Havish Chennamraj
903201642
Porgramming Assignment 2 Readme

Building Dataset for Map Reduce Example and Reverse Web Link Graph

Dataset for Map Reduce Example

- I've chosen LiveJournal Social network with 4847571 Nodes & 68993773 Edges. You can download the dataset here
- My Map Reduce Example problem basically computes the outward edge with maximum weight for every node. For this, I'd need edge weights as well. 'parseDataset.py' adds a weight to all of the edges as a random number between 1 and 101 and writes it to graph.tsv in the following format:
  sourceID targetId EdgeWt
- To run 'parseDataset.py', make sure you have the source dataset as 'soc-LiveJournal1.txt' in the same directory as the script.

Dataset for Reverse Web Link Graph

- I've chosen to crawl CS 7646: Machine Learning for Trading, link here.
- 'scrape.py' does a BFS with the above link as the starting point. I've ran BFS for 200 iterations and managed to get a decent sized dataset of about 2700 nodes and 5500 edges.
- To run scrape.py, please follow the instructions below
  1.) Create a virtual environment by running 'virtualenv venv' in the util directory.
  2.) Load the Virtual environment by running 'source venv/bin/activate'
  3.) Install the necessary pip dependencies using 'pip install -r requirements.txt'
  4.) Run the scraper with 'python scrape.py'
- The scripts gives 'webLink.tsv' as the output file which follows the following format:
  sourcePage targetPage

Setting up the Development Environment

- We essentially need Java, maven and Hadoop for this project.
- Use this link below to download preconfigured virtual machine (VM) image (~3.0 GB). The virtual image comes with pre-installed Hadoop, Maven and Java.
- Download and install VirtualBox 6.x.x (https://www.virtualbox.org/wiki/Downloads)
- The VM is a 64-bit Ubuntu operating system and for it to work VirtualBox has a couple of requirements:
  1.) Your Host OS needs to be 64-bits.
  2.) You should enable VT-x or VT-d (depending on your computer, you may have either) setting in your BIOS or UEFI Firmware.
  3.) You should disable (in Windows) the Hyper-V platform in your Windows Feature list.
  4.) Refer http://www.fixedbyvonnie.com/2014/11/virtualbox-showing-32-bit-guest-versions-64-bit-host-os/#.WBmVV3eZMxG for more details.
- Install the VM by following the below instructions.
  1.) Open VirtualBox and click File >> Import Appliance…
  2.) Navigate to your downloaded hadoopVM.ova file and click Next.
  3.) Keep all the default settings and click Import.
  4.) VirtualBox will now install the VM. Once the Hadoop VM is installed, highlight it and click Start.
  5.) The username for the VM is cse6242. The password for the VM is cse6242. (Its basically a VM I've used for one of my past courses)

Loading Data into HDFS

- You can create the datasets from scratch by following the instructions in the 'Building Dataset' section or you can just download them from here.
- Move the codebases 'MapReduceExample' & 'WebLink' as well as the datasets into the VM.
- Navigate to the 'MapReduceExample' folder that contains the src directory, pom.xml, run.sh and an empty 'data' folder and load the data using '*hadoop fs -put path/to/graph.tsv /data'*
- Navigate to the 'WebLink' folder that contains the src directory, pom.xml, run.sh and an empty 'data' folder and load the data using '*hadoop fs -put path/to/webLink.tsv /data'*

Running Map Reduce Jobs for both Datasets

- The src directory in each of the above two directories contains a Java file with the Map Reduce Code in it.
- pom.xml contains the necessary dependencies and compile configurations for each Map Reduce Job. To compile, simply call Maven in the corresponding directory (where pom.xml exists) by running ‘mvn package’
- It will generate a single JAR file in the target directory (target/mapReduce-1.0.jar or target/mapReduceWebLink-1.0.jar).
- Now simply run ‘./run.sh’ in the corresponding directories to generate the output file in the same dir. The bash scripts does the following:
    - Run the JAR on Hadoop specifying the input file on HDFS (the first argument) and output directory on HDFS (the second argument)
    - Merge outputs from output directory and download to local file system.
    - Remove the output directory on HDFS.

Results

- To see how the output files produced by the Map-Reduce jobs look like click here.