NYU | Center for Data Science

Lab 2: Map-Reduce

Lab 2 outline

1. Login to HPC
2. Set up github account permissions
3. Run starter code (word counting) directly and via Hadoop
4. Translate a SQL query to map-reduce
5. Computing document similarity over a collection

NYU – HPC Clusters

- Login: ssh <YOUR NETID>@peel.hpc.nyu.edu
- HPC Wiki: https://sites.google.com/nyu.edu/nyu-hpc
- VPN (when outside NYU):

  https://www.nyu.edu/life/information-technology/getting-started/network-and-connectivity/vpn.html

HPC, modules, and mrjob

- HPC uses “modules” to add libraries and software to your environment

- The file “**shell_setup.sh**” included with Lab2 sets up everything you need

- If you want to develop and test on your own machine, you need **mrjob**:
    - **pip install mrjob**

      OR
    - **conda install mrjob**

Using HDFS

Uploading File:	**hadoop fs -put <file>**
List files:	**hadoop fs -ls**
Remove file (directory):	**hadoop fs -rm (-r) <file or directory>**
Retrieve file from HDFS:	**hadoop fs -get <file>** **hadoop fs -getmerge <file> <output-path>**

Note: HDFS is separate from the peel filesystem!

Running the word count demo

- Directly (for development/testing):
  - `cd Lab2/word_count/src/`
  - `python mr_wordcount.py ../book.txt`
- On the cluster:
  - `cd Lab2/`
  - `source shell_setup.sh`
  - `cd word_count/src/`
  - `bash run_mrjob.sh` ← Open this file in an editor to see how it works

- To get the results:

  ```
  hfs -get word_count
  hfs -getmerge word_count word_count_total.out
  ```

mrjob and Hadoop

- Read the word-count source carefully to see how mrjob works
- Read the shell scripts to see how to execute either locally or by Hadoop
- We provide the basic skeleton for the next parts, but you will need to write the mappers and reducers

First question: translating SQL

- You are given a dataset of movies and a SQL query to translate
- Edit **filter/src/mr_sql.py** to implement map and reduce
- Each call to the mapper will see one line of **movies.csv**
- You need to determine what the intermediate key/value structure is

Second question: document similarity

- mrjob allows you to write multi-stage pipelines

  map → reduce → map → reduce → ...

- Here you are given a collection of documents, and your job is to compute the bag-of-words similarity between each pair of documents

$$\textbf{Similarity}(A, B) = \textstyle\sum_{\text{words } w} \textbf{min}(\#w \text{ in } A, \#w \text{ in } B)$$

Tips

- Your program will produce an output file on HDFS
  - If the file already exists, your program will fail!
  - Get and remove the file between runs of your program
- Develop and test-run locally. MrJob makes this easy!
- Use the HPC's job status monitor to track your job progress
- Learn to parse the console output of mrjob and Hadoop!