Cours Advanced Database – MSE HES-SO

Laboratory 2 – Diving deeper with Neo4j

This laboratory exercise aims at showing databases from a performance and operation point of view. It is again based on Neo4j, and will build on the knowledge acquired during the first laboratory.

For this laboratory, it is allowed to create teams of two persons. But you're allowed to work on your own as well.

Goal

Essentially, this laboratory consists of loading in neo4j a relatively big graph: the dblp article data. DBLP was an original effort led by University Trier in Germany to create a map of scientific contributions, and this way before scholar.google.com emerged as the leading tool to track papers and citations.

Your goal is to exploit this data to create in Neo4j a graph with more than 5 millions of nodes.

But this is not all. We also ask you to achieve this goal

1. without scaling-up too much the environment that loads the data…
2. In a decent time…

Dataset

The DBLP dataset is openly available here: https://www.aminer.org/citation. We will consider the latest dataset (v14) which is available for download here: https://originalfileserver.aminer.cn/misc/dblp_v14.tar.gz **but do not download it for now.** Mind that the compressed file is 2.5 GB big, and once uncompressed, the resulting JSON file is nearly 18GB big. This is a big file to deal with.

At the beginning, you can realise tests with this small test dataset:
http://vmrum.isc.heia-fr.ch/test.json

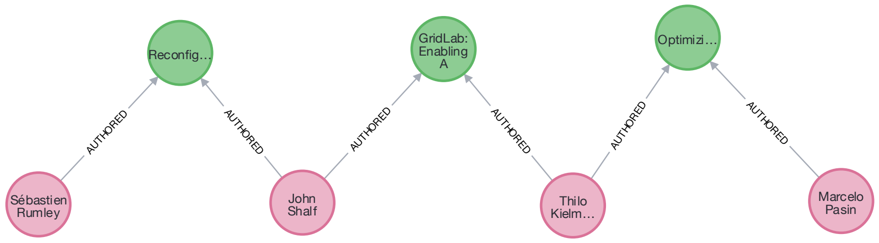
You can test scaling with this file:
http://vmrum.isc.heia-fr.ch/biggertest.json

The v13 is available (uncompressed) here:
http://vmrum.isc.heia-fr.ch/dblpv13.json

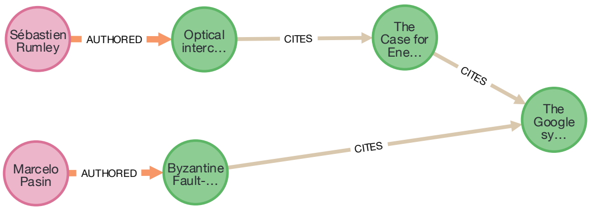
The v14 will be soon available uncompressed here:
http://vmrum.isc.heia-fr.ch/dblpv14.json

Graph structure

At the end we would like something like that:



We want to be able to find thru which co-authors one can relate an author to another.

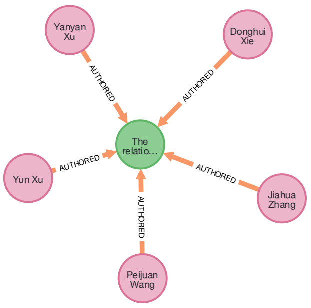


We also want to be able to navigate thru citations. Marcelo and Sébastien have apparently never cited the same paper, but by relaxing the constraint to citations of citations, one can find a "root citation".

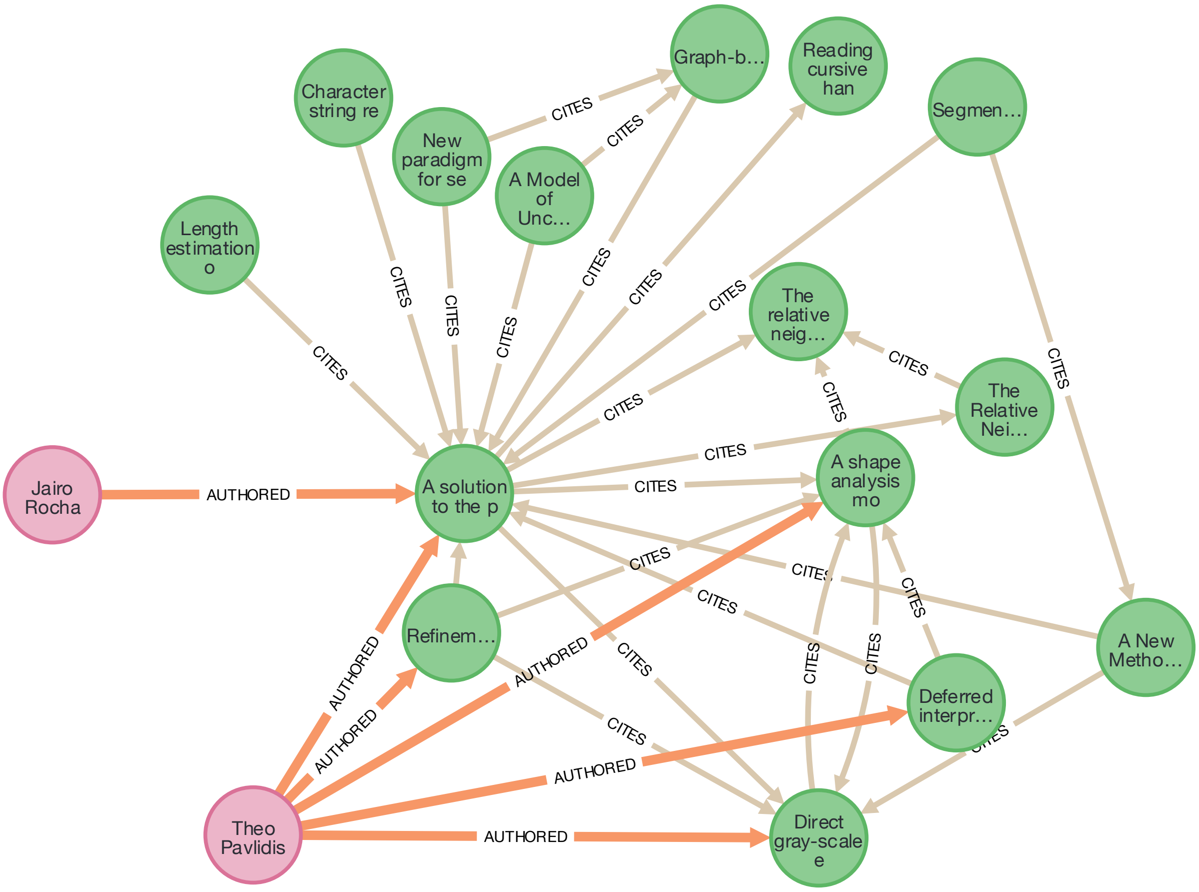
So we will build the graph as follows:

- ARTICLE nodes
- AUTHOR nodes
- CITES relationships
- AUTHORED relationships
- Each node (AUTHOR and ARTICLE) with have a key "_id" that corresponds to a key in the JSON file. For instance the _id of the first article is *"_id" : "53e99784b7602d9701f3e3f5"*.
- ARTICLE nodes will also have a title property. For the first instance of article in the JSON, the title is "*title" : "3GIO."*
- AUTHOR nodes will have a name property. For the second instance of article in the JSON (the first one has no authors), the first author is *"name" : "Peijuan Wang"*
- AUTHOR node will be connected to the ARTICLE nodes with the AUTHORED relationship.
- ARTICLE nodes will be related to other ARTICLE nodes they cite with the CITE relationship.

Below the graph resulting from the second article in the file. This article has no references and is not cited.



Below the neighbors of the third article in the JSON. It has two authors, a few references (5) and has collected 8 citations.



Docker limitations

We want the RAM of the neo4j container to be restricted. Hence, we do not want you to take too much advantage of caching because environments (container, vm) with a lot of ram are expensive. Any code that needs more than 4GB will not lead to any bonus. The less you'll use, the higher your bonus.

This should be how you have to run your neo4j container with a memory limitation:

```
docker run --name advdaba_labo2 -p7474:7474 -p7687:7687
-v $HOME/neo4j/logs:/logs
-v $HOME/neo4j/data:/data
-v $HOME/neo4j/import:/var/lib/neo4j/import
--memory="3g" --env NEO4J_AUTH=neo4j/testtest neo4j:latest
```

Initial instructions (more will be provided later on during the semester)

You are totally free to choose any programming language to load the graph into the DB. But it should work within a docker container (and later, within k8s).

It is recommended to plan a command line parameter (or environment variable) that can be used to load up to N articles.

Deadline(s)

The final deadline for the assignment is **May 25, 23h59**.

By this date, each team will share a git repository with its code. Each team will also return one or more performance result as follows:

{"team"="<team_name>", "N"=XX, "RAM_MB"="3000", "seconds"="YY"}

How will we be sure that your code has realized the performance mark? Because we will look at the logs **on Kubernetes**. Yes, you will have to package your "experiment" as docker and run it on a K8s cluster. Deployments details will be shared on April 14th.

Intermediate deadlines

Please send an email to sebastien.rumley@hefr.ch by April 13th with subject "AdvDaba TP2", indicating who is part of the team. Based on these emails, accesses to a k8s environment will be created.

A Q&A session about this laboratory will be hold on April 14th. By this date, it is recommended to have a script packaged inside a docker image that loads the "biggertest.json" of the graph inside a docker container.

Good luck!