Task: Network of Websites

Solutions engineering

Solutions engineering in Memgraph is all about connecting and solving customer needs with graph database capabilities. We sometimes like to call ourselves Field engineers.

1. Goal

The goal is to build a command-line script in Python that will be able to get all website page links from starting website **START_URL** with a max depth of **DEPTH**. The network of website URLs should be stored in the Memgraph database. The script should also be able to find the shortest path from **START_URL** to **END_URL** from a scraped network of websites in the Memgraph database.

Definition

```
$ python main.py network START_URL [--depth | -d DEPTH]
$ python main.py path START_URL END_URL
```

Examples

```
$ python main.py network https://memgraph.com --depth 3

$ python main.py path https://memgraph.com
https://discourse.memgraph.com/t/memgraph-lab-1-1-0-is-now-available/26
Shortest Path: 2 clicks
0 - https://memgraph.com/
1 - https://discourse.memgraph.com/
2 - https://discourse.memgraph.com/t/memgraph-lab-1-1-0-is-now-available/26
```

Service:

- Default **DEPTH** should be 2
- Raise **WebsiteNotFoundError(URL)** if **START_URL** doesn't exist when scraping for network
- If **START_URL** or **END_URL** doesn't exist in Memgraph database when fetching shortest path, raise **WebsiteNotFoundError(URL)**
- Raise **ShortestPathNotFoundError** if there is no path between **START_URL** or **END_URL**
- If more than one shortest path exists, choose any of them for output
- Extract website links only from a static page — no need for evaluating Javascript on Single Page Applications (SPA) or handling Flash websites

- Feel free to combine and use any other technology, library, and external resource
- Structure the project and directories however you like, but there should be a script **`main.py`** in the root directory that the user interacts with
- Be gentle with website servers when scraping for website links :)

Memgraph:

- Download community version of the Memgraph database from Memgraph Download Hub
- Use Cypher query language to manage data from/to Memgraph database
- Check Graph Algorithm Concepts to help you out with the shortest path query
- Use any Python library that supports Bolt protocol for connecting your service/script to Memgraph (check Appendix for guidance)
- Use Memgraph Lab to run Cypher queries and visualize results from Memgraph or Memgraph Client to run Cypher queries in terminal

2. Requirements

Main requirements:

- Working E2E solution of scraping and finding the shortest path for a simple use case
- Baseline technology is Python (version 3.5 or above)
- Data is stored in Memgraph database
- Unit and integration tests

Bonus requirements:

- Installation of the required libraries should be painless, so having a single command to install and set up a project so that **`main.py`** works after it
- The project should be runnable with Docker — setup **`docker-compose.yml`** for the service and Memgraph database

3. Process

Once you receive a task, you can ask us anything about the task (or life in general). Structure and organize your questions wisely because you can send us a list of questions **only twice** during this process. Once you are done with the project, send us a link to the git repository or zip the project and send it as a .zip archive.

4. Review

The idea is to present a correct, simple, and complete solution for the given task. The solution will be evaluated upon various criteria such as correctness, software design, proper testing, maintainability, extensibility, documentation, performance.

5. Appendix

Any Python library that supports Bolt protocol can be used to connect to Memgraph. Memgraph's Python library, pymgclient, currently only works on Linux-based systems. Neo4j's Python library, neo4j-python-driver, which can be used to connect to Memgraph as well, lacks performance but works on all platforms.

In case you will be using pymgclient, because it is a wrapper around C library mgclient, make sure to install and build mgclient library:

```
apt-get install cmake

git clone https://github.com/memgraph/mgclient.git /mgclient
cd mgclient && \
   mkdir build && \
   cd build && \
   cmake .. && \
   make && \
   make install
```