CS118, Fall 2024, Project:
Analyzing the popularity of Wikipedia articles

Assigned: Dec. 02, 2024

Due: December 18, 2024, 11:59PM

For this project, you will develop a MapReduce application for analyzing the popularity of Wikipedia articles. Specifically, you will build an infrastructure that ranks English Wikipedia pages according to total page views and popularity trend.

You will develop your application as two sets of mappers and reducers. We recommend using Python 3. You will run your code on the June ^{1st} and June^{2nd} days of the dataset.

1 Collaboration & cheating policy

This project should be completed in your project/presentation groups. Do not share code or pseudo-code with other groups.

Cheating will not be tolerated. You are prohibited from using any solutions or code you find online, especially from the Github repository of Trending Topics website. Please check the official academic integrity resources for additional details.

2 Dataset

The Wikimedia dataset repository contains statistics about all articles published by Wikimedia, including Wikipedia, Wiktionary, and so on. Details about this repository can be found at `http://dumps.wikimedia.org/other/pagecounts-raw/`. We have uploaded days 1 to 5 of the June 2016 dataset to this box folder. *https://tufts.box.com/s/scu63p2ypjwba5iqcpig0pmc2s8bw84w*.

The dataset contains one file for each hour of the day-long period. Each line of each file contains four fields: `projectcode`, `pagename`, `pageviews`, and `bytes`. Many items in the `pagename` field are percent-encoded.

3 Your task: Filter, transform, & aggregate the input dataset

You should write a MapReduce job (a mapper and reducer, each as seperate python3 programs). Your MapReduce job should filter and transform the input dataset according to the specifications listed below. The reducer should then output article names (page names) along with the total number of views across the two day period.

Each line of the mapper's output should include a key and the number of page views as the value. Use a `\t` (Tab) character to seperate keys and values.

Each line of the reducers output should look like:

```
Modern_art\t50
```

Where `Modern_art` is the article title, `\t` is a seperator, and `50` is the number of page views.

You should be able to run your map/reduce job on the command line as:

```
gzcat 20160601/* | python3 mapper.py > map_20160601_output.txt
gzcat 20160602/* | python3 mapper.py > map_20160602_output.txt
cat map_20160601_output.txt map_20160602_output.txt | LC_C ALL=C sort -k 1 |
python3 reducer.py > reducer_output.txt
```

The first two commands mimic two mappers working on different parts of a large input dataset. The third command mimics a reducer that is aggregating the sorted/filtered data. The sort command sorts input rows by the key before passing it to the reducer. **Please note the `LC_ALL=C` prefix, which guarantees the same rules are used for sorting non alpha-numeric characters. Use it whenever you use the sort command.**

These commands may take some time to run. Note that your reducer should not keep every key in memory. It should only keep the current key and its data in memory.

Aside from Module 1 of our course: Note the use of pipes (`|`) to setup inter-process communication between the cat process, the sort process, and the reducer process. These pipes redirect the output of each process to a shared memory region. The memory is shared between the process outputing data and the process after the pipe operator. All of the processes are started (forked) at the same time. The latter ones wait on data being available in the shared memory regions before executing.

3.1 Specifications for filtering and transforming the input dataset

1. Transform `pagename` entries are percent-encoded. Change them to article names by replacing percent-encoded symbols with their single-sharacter equivalents. For example, this can be done using the `urllib.unquote_plus(<pagename>)` command from the `urllib` library in Python.

2. Exclude pages outside of English Wikipedia by filtering out any items whose `project` field does not begin with `en`. `Project` fields that begin with `en` should be included only if they have no suffix.

3. Exclude pages that do not need to be considered when finding trending topics. Exclude any pages whose title starts with `Media`, `Special`, `Talk`, `User`, `User_talk`, `Project`, `Project_talk`, `File`,`File_talk`, `MediaWiki`, `MediaWiki_talk`, `Template`, `Template_talk`, `Help`, `Help_talk`, `Category`, `Category_talk`, `Portal`, `Wikipedia`, or `Wikipedia_talk`.

4. Wikipedia policy states that all English articles must start with an uppercase character. So, filter out articles that start with lowercase English characters. You may notice that some articles have non-English titles. You should not filter out these articles.
5. Exclude any article that ends with an image or text-file extension (`jpg`, `gif`, `.png`, `.JPG`, `.GIF`, `.PNG`, `.ico`, and `.txt`).
6. Exclude boilerplate pages (`404_error`, `Main_Page`, `Hypertext_Transfer_Protocol`, `Favicon.ico`, and `Search`).

4 Starter code

There are some articles newline, tab, and other whitespace characters within the article name. To guarantee consistent handling of whitespaceYour mapper should start with the following code. Please use a \t character to seperate keys and values.

```
for line in sys.stdin:
  line = line.strip()
  # split the line into words
  words = line.split()
  if len(words) != 4:
      continue

  #words = [x.decode('utf-8') for x in words]
  if words[0] == 'en':
      words[1] = unquote_plus(words[1])
      ....
```

Your reducer should start with the following code. Please use a \t character to seperate keys and values.

```
from operator import itemgetter
import sys

current\_word = None
current\_count = 0
word = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    if (len(line.split('\t')) != 2):
        print (line)
        continue
    word, count = line.split('\t', 1)
```

5 Testing your code

During development, test your code during development using "toy" datasets made up of a few hours of the provided dataset.

We will provide results for the June 3rd and June 4th days of the dataset. You can verify your results against them. But, this does not guarantee your code will work for the June 1st-2nd datasets.

6 Deliverables & grading

There will be **40 points** for the correctness of source code. In addition, there are two graded deliverables for this project. All of your deliverables should be otained by running your project code on the 1st and 2nd days of the June 2016 dataset provided at *https://tufts.box.com/s/scu63p2ypjwba5iqcpig0pmc2s8bw84w*. When creating your deliverables, please adhere EXACTLY to the formats specified below. Plese hand in everything code via Gradescope.

6.1 Graded deliverables

We describe deliverables for the project below. Your output must match the specifications below **exactly**.

40 points Turn in your mapper and reducer code.

30 points Turn in a textfile with the first 10 lines from your mapper's sorted output. This is the output generated after the sort operator.Please be sure to adhere to the stated mapper output format.

30 points Turn in a textfile with the first 10 lines of your reducer's output. It should be called group_<number>_reducer_output.txt. Please be sure to adhere to the stated output format. Your output file should be sorted by article name. Break any ties by additionally sorting by article views so that the article with more views is ranked higher. Sorting by article name may require an additional sort operation after the reducer's output.

7 Resources

You may develop your code on your own laptop or on a cloudlab machine.

Cloudlab: If running on CloudLab, please choose a machine with at least 32GB of memory. For example, you may choose to use the *m410* or *m540* machines in CloudLab Utah. When starting your experiment, choose the default *small-lan* profile. When parameterizing your experiment, choose *advanced* and select 50GB as the temporary filesystem size. Please be sure to store your mapper and reducer otputs to this temporary filesystem, which will be located at `/mydata`.

Cloudlab experiments will expire after one day. You may choose to extend your experiment up to seven days, but you must do so before it expires. You can ask for extensions for more than seven days as well.

Acknowledgements

The steps for this project were obtained from a tutorial on how to create the Trending Topics website.