Lessons Learned

Kenne deine Datenbanken

Der naive Ansatz, sich auf eine Datenbank zu stürzen weil man neugierig ist und in der Vorlesung erlerntes Wissen gerne einmal hands-on anwenden möchte ist, wenn auch sicherlich von gutem Willen strotzend, kein methodisch sauberes Vorgehen. Bei der Auswahl der Technologie würden sich die Teammitglieder in Zukunft auf folgendes Vorgehen verlassen:

- Anforderungen evaluieren
- Mit einem kleinen Datenset und (wenn auch evtl funktionseingeschränkten) offiziellen Docker-Images ein Proof-Of-Concept (POC) erarbeiten
- Auf Basis der Learnings des POC die Datenmodellierung beenden
- Die Container auf die spezifischen Anforderungen anpassen und somit den ersten Demo-Stand erreichen

Besonderheiten/Einschränkungen von Cassandra

Die Datenmodellierung in C* entspricht nicht den klassischen Konzepten der Datenmodellierung relationaler Datenbanken. Statt sich anhand eines ER-Diagramms die benötigten Entitäten und deren Beziehungen zu überlegen, verfolgt man hier den Ansatz, als erstes die Queries zu entwerfen. Dies hat letztendlich zur Folge, dass für so gut wie jede Abfrage eine eigene Tabelle entworfen wird, welche die Daten in einem für die Query optimalen Schema bereitstellt.

In C* werden gewisse Operationen, welche in relationalen Datenbanken als selbstversändlich angesehen werden, schlichtweg nicht unterstützt. Schwierige sind daher Operationen wie bspw. **GROUP BY** und **ORDER BY**, welche nur unter erheblichen Einschränkungen, im Vergleich zu relationalen Datenbanken, verwendet/umgesetzt werden können.

Reine Schreibgeschwindigkeit rechtfertigt nicht alles

Im Anbetracht der Zeitaufwände, die aufgrund der Wahl von C* als Datenbank erbracht werden mussten, um ein starres, sehr eingeschränktes Query-Schema auf den gegebenen Use-Case anzuwenden sowie der teils stark suboptimalen Lösungen die daraus resultierten ist abzuwägen, ob nicht in vielen Fällen eine flexiblere Datenbank in ihren Vorteilen überwiegt. Gerade im Bereich Datenanalyse sowie nested Queries ist Cassandra in reinform extrem eingeschränkt.

BigData Datenbanken brauchen Platz

Cassandra ist, wie andere Datenbanken im BigData Bereich recht ressourcen-hungrig. Die Allocation von 4GB ermöglicht es, einen Cassandra Cluster (ohne weitere, zeitintensive und experimentelle Anpassungen an der cassandra.yaml) stabile Container im Betrieb zu haben, welche von einem modernen Home-PC oder Laptop noch zu handhaben sind. Für einen Produktivbetrieb empfiehlt sich als Untergrenze 8GB.

Fremde Images - Fremde Bedürfnisse

Bei der verwendung von fremden Images, welche zudem noch unzureichend Überprüft wurden ist (wie bei jedem fremden Code) mit schlechten Überraschungen zu rechnen. Im Falle dieser Abgabe wurde das Team mit folgenden Problemen konfrontiert:

- jeffharwell/cassandra-lucene
    - Jeff Harwell scheint ein Entwickler mit starken Überzeugungen zu sein. Leider treiben diese Überzeugungen den Entwickler zum verfassen eines Scripts, das dynamisch die cassandra.yaml überschreibt. Dieses Vorgehen ist sowohl unsauber (nicht dokumentiert) als auch unfassbar frustrierend für das Teammitglied, welches mit dem mysteriösen "nicht übernehmen" der Settings beschäftigt ist.
    - Duplizierte cassandra.yaml (/opt/jmx-exporter/etc/cassandra.yml und /etc/cassandra/cassandra.yaml) führen auf alle fälle zu verwirrung - vor allem wenn man zuerst die falsche YAML-Datei aufspürt.
- From Scratch
    - Das von Grundauf installieren von Cassandra auf einem Dockercontainer ist recht mühsam - vor allem wenn die meisten Guides auf veraltete und inzwischen archivierte Inhalte zeigen.
    - Das Aufspüren der jar-files eines vor 4 Jahre eingestellten Plugins kann Kopfschmerzen bereiten. Unter Umständen hätte das Team bei Aufmerksamer betrachtung des Repositories früher bemerken können, dass das Projekt inzwischen eingestellt ist und hätte sich dann nach anderen Technologien umgesehen.
    - Selbst nach erfolgreicher installation von Cassandra und dem Lucene-Plugin sowie der erzeugung eines stabilen Docker-Containers kann es bei der Bildung eines Clusters anscheinend wieder kritisch werden - an diesem Punkt hat das Team sich für das neue Base-Image entschieden (Zeitmangel, Frust)