Introduction

Learning Objectives

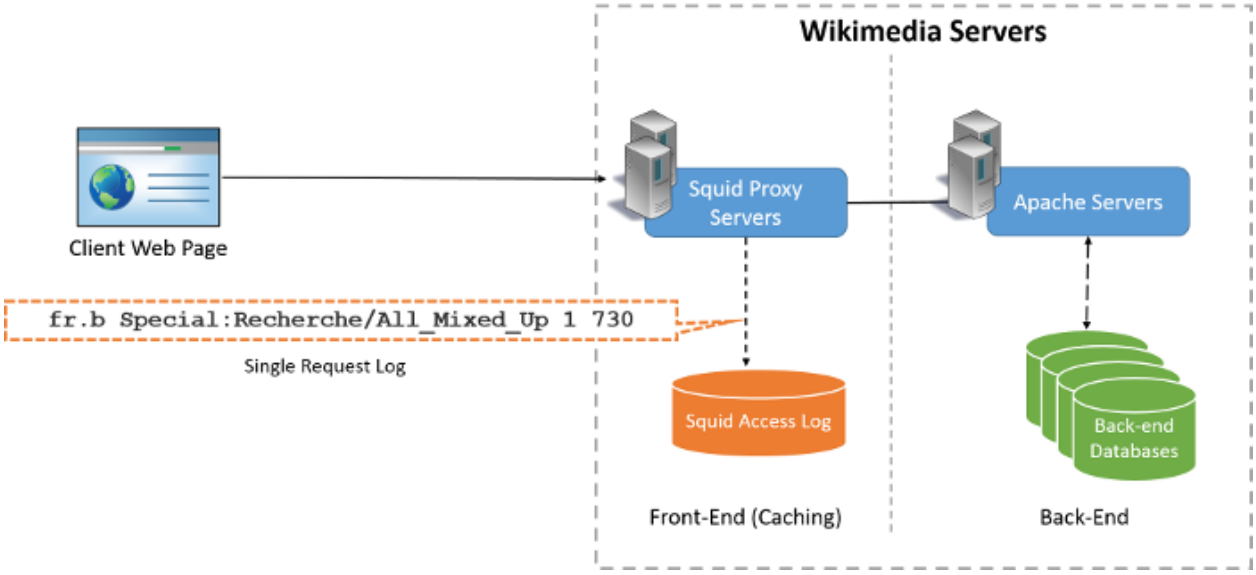
This project will encompass the following learning objectives:

1. Explore a large-scale dataset
2. Process part of a large-scale using sequential programs
3. Make interesting inferences from the processed dataset
4. Understand the limitations of sequential methods to analyse large volumes of data

In this first project, we will get a feel for big data by diving head first into analyzing a large dataset, specifically hourly page view statistics from Wikipedia.

Wikimedia maintains hourly page view statistics for all objects stored in Wikimedia servers as publicly accessible datasets. We will use these statistics to analyse page-view trends and derive the trending topics on wikipedia for a particular time range.

Exploring the Dataset



A simplified diagram of a page access from Wikimedia. More Information.

Every request made to Wikipedia's servers is serviced by a squid cache proxy, which also logs the request. These logs are shared publically every hour in flat text files. Each line of this file corresponds to a single access from the Wikimedia servers in the following format:

```
<project name> <page title> <number of accesses> <total data returned in bytes>
```

<project name> has two parts, a language identifier and a subproject suffix. The following abbreviations are used in the subproject suffix:

- (no suffix) : wikipedia
- .b : wikibooks
- .d : wiktionary
- .m : wikimedia
- .mw : wikipedia mobile
- .n : wikinews
- .q : wikiquote

- .s : wikisource
- .v : wikiversity
- .w : mediawiki

For Example, the following line:

```
fr.b Special:Recherche/All_Mixed_Up 1 730
```

denotes that from the French Wikibooks page, the page **Special:Recherche/All_Mixed_Up** was accessed once and 730 bytes were transferred in total. The article's title in this line is **Special:Recherche/All_Mixed_Up.**

In this project, you should pick any page view logs in the provided repository. You will use your scripts in the next project for running MapReduce job on Hadoop cluster.

The following video will walk through the process of writing a Streaming MapReduce job flow using Python and Java: https://www.youtube.com/watch?v=7n0XtbFEBBI

Filtering an Hour's worth of Data

Our aim is to identify trending topics from the English Wikipedia articles. In order to do this, develop a mapreduce task in any language to:

1. Filter out all pages that are not english wikipedia. (This means that the log lines should start with en (case sensitive), without any suffix attached).
2. There are many special pages in wikipedia that do not need to be considered when trying to find trending topics. Exclude any pages whose title starts with the following strings:
   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:
   Wikipedia_talk:
3. Wikipedia policy states that all English articles must start with an uppercase character. Filter out all articles that start with lowercase English characters. You may notice that some articles have non-english titles, you should choose to retain them in the analysis.

4. You may also get results which refer to image files, exclude any article that ends with the following extensions (Keep all other extensions intact). (.jpg, .gif, .png, .JPG, .GIF, .PNG, .txt, .ico). Do not use case-insensitive matching– remove exactly those file extensions.
5. Finally, there are some boilerplate articles which are returned by Mediawiki, which should be excluded as well. Articles with titles that exactly (case sensitive) match any of the following strings should be excluded:
   404_error/
   Main_Page
   Hypertext_Transfer_Protocol
   Search
6. Once the filtering is done, output the remaining articles in the following format:
   <page title>\t<number of accesses>
7. Submit your scripts to the Moodle. Good luck!

Notes

- You may notice that the data set consists of files beginning with languages en, En and EN. Apply the filter to only en(case sensitive).
- \t stands for the tab character.
- The output should be sorted in ascending the order of the number of accesses.