CPSC 4330 Big Data Analytics

In-Class Exercise 4 – Elastic MapReduce (EMR)

In this exercise, you will run a MapReduce application (the one that you created in exercise 3) on AWS using Elastic MapReduce (EMR).

Step 1: Develop a MapReduce Java Program
In exercise 3, you have developed and tested your application locally. Hopefully it's working well now. For many types of applications in the cloud, this is what you want to do – use local resources to develop and debug, and test as thoroughly as you can before scaling. This is often easier and keeps costs low since you're not paying for the cloud and just using resources that you already have.

Step 2: Open a web browser and go the website:
https://awsacademy.instructure.com/login/canvas. Log in using your AWS academy account and password.

Step 3: On the Dashboard, go the course "AWS Academy Learner Lab". Click "Modules" and follow the instructions in the file "AWS Academy Learner Lab Student Guide.pdf" to start the leaner lab and go to the AWS Management Console.

Step 4: Upload the JAR and input files to Amazon S3
To run your application on EMR, the first thing you need to do is to upload the application and input data to AWS Simple Storage Service (S3). The general S3 documentation can be found here: https://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS3.html.

How to upload data to S3? First, create a bucket for your application (See the "Create a Bucket" section of the document
https://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS3.html for instructions.) The bucket name is globally unique. For example, I named my bucket as "lil-lab4".

Now upload your avgwordlength.jar file into the S3 bucket.

Create a folder in the bucket named "input". Now go into the "input" folder and upload all your application's input data files here (i.e. comedies, histories, poems, tragedies under the shakespeare folder).

Step 5: Run an elastic MapReduce job
Go to the Elastic MapReduce (EMR) service in AWS. From the control panel there, click the "Create cluster" button, and then on the configuration screen,

- Under "Name", give a name to your cluster.
- Select Release "emr-7.6.0". Leave all the default services checked.

- You can keep the default setting for “Cluster configuration”. See https://docs.aws.amazon.com/prescriptive-guidance/latest/amazon-emr-hardware/capacity.html for the difference between Primary nodes, Core nodes, and Task nodes. The instance m5.xlarge is one of the Amazon EC2 instances (https://aws.amazon.com/ec2/pricing/on-demand/). For the subsequent exercises and assignments, m5.xlarge should be good enough for usage.
- You can keep the default setting for “Cluster scaling and provisioning” and for “Networking”.
- Under “Steps”, do the following things
    - Click “Add”, then click “Custom JAR”.
        - Give the jar a name
        - Set the “JAR location” to where you uploaded your .jar file in S3 (e.g. for my jar file, the location is s3://lil-ex4/avgwordlength.jar).
        - For the “Arguments” part, it needs two arguments. The first is the folder containing the inputs, and the second is the folder where you want the outputs to be saved (**The folder for the outputs should NOT exist – EMR will create it for you**). One argument in each line. For example, for my job, it should be
          s3://lil-ex4/input
          s3://lil-ex4/output
    - “Action on failure” can be left as the default.
- You can keep the default setting for “Cluster termination and node replacement”.
- Under “Amazon EMR service role”, select the service role “EMR_DefaultRole”.
- Under “EC2 instance profile for Amazon EMR”, select the instance profile “EMR_EC2_DefaultRole”.
- Keep all other defaults and click “Create cluster”.

Your cluster will take a little while to start and your job will also take a little while to run. Once it’s done, if it is successful, you can go back to your S3 bucket and see the results. You can download the files to look at them.
You may notice that there are three files: part-r-00000, part-r-00001, part-r-00002 in the output. This means that three reducers are used for this application. This is because for a m5.xlarge instance, the default setting is that the maximum number of reducers is 3. EMR tries to maximize the parallelism by using more reducers. If you only want one reducer for this application, you can add a line of code “job.setNumReduceTasks(1)” in the driver code.

Note: If your cluster cannot be started due to an error “The requested instance type m5.xlarge is not supported in the requested availability zone. Learn more at https://docs.aws.amazon.com/console/elasticmapreduce/ERROR_noinstancetype”. You can try changing instance type (e.g. m3.xlarge).

Step 6:

When your mapreduce job finishes, remember to click “End Lab” in the Leaner Lab.