hochschule mannheim

BDEA SS23

Aufgabe 3: NoSQL Szenario (Social Network)

Lessons Learned

Gruppe B

Maximilian Broszio | Marvin Karhan | Rafael Kosiel

Erfahrungen und Lessons Learned

Installation:

- Die Inbetriebnahme und Anbindung der Datenbank war im Großen und Ganzen unproblematisch.
- Mit Hilfe der Online-Dokumentation war das Aufsetzen von mehreren ArangoDB Containern unter Docker gut machbar (vgl. https://www.arangodb.com/docs/3.11/install-with-docker.html).
- Ein ArangoDB Cluster setzt sich dabei aus mindestens einem Agent, einem Coordinator und einem DB-Server zusammen. In unserem Fall kommen zwei DB-Server zum Einsatz.
- Viele Programmiersprachen werden mit offiziellen Treibern unterstützt, wir haben uns für Python entschieden.

Datenimport:

- Direkter Import der Daten nicht durchführbar
    - Speicherüberlauf (> 100 Mio. Likes-Relationen)
    - Stückelung der Daten notwendig
- RAM Überlauf auch trotz Stückelung der Daten
    - Ursachensuche: zeitaufwändig
    - Problem: die Rest API (die unser Python Client nutzt); diese scheitert irgendwann bei sehr großen Datenmengen wie in dieser Aufgabe und der RAM-Speicher läuft voll.
    - Lösung: Import der Daten über das arangoimport CLI Tool
- Import dauert lange (um die 30 Minuten)
- Die Vorverarbeitung der Daten und das Erstellen der likes.json zum Import per CLI erfordert rund 5,5 GB zusätzlichen freien Festplattenspeicher
- Limitierung des Fanout und der Likes notwendig, da sonst ~2h Dauer des Datenimports. Der Fan out alleine, limitiert auf 100 User, dauert auf einem performanten Rechner ~616 Sekunden.

Datenbank:

- ArangoDB Query Language (AQL) Syntax ist verständlich (vgl. https://www.arangodb.com/docs/stable/aql/data-queries.html)
- Web-UI von Arango sehr gut gemacht und einfach zu bedienen
- Python Bibliothek gut dokumentiert und einfach zu verwenden
- Wenig Online-Hilfe abseits der offiziellen Doku zu finden (Stackoverflow etc.)
- Durch Edge-Collections können Relationen einfach und direkt indexiert erstellt werden.
- Das Arango Graph Objekt hat in unseren Tests nicht viel in Sachen Performance geholfen, daher sind wir größtenteils bei Edge-Collections geblieben

Entwicklung:

- Hohe Entwicklungszeit
    - Lange Query Laufzeit
        - Führen zu timeouts des Arango HTTP-Clients
        - Lösung: Timeout erhöhen

- Lange Daten Ladezeit
- Lösung: Daten limitieren

Wir denken, dass die Wahl unserer Datenbank letztlich kein Fehler war, insofern würden wir beim nächsten Mal ähnlich vorgehen. Sicher können wir das nicht sagen, da wir nur diese Datenbank getestet haben. In Zukunft wissen wir jedoch, dass es einen großen Unterschied macht, ob man die Daten per API oder CLI in die Datenbank importiert.