Cloud Computing

Assignment 4

You can collect up to 25 points for this assignment

MapReduce

- **Do start working on this assignment several days before the deadline!**

This is an individual assignment. *Discussing the assignment tasks and specific issues with other course participants is allowed and even encouraged. However, you should be the only author of all the solutions you provide in this assignment. Teamwork, pair programming, or copying solutions or program code from other persons is consider plagiarism and it will be handled following the Åbo Akademi University protocol for such cases.*

Instructions:

- Upload your assignment report as a **PDF** file along with your **source code files** and **program/EMR output files** on Moodle. Create a ZIP file for each subtask (Task 1.1, Task 1.2, Task 1.3, Task 2.1, and Task 2.2) and add the **source code and output files** in the ZIP files. Altogether, you should upload 1 PDF file and 5 ZIP files on Moodle.
- The PDF file should be named **Assignment4_LastName_Firstname_studentID.pdf**.
- **Each page** of your PDF file should have:
  - Your name
  - Student ID at Åbo Akademi University
  - A page number / total number of pages
- **Pay attention to the readability of your report!**

The goal of this assignment is to get familiar with the Hadoop framework provided by Amazon called the Amazon Elastic MapReduce (EMR).

Problem 1: Word Counting (15 points)

You will need to implement 3 word-counting applications using Amazon Elastic MapReduce:

Task 1.1 A MapReduce program counting from an input file the total number of words and providing as output the 100 most frequent words in decreasing order.

Task 1.2 An extension of the previous program implementing a combiner in the map function

- Question: how much performance is gained when using the combiner?

Task 1.3 A MapReduce program counting the number of words of length 3 and 5 (i.e., how many words having 3 and 5 characters does the input file contain).

You must write your code in Python (or Java) and submit your source code files along with your report on Moodle.

- The input of the programs is the **fiwiki-latest-pages-articles_preprocessed.txt** file containing some content of Finnish pages from Wikipedia. The file size is about 700MB, contains about 7 million lines and 700 million characters. You can download the input file from the following link: **Assignment 4 Datasets**
  NB: People in Åbo Akademi O365 with the link can view and download the above datasets.

- You should create an S3 bucket and store the input file in your bucket. You should then use the S3 link as an input to your programs.
- The outputs of your MapReduce applications should also be stored in a S3 bucket.

You can use an Amazon EMR cluster (AWS Management Console -> All Services -> Analytics -> EMR) of 3-6 instances. The number of instances will naturally impact the performance of the application. Feel free to test with clusters of different sizes (you can resize your cluster without the need to create a new one). Some instance types might not work with EMR. One of the instance types that should work is **m4.large**.

Include your **source code files** and **program/EMR output files** in your assignment submission.

In your report, explain the architecture of your code and its behavior. Comment on the obtained results and performance of your MapReduce application, and the impact of using a combiner in the map function.

Extra resources:

- https://riptutorial.com/hadoop/example/13413/word-count-program-in-java---python-
- https://web2.qatar.cmu.edu/~msakr/15319-f13/lectures/R3Demo.pdf
- https://dzone.com/articles/running-elastic-mapreduce-job
- http://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-common-programming-sample.html
- http://docs.aws.amazon.com/emr/latest/ReleaseGuide/CLI_CreateStreaming.html
- http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python
- http://snap.stanford.edu/class/cs341-2013/downloads/amazon-emr-tutorial.pdf

Problem 2: CDN billing (10 points)

The **apache_log_UTF8.txt** log file contains information about around 3.4 million requests to an Apache web server. You can download the log file from the following link:

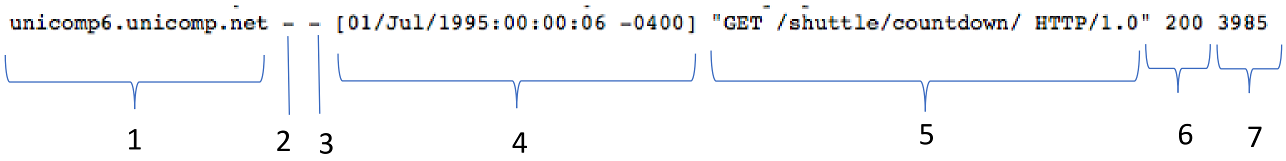
Assignment 4 Datasets
NB: People in Åbo Akademi O365 with the link can view and download the above datasets.

You should create an S3 bucket and store the input file in your bucket. You should then use the S3 link as an input to your programs.
The outputs of your MapReduce applications should also be stored in a S3 bucket.

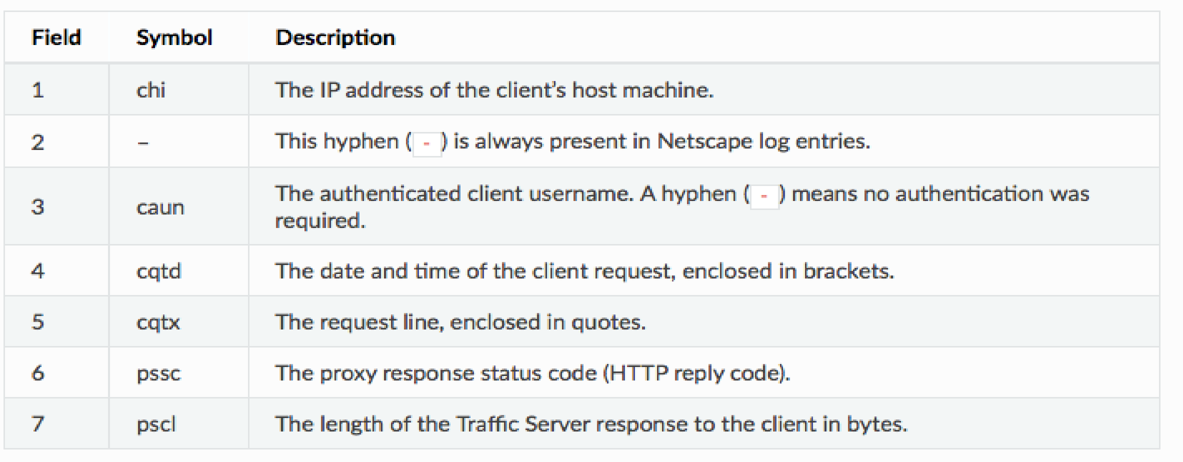
Assuming these pages were served through a CDN charging the following rates:

- 0,001 EUR per served request
- 0,08 EUR per GB of transferred data

The Apache server log data is formatted as the following:



```
unicomp6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985
```



Field	Symbol	Description
1	chi	The IP address of the client's host machine.
2	–	This hyphen (-) is always present in Netscape log entries.
3	caun	The authenticated client username. A hyphen (-) means no authentication was required.
4	cqtd	The date and time of the client request, enclosed in brackets.
5	cqtx	The request line, enclosed in quotes.
6	pssc	The proxy response status code (HTTP reply code).
7	pscl	The length of the Traffic Server response to the client in bytes.

Task 2.1 Implement a MapReduce program to calculate the resulting CDN costs due to the number of served requests and the transferred data. Provide the total number of requests and the total volume of transferred data (in base 2, 1024B=1kB, 1024kB=1MB, etc.).

Task 2.2 Implement a MapReduce program to provide the 5 most popular domain names (e.g., unicomp.net, letters.com or aa.net) from the client's machine names. Skip the requests having only IP address for the client's machine name.

Some instance types might not work with EMR. One of the instance types that should work is **m4.large**.

Include your **source code files** and **program/EMR output files** in your assignment submission.

In your report, explain the architecture of your code and its behavior. Comment on the obtained results and performance.

At the end of the report, you should also provide a **reflection** on what you learned during this exercise. This section could provide answers to the following questions:

- Have you learned anything completely new?
- Did anything surprise you?
- Did you find anything challenging? Why?
- Did you find anything satisfying? Why?

Shut down your EMR cluster when not using it!
Remember to terminate all VMs, terminate the load balancer, when you are done!
But keep your S3 bucket(s) used to store your results alive!