VE472 — Methods and tools for big data

Lab 2

Manuel — UM-JI (Summer 2022)

Goals of the lab

- Install Hadoop
- Setup a Hadoop cluster
- Run a simple test program

1 Lab organisation

This lab is expected to be completed within one week and two lab sessions. It means that your cluster and the simple text program must both be functioning by the second lab session.

The goal of the first lab session is to get started and benefit from the guidance of the teaching assistants when installing Hadoop. Ensure you have completed the download of Hadoop before the lab such that you can focus on its installation and setup during the lab time.

The second lab session will feature a short initial time where you can boot up your computer and ensure all the Hadoop services function properly. By the beginning of this second lab session all the tasks must have been fully completed, i.e. no work should be left. To ensure you have successfully completed all the exercises we expect you to demonstrate the running of the simple test program to the whole class and have an oral presentation on the main challenges you faced and how you overcame them

Groups can be freely formed, in the limit of four to five students.

2 Tasks

Ex. 1 — *Hadoop installation*

Download, install, and set up Hadoop 3.2.2.

Hints:

- Official documentation: `https://hadoop.apache.org/docs/r3.2.2/`
- Single node setup:
  - `https://data-flair.training/blogs/installation-of-hadoop-3-x-on-ubuntu/`
  - `https://tecadmin.net/setup-hadoop-single-node-cluster-on-centos-redhat/`
- Multi node setup: `https://www.tutorialspoint.com/hadoop/hadoop_multi_node_cluster.htm`
- Hadoop integration through regular Linux packaging systems: `https://wiki.debian.org/Hadoop`
- Ambari: `https://ambari.apache.org/`
- Use docker to facilitate Hadoop deployment ove rthe whole cluster
- Warning: not all those links are relevant to Haddop 3.2.x, but they should all contain insightful information on how to setup Hadoop

Ex. 2 — *Simple Hadoop streaming*

The goal is now to test the Hadoop installation. While this exercise can be completed in any programming language, make sure not to spend too long writing complex programs using a low-level language. A recommended choice allowing to complete all the tasks in a very short time and minimum number of lines, is bash together with `sed` or `awk`.

1. Write a short program that uses the lists of first-names and last-names to create a `csv` file where the first column contains a list of students, the second a ten digit random student ID, and the third one a random grade in the range 0 to 100. Each students should appear a random number of times all along the file with different grades.
2. Write a short program which extracts the grades from the previous file and for each line outputs on the standard output a pair of values constructed as follows: studentID<`TAB`>grade, e.g. 1234567890 34. Name this program `mapper`.
3. Write a short program which reads pairs from the standard input. Each tab-separated pair is composed of a studendID and a list of grades. Return the max grade for each student on the standard output. Name this program `reducer`.
4. Copy the `csv` file on HDFS and use Hadoop streaming to process it using the previous mapper and reducer programs.
5. Present some graphs and/or tables showing how the speed evolves as the size of the file increases. Compare the results for a single computer and a Hadoop cluster.