

CA3 - Abschlusspräsentation

...

Gruppe A - Jannis Seefeld, Luca Gauß, Salvatore Scionti

Gliederung

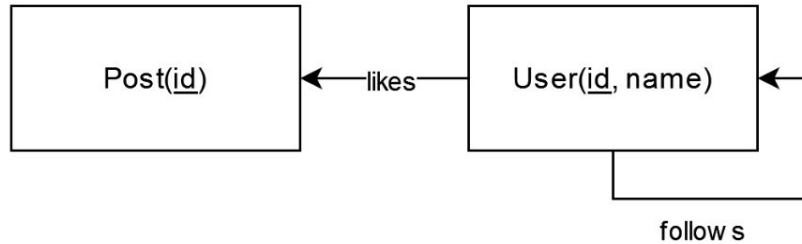
1. Datenmodell
2. Queries
3. Fanout

Datenmodell

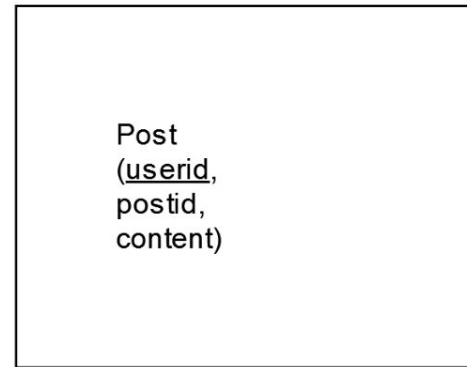
- Cassandra und Neo4j
- Cassandra als Cluster, Neo4j Cluster kostenpflichtig
- Datenbanken als Docker-Container
- Cluster automatisiert mit docker-compose

Datenmodell - Übersicht

neo4j



cassandra



postid from date_time
as timeuuid

Datenmodell - Vorverarbeitung

- Einlesen twitter_combined.txt
 - Einzigartige User auflisten, mappen mit Anzahl der Follower
 - User IDs mit Liste zufälliger Namen kombinieren -> Speichern als CSV
-
- Einlesen tweets.csv
 - Zeilen reduzieren auf userid, postid und content -> CSV
 - userid zufällig aus 100 Usern mit meisten Followern
 - 1/1000 der Likes eines Posts zufällig auf User verteilt -> CSV

Datenmodell - Einlesen in DB

- Kopieren der generierten CSV-Dateien mit docker copy
- Verbinden mit Cassandra/Neo4j-Client
- Einlesen der Daten
- Cassandra: COPY-Befehl
- Neo4j: LOAD CSV (mit PERIODIC COMMIT)
- Problem: COPY-Befehl ist CQLSH-Befehl, nicht CQL-Befehl
- Abhilfe: `docker exec cass_1 cqlsh -e "[...]"`

Queries

- Wie sehen die Datenbanken aus?

```
cqlsh:tweeter> describe posts_by_user  
  
CREATE TABLE tweeter.posts_by_user (  
  userid bigint,  
  postid timeuuid,  
  content text,  
  PRIMARY KEY (userid, postid, content)  
) WITH CLUSTERING ORDER BY (postid DESC, content ASC)
```

Node Labels

*(133,847)

Post

User

Relationship Types

*(2,354,741)

FOLLOWS

LIKES

POSTS

Queries - 1. Abfrage

- normaler select bei Cassandra

```
cqlsh:tweeter> select content from posts_by_user where userid = 34428380;
```

```
content
```

```
-----  
Last night I was stuck in 0 miles an hour traffic. Hungry, tired, I just want  
Just doin' what I love.. ☺ ☺  
try to meet everyone.. I love you all and I h  
I love this scene from #BackToTheFuture hahaha https://t.  
We  
Just tried to buy the remaining Holiday Spice inventory from the @starbucks next  
Go get my lil sis @BeaMiller's new song!! https://t.co/j46  
RIP my beautiful angel.. Forever will never  
You guys know I ☺ working out, but do you want to know my latest obsession? Find out in th  
Filmmaker @FinalCutKing's Halloween tricks result in hilariously delightful treats  
The magic of Mexican trio @kaayoficial and their first @LatinGRAMMYs nomination  
"I'm a staunch
```


Queries - 2. Abfrage

- MATCH in neo4j

```
match (u1:User)←[:FOLLOWS]-(u2:User) return u1,  
count(u2) as followers order by followers desc limit 100
```

"u1"	"followers"
{"name":"Dakota","id":"115485051"}	3383
{"name":"Muhammad","id":"40981798"}	3216
{"name":"Clayton","id":"43003845"}	2735
{"name":"Killian","id":"813286"}	2647

Queries - 3. Abfrage

- wieder reiner MATCH in neo4j

```
CALL [
  match (u1:User)←[:FOLLOWS]-(u2:User)
  return u1, count(u2) as followers
  order by followers desc limit 100
]
match (u3:User)-[:FOLLOWS]→(u1)
return u3, count(u1) as followingInTop100
order by followingInTop100 desc limit 100
```

"u3"	"followingInTop100"
{ "name": "Dashawn", "id": "3359851" }	69
{ "name": "Lyric", "id": "15846407" }	55
{ "name": "Natalia", "id": "14691709" }	50
{ "name": "Linda", "id": "7846" }	49

Queries - 4. Abfrage (1)

- “Mix und MATCH” zwischen neo4j und Cassandra
- Zuerst: Wie viele Personen folgen unserem gewählten User?

```
match (u1:User {id: '3359851'})←[:FOLLOWS]-(u2:User)  
return count(u2) as followers
```

"followers"
1905

Queries - 4. Abfrage (2)

- Dann: Wie vielen Personen folgt unser gewählter User?

```
match (u1:User {id: "3359851"})-[:FOLLOWS]→(u2:User)  
return count(u2) as following
```

"following"
1158

Queries - 4. Abfrage (3.1)

- Auflistung der 25 beliebtesten Posts
- Zuerst: Top Posts der Personen, denen ich folge

```
match (u1:User {id: "3359851"})-[:FOLLOWS]→(u2:User) match (u2)-[:POSTS]→(p:Post)
match l=()-[:LIKES]→(p) return u2, p.id, count(l) as likes order by likes desc limit 25
```

"u2"	"p.id"	"likes"
{"name":"Beckett","id":"63485337"}	"a6ce3a27-a439-11e6-8796-98b882b5c649"	430
{"name":"Ana","id":"48485771"}	"bf5ef0eb-fea8-11e6-b802-4ddf13858ab5"	378
{"name":"Kyan","id":"16303106"}	"2a511250-ef49-11ec-8179-ae27ddaa1fd0"	329
{"name":"Alijah","id":"18776017"}	"e09d41ed-e6e4-11e5-95a7-a5434e35511d"	317
{"name":"Conrad","id":"783214"}	"ae6028d4-c949-11e5-927f-22aa58c96bd2"	316

Queries - 4. Abfrage (3.1)

- Dann: Pro postid aus neo4j ein select in Cassandra in einer Schleife

```
('select content from posts_by_user where postid = ' + element + ' ALLOW FILTERING;');
```

- Grund: vorläufige Browser-Darstellung und Sortierung

```
["FOLLOWER COUNT:1905","FOLLOWING 1158 people","TOP POSTS OF FOLLOWED PEOPLE:","NAME:Beckett  
https://t.co/C160vg1Mcn,LIKES:378","NAME:Kyan,ID:16303106,POST:https://t.co/FL1tOHAZvN,LIKES  
https://t.co/nh8Xfc7IWU,LIKES:317","NAME:Conrad,ID:783214,POST:https://t.co/ZWMJvgSRpV #PURP  
https://t.co/SzLYRjhHfK,LIKES:242","NAME:Alfredo,ID:46423291,POST:Purpose https://t.co/e6wd8  
Tokyo,LIKES:230","NAME:Ana,ID:48485771,POST:#7YearsofMyWorld https://t.co/cSYIitfGj0,LIKES:2  
https://t.co/G4IHvAGrqX,LIKES:226","NAME:Annabel,ID:27855118,POST:Beliebers.. We did it! I l
```

Queries - 4. Abfrage (3.2)

- Auflistung der 25 neusten Posts
- erste zwei Abfragen gleich
- dieses Mal Sortierung nach date -> timeuuid in Cassandra
- Erstellung eines Strings, der alle postids (aus neo4j erhalten) enthält, dann eine IN-Abfrage in Cassandra (timeuuid sortiert automatisch)

```
var stringCass: string = '';  
for (const element of arrCass) {  
  stringCass += element + ',';  
}  
stringCass = stringCass.slice(0, stringCass.length-1);
```

```
e('select userid, postid, content from posts_by_user where postid in (' + stringCass + ') limit 25 Df
```

Queries - 4. Abfrage (3.2)

```
"FOLLOWER COUNT:1905","FOLLOWING 1158 people","NEWEST POSTS OF FOLLOWED PEOPLE",
https://t.co/7r3KrZ2WQILIKES: 16","NAME: DashawnID: 5442012POST: 🤔🤔🤔 https://t.co/7r3KrZ2WQI
https://t.co/CeiYkbN9EYLIKES: 42","NAME: DashawnID: 5442012POST: I ❤️ NY #CNY
DashawnID: 5442012POST: Here we goooo! @nickjonas & I are ready to take
pictures! ❤️ @devonnebydemi https://t.co/sqdPootqQkLIKES: 27","NAME: Dashawn
DashawnID: 5442012POST: Living in the moment with Beaver Suit https://t.co/a
```


Fanout

- User in neo4j bekommt zusätzliches Attribut “timeline”
- timeline enthält die Posts, die auf der Startseite angezeigt werden sollen
- neue Posts werden in die Timeline der Follower geschrieben
- bestehende Timeline -> neuer Post Bedingungen für Startseite, wird er eingefügt
- keine Timeline -> neue Timeline mit Post

Danke für die Aufmerksamkeit!