Lessons Learned im Umgang mit Neo4J

Allgemein

- Der “Import”-Ordner des Netzwerks muss explizit freigegeben werden, um den Neo4J CSV-Import über das Jupyter-Notebook ausführen zu können
- Wenn man große Datenmengen in die Datenbank laden will, sollte man diese nicht in einzelnen Queries oder einer großen Query einladen, sondern als CSV
- Man sollte eine UNIQUE-Constraint auf die IDs der Knoten legen, um das Ausführen von Queries anhand dieser IDs zu beschleunigen. Dies ist insbesondere beim Erstellen der Relationships essentiell
- Neo4J ist teilweise langsam im Schreiben (vor allem von neuen Relationships), dafür aber schnell beim Lesen der Daten, insbesondere über die bestehenden Relationships

Clustering

- In der Dokumentation für das Docker-Compose-Setup mit Neo4j wird eine Neo4J-Konfigurationsdatei angelegt und über ein Volume auf die verschiedenen Container gemountet. Dieser Mount wird allerdings nicht beim Laden des Containers aufgerufen, sodass die Konfigurationsdatei nicht vom Neo4J-Server gefunden wird. Es ist auch nicht möglich, die Konfigurationsdatei direkt in den vorgesehenen Conf-Ordner von Neo4J zu mounten, da die Neo4J-Instanzen diesen bei jedem Neustart komplett löschen.
    - Lösung: Die Konfiguration wird nun nicht in einer Konfigurationsdatei an Neo4J übergeben, sondern im Docker-Compose-File selbst festgelegt
- Die Konfiguration benötigt einzelne IP-Adressen für die Container. Deshalb musste jedem Container über das Docker-Compose-File eine eigene statische IP-Adresse im Netzwerk zugewiesen werden. Dies war nicht in den Beispielen von Neo4J dokumentiert.
- Die Informationen von Neo4J beim Starten der Container sind sehr unzureichend. Man muss tief in die umfangreichen Log-Files innerhalb der Container gucken, um Hinweise auf das zugrundeliegende Problem zu erhalten.
- Fehlermeldungen sind oft nicht aussagekräftig. Beispielsweise gab es die Info “Knoten 1 konnte Knoten 2 nicht finden”. Es wurde dafür kein Grund oder Lösungsvorschlag angegeben. In unserem Fall war das Problem eine fehlende Konfiguration für die Einstellungen “clustering_raft__advertised__address” und “clustering_transaction__advertised__address”
- Ein Neo4J-Cluster benötigt mindestens 3 Knoten. Wir haben über einen langen Zeitraum erfolglos versucht, ein Cluster aus 2 Knoten zu starten, allerdings wurde intern immer auf einen dritten Knoten gewartet, sodass das Cluster nicht richtig gestartet ist. Auch dies wurde unserer Ansicht nach nicht ausreichend dokumentiert.

Was wir in einem zweiten Versuch anders machen würden

- Nicht auf Linux und Windows abwechselnd entwickeln
- Zuerst den Cluster aufsetzen bevor die Queries formuliert werden, da damit einige Änderungen einhergingen
- Die Konfigurationseinstellungen von Neo4J direkt ins Docker-Compose-File schreiben und nicht in ein separates Config-File
- Import direkt über CSV-Files ausführen
- Beim Erstellen von CSV Files nicht erst Strings bilden, sondern die Inhalte direkt in das CSV schreiben