Data Analysis in Parallel

Filip Ter - filip.ter@gmail.com

Motivation

- Many problems are conceptually trivial to parallelise (embarrassingly parallel)
  - Still many analysts, and data scientists avoid it due to difficulty of implementation
- IPyParallel allows for parallelisation with relatively minor changes to a typical Python data analysis workflow
  - No need to re-write whole codebase to use parallel computing
  - Even allows for the possibility of running on remote machines - removing limits on scalability
- With the size of datasets and average CPUs/computer increasing, this is becoming an increasingly relevant topic.
  - For some tasks it will become vital to run in parallel to complete it in a reasonable time

REPO LINK: https://bit.ly/2xCOYIq

Objectives

- To show how to use parallelism in the very basic form, splitting up independent data and preforming the same computation on these at the same time (SPMD parallelism).
- To show researchers that this can be done using their typical environment without needing to write complex code or changing their workflow excessively
- Main Goal: Run independent tasks sequentially, and then in parallel showing the performance benefit
- As further enhancements:
  - Analyse the performance improvement, identifying potential bottlenecks
  - To introduce the idea of debugging parallel processes
  - To introduce the idea of running this on remote machines - as opposed to just multiple cores.

REPO LINK: https://bit.ly/2xCOYIq

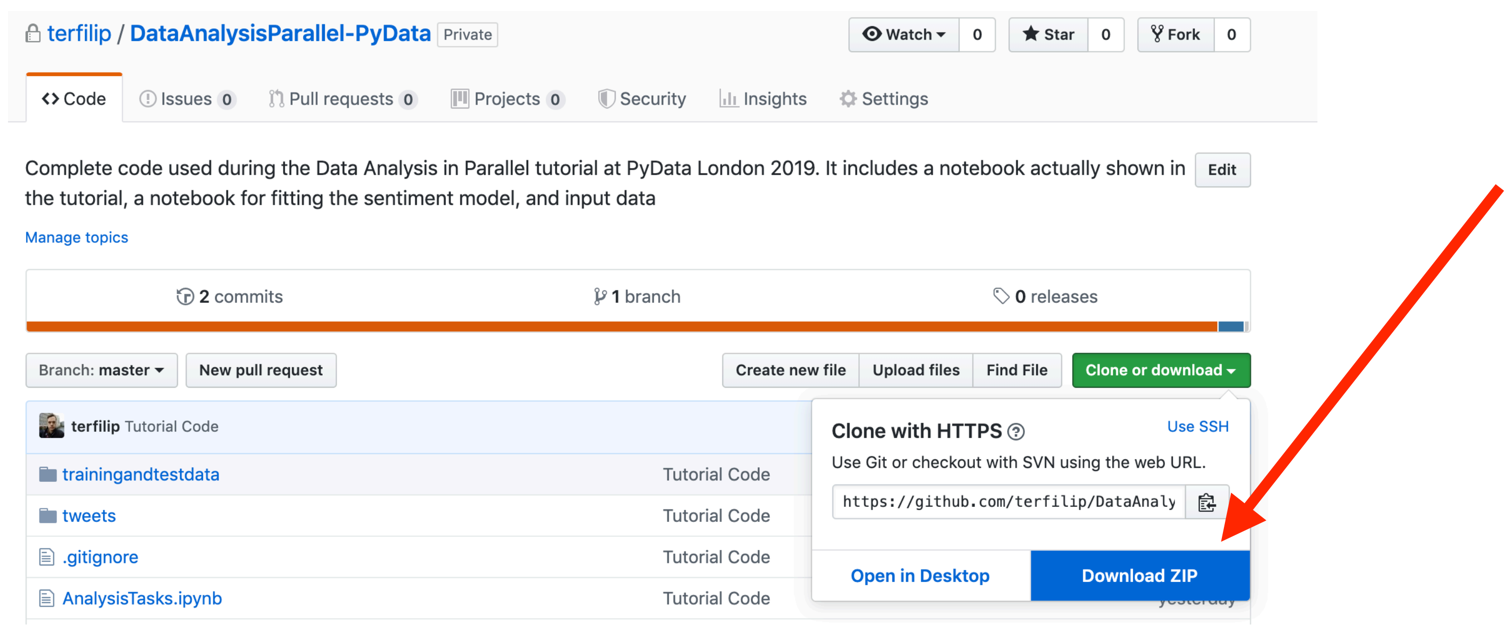
Structure of Tutorial

- An interactive Jupyter notebook will be used to show three analysis tasks. The code for these will be shown, and executed sequentially.
- Then I will run the same tasks in parallel, showing what the level of speedup is and how the usage of CPU cores changes
- For the second part I will explore the further points including performance analysis and debugging
  - If there is time, I will show what further setup would be needed to run this on remote machines
- Final 15 minutes for Q & A

REPO LINK: https://bit.ly/2xCOYIq

Setup

- Instructions for running my code during the tutorial:
- Install Anaconda if you don't already have it
- Code for the tutorial is in my Github repo (link in bottom right corner)
- Download the code from the repo as a zip file (or clone), unpack
- Follow the instructions in the README to get set up.
- Once you've done that you should be able to have a Jupyter notebook with the code open and be able to execute it.



Complete code used during the Data Analysis in Parallel tutorial at PyData London 2019. It includes a notebook actually shown in the tutorial, a notebook for fitting the sentiment model, and input data

terfilip Tutorial Code

trainingandtestdata	Tutorial Code
tweets	Tutorial Code
.gitignore	Tutorial Code
AnalysisTasks.ipynb	Tutorial Code

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/terfilip/DataAnaly
```

REPO LINK: https://bit.ly/2xCOYIq

Analysis Tasks

- NLP on tweets that mention 4 tech companies. For each 4 of these datasets the following will be performed
- Tweets were gathered from http://followthehashtag.com
- The tasks are as follows:
  1. Aggregate statistics on the tweets: count, avg. tweet length, date range of posting
  2. A histogram of the lemmatised tweets, this plot will show the frequency of the lemmas used in all the tweets for the particular company
  3. Classifying the sentiment of each tweet using a pre-fitted neural net model, and calculating the % of positive tweets
     - Code for fitting the sentiment classifier with Keras is provided in a separate notebook (FitSentModel.ipynb) in the Github repo for those who are interested, though it is not the subject of this tutorial.
- There will be both sequential and parallel implementations of these tasks

REPO LINK: https://bit.ly/2xCOYIq

Parallelisation - Main Topic

- Parallelism is the idea of running a computation at the same time on different processors and/or machines
- Not necessarily the same as concurrency where the execution may happen on a single processor but asynchronously
- Embarrassingly parallel problems are those that are can be implemented in parallel without any additional logic
    - Just split the data and run it on multiple cores
- I will show how to run the analysis form the previous slides on the 4 companies in parallel so that the total amount of time taken is reduced due to using multiple cores

REPO LINK: https://bit.ly/2xCOYIq

Further Parallelism Concepts

- Performance Analysis:
  - Explaining the level of speedup seen in the main demo, looking at bottlenecks and potentially counter-intuitive results
- Debugging:
  - Looking at how to debug the parallel processes, by writing to files, and raising exceptions; and discussing potential issues
- Running on Remote Machines:
  - There will not be a demo of this, but it will be discussed and there will be an introduction to some code for how it can be implemented if time allows

REPO LINK: https://bit.ly/2xCOYIq