Projects

Jean-Pierre Lozi and Arash Vahdat

April 2, 2015

Instructions The last few weeks of CMPT732 will be dedicated to projects. You will work in pairs on a project of your choice. When you have decided on your subject, you are invited to tell us, during the labs, or by e-mail, what you wish to do. Once we have validated your subject, you can start working! The schedule will be the following:

- Your idea should be validated by one of us by March 30.
- You will present your project in a poster session on ~~April 9~~ Thursday April 16th, 11:30am-2:30pm.
- The code and final report of your project should be submitted before April 18, 23:59 PST.

There are no general rules about the projects, other that they should be using at large enough amount of data, they should be *scalable* (writing a single-threaded application that happens to run in a reasonable amount of time on your machine is *not* satisfactory, for instance), and they should be sufficiently challenging.

Report The report should briefly cover the following topics:

- Problem Definition: What is the problem that you are trying to solve? What are the challenges of this problem?
- Methodology: What is your methodology to attack the problem and the associated challenges? What is the computational and space complexity of your solution in terms of input size?
- Results and Discussion: What are the outcomes of the project?
- Guideline: Similar to the assignments, briefly explain which code was used for which task.

Note that your report should not exceed 5 pages. The rest of this document presents example project subjects to help you get started. While you can just pick a project from the list or get your inspiration from one of them, you are more than welcome to come up with your own ideas. Be creative!

1 Thread migrations: visualization of `perf` data

Using the Linux kernel's `perf` tool, you can record all thread migrations on a machine by using `perf sched record`. Doing so will produce large `.data` files, and you can then use `perf sched map` and `perf sched trace` to produce traces of the execution, in text mode, as explained here: `http://lwn.net/Articles/353295/`. Visualizing such data in text mode is often difficult, however, because of the very large number of thread

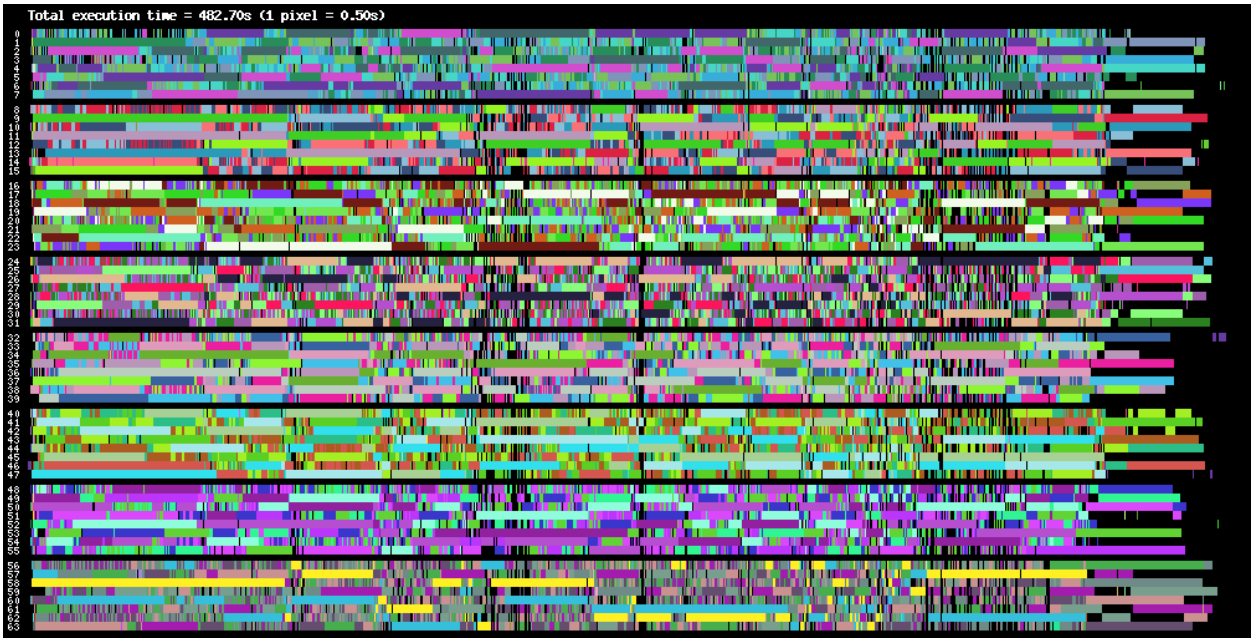

Figure 1: Plot of a run of TPC-H with Oracle on a 64-core machine

migrations that occur even during short executions. Make it possible, using a Hadoop program, for instance, to generate visualizations (images, animated GIFs, videos...), that make analyzing the data easier.

You will find, in `/cs/bigdata/datasets/oracle_run_queues_+_thread_placement`, `perf` traces obtained when running a TCP-H benchmark with Oracle on a 64-core machine. There are three `data.gz` files, one that correspond to a run where threads are free to move on all cores ("Free Threads"), one that corresponds to a run where threads can only move on their own NUMA node ("Pinned to Nodes"), and a run where threads are bound to their own node and cannot move around ("Pinned to Cores"). A basic low-resolution plot of the execution for Pinned to Nodes is shown in Figure 1: each color represents a different thread, the X-axis represents time, and the Y-axis represents the 64 cores of the machine.

There are way too many migrations for a low-resolution image like the one shown in Figure 1 to show all of them. At the following URL, you will find high-res plots of the execution that produce very large high-resolution images for every 15-second chunk of the execution (disregard the run queue plots):

`http://sfu.lozi.org/results/140913-oracle-run-queues-+-thread-placement-zoom`

For your project, you can, for instance, write a Hadoop program that reproduces theses images (both low-res and high-res). You can also make it possible to zoom into any part of the execution, at any resolution, ideally with a front-end that makes it possible to select which area to zoom in, graphically. If you have other ideas regarding how to visualize the data, go for it! Don't forget that if you need more traces than the ones provided, you can just use `perf sched record` to record them.

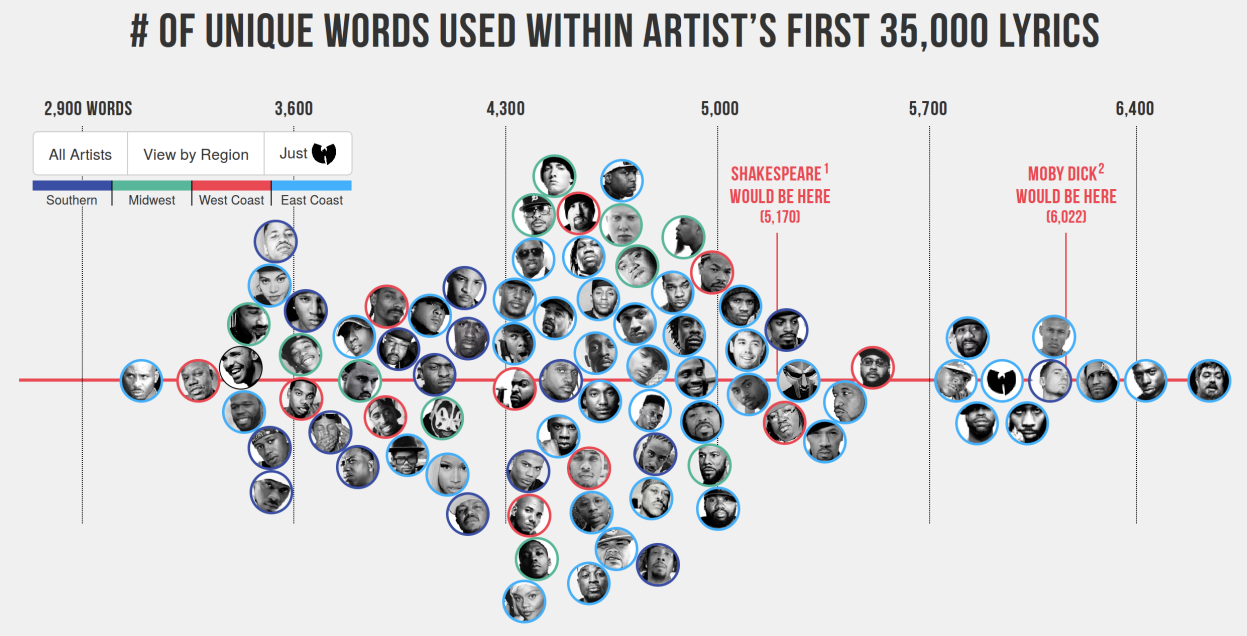

Figure 2: Vocabulary of various rappers: a project by Matt Daniels

2 Vocabulary size of singers and writers

Matt Daniels did a very interesting analysis of the vocabulary size of rappers. He used token analysis to determine each artist vocabulary, and produced various graphs, such as the one shown in Figure 2. The complete analysis can be found here:

http://rappers.mdaniels.com.s3-website-us-east-1.amazonaws.com/

Work on a similar analysis, either for a different musical genre, or for writers, for instance. You will first have to build a dataset, and to then write programs that extract various statistics from that dataset. Feel free to go beyond Matt Daniels' work and to produce graphs that show other interesting trends. If you decide to work with authors, you can, for instance, use data from Project Gutenberg (http://www.gutenberg.org).

3 Processing public tweets on Twitter

Twitter has an open API that allows you to aggregate public tweets. Twitter data comes with time and geo locations. It has a great potential to know when and where people are talking about what. Edward Chen has an interesting visualization shown in Figure 3 that illustrates where in United States people use terms 'pop', 'soda' or 'coke' to refer to soft drinks. You can find more information about this visualization and other project ideas from his blog:

http://blog.echen.me/2012/07/06/soda-vs-pop-with-twitter/

Kaggle has several tweet datasets available with interesting problems:

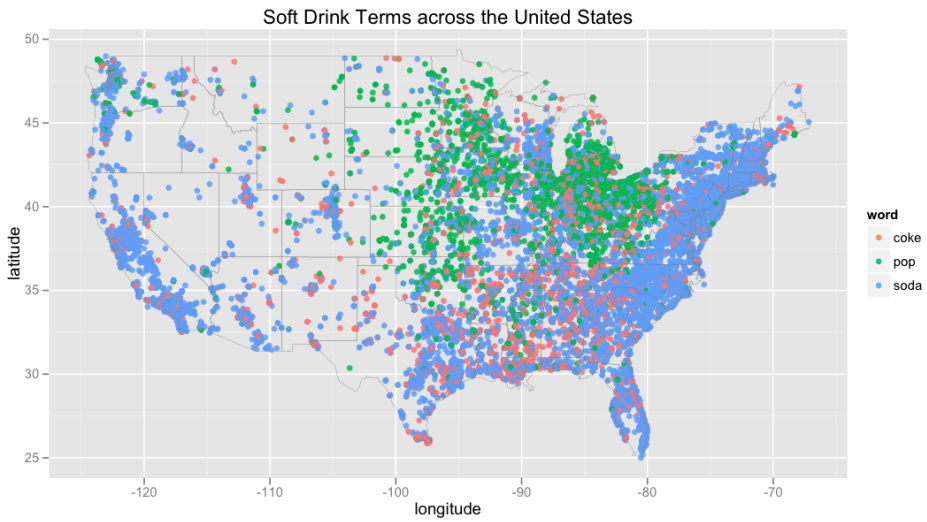

Figure 3: Soft drink terms across US by Edward Chen

https://www.kaggle.com/c/crowdflower-weather-twitter
https://www.kaggle.com/c/twitter-psychopathy-prediction
https://www.kaggle.com/c/twitter-personality-prediction

4 Processing large social networks

Large social networks are rich sources of information to analyze and extract patterns. Friend recommendation, attribute recommendation, finding central nodes or groups of similar people are examples of common challenges in social networks. A few years ago, Facebook hosted a challenge for the friend recommendation problem. The winner would be hired by Facebook:

http://www.kaggle.com/c/FacebookRecruiting/

A dataset containing Twitter graph is also available at:

http://an.kaist.ac.kr/traces/WWW2010.html

5 Projects from the Big Data for Social Good Challenge

If you need more project ideas, you can check IBM's Big Data for Social Good Challenge. Check the submission page to find Big Data projects that you can take inspiration from, or reimplement your own way:

http://ibmhadoop.challengepost.com/submissions

This page includes, for instance:

- A website with various graphs and statistics about 311 calls in New York City.
- A tool that shows the progression of influenza epidemics in the US and forecasts their evolution.

- A tool that makes it possible to correlate housing price in different areas of London with various demographic variables (crime in particular).

IBM provides many datasets to work with that you can use in your projects:

http://ibmhadoop.challengepost.com/details/data

6 Other resources

Many datasets can be found online, a good list can be found here, for instance:

http://www.quora.com/Where-can-I-find-large-datasets-open-to-the-public

You can also find inspiration in the Canada Open Data Experience hackathon, even though it was not focusing on Big Data in particular:

https://www.canadianopendataexperience.ca/

Don't hesitate to browse the web to find more datasets and ideas that the ones proposed in this document. Be creative, and don't forget to have fun!