Extra Credit Project

Student ID: hgudigun

Github: https://github.com/gudigundla/HadoopGeoLocation

Description:

DBPedia[1] is a collection of structured content extracted from Wikipedia. DBPedia has a huge dump of geo locations extracted from various Wikipedia articles. This project aims to group all the articles together by their geo location. The project is built on top of an existing project [2], it takes the existing project, modifies it, deploys and executes on the FutureSystems cloud infrastructure.

Hadoop MapReduce Job:

A Mapper and a Reducer are used in this project. The mapper parses the given input, tokenizes the input and extracts 3 things namely- ArticleName, PointType & GeoPoint. If the PointType matches to the following URL, then it is a valid data point, then the mapper extracts the latitude & longitude values and creates a location key by rounding up the latitude & longitude values. The mapper emits the location key & the article name. The Reducer groups all the articles based on the location key and dumps them into a output file.
URL: "*http://www.georss.org/georss/point*"

Dataset: DBPedia[3]

The dataset is very huge (4GB), so a subset of it taken and modified in order to spend more time to work on FutureSystems cloud infrastructure.

The project is run on the FutureSystems infrastructure. The following are the steps in details to setup the environment on FutureSystems and to execute the project on a 3 datanode Hadoop cluster.

Environment Setup:

You will require futuresystems & chameleon accounts before proceeding.

- Login to the futuresystems using the your userID
  ```
  $ ssh hgudigun@india.futuresystems.org
  ```

- Source the respective bashrc file, for Chameleon we have used
  ```
  $ source ~/CH-817724-openrc.sh
  ```

- Load the Openstack module, python module
  ```
  $ module load openstack
  $ module load python
  ```

- Start the agent

  ```
  $ eval $(ssh-agent)
  $ ssh-add
  ```

- Create a python virtual environment

  ```
  $ virtualenv bdsChameleon
  ```

- Cd into *bdsChameleon directory*

  ```
  $ cd bdsChameleon
  ```

- Activate the *bdsChameleon* virtual environment

  ```
  $ source bin/activate
  ```

- Install ansible

  ```
  $ pip install --trusted-host pypi.python.org ansible
  ```

- Add your public key to github.com and clone the big-data-stack github repository

  ```
  $ git clone --recursive https://github.com/futuresystems/big-data-stack.git
  ```

- Cd into big-data-stack directory and install the requirements.txt using pip

  ```
  $ cd big-data-stack
  $ pip install -r requirements.txt
  ```

- Create 3 virtual instances of Ubuntu on Chameleon, .cluster.py is already configured for this, change the default image to CC-Ubuntu14.04, set create_floating_ip to True in .cluster.py and change the remote_user from Ubuntu to cc in ansible.cfg.

  ```
  $ vcl boot -p openstack -P $USER-
  ```

- You can check the created VMs status using your userID

  ```
  $ nova list | grep hgudigun
  ```

- To make sure if all the 3 VMs are started and active, ping them using ansible

  ```
  $ ansible all –m ping
  ```

- Install the base system using ansible

  ```
  $ ansible-playbook play-hadoop.yml
  ```

- The vcl boot generates the inventory.txt file with all the ip addresses of the VMs along with their roles. To make sure that hadoop is working, ssh into the frontendnodes using username hadoop. You will find hadoop installation related files at /opt.

  ```
  $ ssh hadoop@129.114.110.30
  $ hadoop version
  ```

- Make sure HADOOP_HOME is set properly

  ```
  $ echo $HADOOP_HOME
  /opt/hadoop
  ```

Copy the 4 files, which are submit in the github/ canvas to the big-data-stack directory. The names of the 4 files are:

1. HadoopGeoLocation.jar – The java archive of our Hadoop GeoLocation project
2. geolocation.input– The input dataset for our project
3. geo_analysis_script.sh – The analysis script file
4. geolocation.yml – The ansible playbook used for deployment of code & data

Deployment:

I used ansible to deploy the project source code (jar), input data & the analysis script file. Execute the provided ansible playbook namely geolocation.yml. It deploys the code, input data & the analysis script file on to the frontendnodes VM listed in the inventory.txt file.

```
$ ansible-playbook geolocation.yml -i inventory.txt
```

Analysis:

SSH into the frontendnodes using the IP address found in the inventory.txt, use hadoop as username. Make sure the files are properly deployed into the home directory. Run the analysis_script.sh script file to start the analysis.

```
$ ssh hadoop@129.114.110.30
$ ls –ltr ~
$ sh geo_analysis_script.sh
```

The script creates a new directory on HDFS called /input. Then it copies the input file into it and then finally starts the hadoop job. It will take few minutes to run the geolocation map reduce job. The comments will guide through the process. After the analysis script is you can find the output of the job in your current directory in the form a directory named *output.* Inside that output directory, you will find 2 files namely *part-r-00000* and *_SUCCESS*. The *part-r-00000* file holds the results of the hadoop mapreduce job; you can find the articles grouped by geo-location.

```
$ cat ~/output/part-r-00000
```

References:

[1]http://wiki.dbpedia.org/
[2]https://code.google.com/archive/p/hadoop-map-reduce-examples/wikis/Wikipedia_GeoLocation.wiki
[3] http://downloads.dbpedia.org/3.3/en/pagelinks_en.csv.bz2