have to familiarize yourself with linux commands and network protocols such as ssh and scp. If you are unfamiliar with typical linux, check out <https://www.cs.rutgers.edu/resources/beginners-info> and look at the resources under the "Introduction to CS Resources" and "Intro to Linux" sections to read up on how to interact with the linux iLab machines.

2 Moving Files to the Cluster

In order to run MapReduce on the cluster, we need to have our runnable jars and input files on the iLab machines. You can transfer data to the iLab machines using the Secure Copy Protocol. Typical scp command usage is as follows:

```
$ scp <source> <destination>
```

Example:

```
$ scp file.txt abc123@data1.cs.rutgers.edu:~
```

Note: Notice the usage of the colon within the destination. The path after the colon specifies the file path the file should be placed on the destination system. In our example we simply used the tilde character, ~, which represents the home directory.

We can also transfer files from the iLab machines back to our local computer, switching the source and destination like such,

```
$ scp abc123@data1.cs.rutgers.edu:~/file.txt file.txt
```

3 HDFS and Preparing Input

Now for Hadoop to use your input, the input files must be on HDFS. By using scp you were able to get your files onto the machine, but not onto HDFS. Now you need to get it from the machine to HDFS. To get files on HDFS, use HDFS commands to move files to and from HDFS while SSH'd in the cluster.

You can move a file to hdfs by using the -put command:

```
$ hdfs dfs -put <source> <dest>
```

You can list your files on hdfs by using the -ls command:

```
$ hdfs dfs -ls
```

Example placing a file "input.txt" onto HDFS:

```
$ ls
input.txt
$ hdfs dfs -put input.txt
$ hdfs dfs -ls
Found 1 item
-rw-r--r-- 1 hadoop hadoop 0 2018-11-20 00:17 input.txt
```

You can read more on HDFS commands here:

<https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-common/FileSystemShell.html>

4 Executing MapReduce

When we want to interact with Hadoop, we run Hadoop commands. To run the MapReduce program, while SSH'd into the cluster, invoke Hadoop to run MapReduce by using the Hadoop jar command:

```
$ hadoop jar <jar location> <Main Class> <input> <output>
```

Example with the runnable jar on the cluster, the input data named input.txt in HDFS, and specifying Hadoop to create a folder named "output" to put the output files in:

```
$ hadoop jar WordCount-0.0.1-SNAPSHOT.jar WordCount input.txt output
```



output folder must be unique: When specifying the output folder, make the output folder should be unique. If a folder with the same name that already exists, the program will terminate and throw an error.

5 Examining MapReduce Output

After the MapReduce job executes, Hadoop should have created a folder on HDFS with name you specified when executing.

You can checkout the output by using the `hdfs ls` and `cat` commands. For example if you specified Hadoop to store the results in a folder called "output" on HDFS, you can list the files within the output folder using the `ls` command

```
$ hdfs dfs -ls output
-rw-r--r--    1 hadoop  hadoop           0 2018-11-20 00:06 output4/_SUCCESS
-rw-r--r--    1 hadoop  hadoop    49831 2018-11-20 00:06 output/part-r-00000
-rw-r--r--    1 hadoop  hadoop    49924 2018-11-20 00:06 output/part-r-00001
```

You can then print out the contents of one of the result files by using the `cat` command

```
$ hdfs dfs -cat output/part-r-00000
```

If you want to copy the results from HDFS back to your main directory within the cluster, you can use the `hdfs get` command:

```
$ hdfs dfs -get output/part-r-00000
```

If you want to transfer the results back to your local computer (not the cluster machine), you can use the `scp` command as described in section 2.

Useful links and Resources

Rutgers Hadoop Cluster

- *Hadoop Cluster Information*
<https://services.cs.rutgers.edu/hadoop.html>

Hadoop Cluster Interaction

- *HDFS Overview and Commands*
<https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-common/FileSystemShell.html>

MapReduce Guides

- *Apache MapReduce Tutorial*
<https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>

Last Updated: December 4, 2018