Notes for Lecture 10: Big Data III

Amazon Cloud Services and Hive

STA 250: Prof. Paul Baines

Written by Justin Wang

Part I: Amazon Cloud Services

Introduction: Large tech companies such as Google and Amazon have a massive number of machines and servers. So many, in fact, that many researchers these days choose to rent time on these machines for large computations, rather than use their own machines. This is beneficial to them as it saves them a lot of hassle in having to maintain machines, and costs less in the long run.

We were fortunate enough to receive an educational grant from Amazon to use time on their machines. The login details to use their servers will be given in an email by Thursday evening.

How to Set up A Cluster on Amazon:

1. Login at *console.aws.amazon.com/console/home*.
2. Click on **Elastic Map Reduce**.
3. Click **Create a New Job Flow** to open a new window. This will set up a cluster machine for you to do your Hadoop computations on.
4. **Job Name page**: Give a name to the Job Flow. Stay with the default for Hadoop and AMI version (don't change anything). For Job Type, select Hive.
5. **Nodes page**: Select **Interactive Session**. *Note*: Core Instances are like data nodes, and Instance Count specifies the number of nodes.
6: **Key page**: For **Amazon EC2 key pair**, you need to set up your key (instructions will be given in Homework 2). Then select your key from the drop down menu. You don't need to change anything else; just continue.
7. **Summary page:** Review the summary page, and then Create Job Flow.
8. You will be taken back to the list of jobs. Wait until the **Master Public DNS Name** appears. This might take a little while to appear. It doesn't refresh by itself, so you need to refresh the page to check if it finished.

Using the Amazon Cluster:

1. Go to your Gauss directory. Then type:

```
ssh -i [YourPrivateKey.pem] hadoop@[Your Master Public DNS Name]
```

2. You are now using Hadoop on the Amazon Cluster. Make a directory and then list the directory (should be empty for now):

```
hadoop fs -mkdir test
hadoop fs -ls
```

3. To transfer distributed data from Hadoop (Amazon) to your local directory (Gauss), type:

```
hadoop distcp [source directory] [destination directory]
```

4. To copy files from laptop to hadoop:

```
scp -i [filename] hadoop@[Your Master Public DNS Name]
```

Part II: Hive

Introduction: On Hive, we want to create a table representing our dataset. The fields of the table we create (which consist of primitive types) will be the columns of the table. To create this table, we will learn a few basic SQL commands.

Ways to Fill a Table: There are two ways to fill a table in Hive.

(1) Upload a file containing a table.
(2) Copy the contents from another table in Hive.

Note: You cannot directly input data into a table in Hive. Also, in order to be able to do (2), you have to do (1) first.

Note: A common file format used in Hadoop is the **Sequence File Format**. It is a compressed format containing keys and values.

Example with N Grams data:

1. Log in to Amazon Cluster (see Step 1 from **Using the Amazon Cluster** above).

2. Launch Hive with a simple one word command: *hive*.

3. To avoid great pain in your Hive session, type the following two commands:

```
set hive.base.inputformat=org.apache.hadoop.hive.ql.io.HiveInputFormat;
set mapred.min.split.size=134217728
```

4. To get the N grams data, call a SQL command of the following form:

```
CREATE TABLE [name of dataset]
{
      [data type] [column1 name]
      [date type] [column2 name]
                ....
      [data type] [columnN name]
}
```

6. To list the names of your columns, type:

```
DESCRIBE [name of dataset];
```

7. To get the number of rows (observations), type:

```
SELECT COUNT(*) FROM [name of dataset]
```

8. To show all tables you have created thus far:

```
SHOW TABLES;
```

9. Using CREATE TABLE once more, create a normalized table called *normalized*. It will start out as an empty table.

10. To copy the contents to the normalized table:

```
INSERT OVERWRITE TABLE normalized
SELECT
(list of column names from other table)
FROM
(name of the other table)
WHERE
(additional specifications)
```

Note: Step 10 will take a while to run. The Map and Reduce it runs are Hive's, so you dont need to worry about it.

Another note: Hive is slow! Even basic tasks will take a long time.