Starting out with Neo4j: Creating relationships between lots of data and server freeze ups

2 posts by 2 authors

Rob Hoelz Apr 10

Hi Neo4j users and developers,

I've been playing with Neo4j a bit lately; I'm using data from the GHTorrent project (http://ghtorrent.org) to answer some questions about the relationships between GitHub users and projects. The first thing I did was to import all users as nodes with the type "User", and the following relationship as nodes with type "Follower"- I figured I'd import the data before establishing the relationships between them. Followers have user_id and follower_id; I created indices on each of these, as well as on User.id. I tried to make a FOLLOWS relationship like so:

```
MATCH (f:Follower),(followee:User),(follower:User)
WHERE f.user_id  = followee.id
AND   f.follower_id = follower.id
CREATE (follower)-[:FOLLOWS]->(followee)
```

For a sense of the size of the data involved, there are about 3.7M Users in the dataset, and about 4.8M following relationships.

I ran the above statement, and after a while, I tried an EXPLAIN on it. EXPLAIN explained to me that it was taking the cartesian product of the two users "tables", and then whittling them down. The cartesian product was a gigantic number, and I'm doubting it would ever finish. Is there a better way to go about creating the relationship?

I then had the idea to not create a Follower type, but just create the relationship via the followers CSV load. To do this, I tried removing all follower nodes:

```
MATCH (n:Follower) DELETE n
```

This also took some time, until the browser page pointed at Neo4j reported that Neo4j was no longer responding. I was unable to connect via neo4j-shell as well (timeout), and the server didn't respond to SIGTERM or SIGINT, so I had to give it a SIGKILL. Before I killed it, I observed a lot of CPU load on part of the java process. Is there something I'm missing about why Neo4j would freeze up like this?

I'm really a novice at Neo4j, and I am probably improperly applying a lot of knowledge from the RDBMS world. Does anyone have any pointers or advice for what I'm doing?

I'm running Neo4j 2.2.0 on Arch Linux. Please let me know if you need more information from me.

Thanks,
Rob

Michael Hunger Apr 10

Hi Rob,

thanks for your question & feedback. Answers inline.

> Am 10.04.2015 um 16:21 schrieb Rob Hoelz <rdh...@gmail.com>:

Hi Neo4j users and developers,

I've been playing with Neo4j a bit lately; I'm using data from the GHTorrent project (http://ghtorrent.org) to answer some questions about the relationships between GitHub users and projects. The first thing I did was to import all users as nodes with the type "User", and the following relationship as nodes with type "Follower"- I figured I'd import the data before establishing the relationships between them. Followers have user_id and follower_id; I created indices on each of these, as well as on User.id. I tried to make a FOLLOWS relationship like so:

I'd probably take a different approach and used the follower data from the input to find the two users directly and connect them, either via a csv input and LOAD CSV or parameters to cypher statements running against the http endpoint.

Not sure you need the :Follower node, as this is a relationship in Neo4j, just putting in the join table from the relational approach does not make sense with a graph database.

Make sure to have indexes in place for quick lookups, you can drop the :Follower node and

```
create index on :User(id);
or
create constraint on (u:User) assert u.id is unique;

explain

        MATCH (f:Follower)

   MATCH (followee:User)

        WHERE f.user_id  = followee.id

MATCH (follower:User)

        WHERE   f.follower_id = follower.id

        CREATE (follower)-[:FOLLOWS]->(followee);
```

For a sense of the size of the data involved, there are about 3.7M Users in the dataset, and about 4.8M following relationships.

-> the query plan looks much better.

creating 4.8M following relationships in one single transaction will very probably exceed your available heap. So it would make sense to limit it, e.g. by only working with user_id's in certain ranges.

e.g. adding a WHERE followee.id > 500000 and followee.id < 75000
or similar for larger ranges.

> I ran the above statement, and after a while, I tried an EXPLAIN on it. EXPLAIN explained to me that it was taking the cartesian product of the two users "tables", and then whittling them down. The cartesian product was a gigantic number, and I'm doubting it would ever finish. Is there a better way to go about creating the relationship?
>
> I then had the idea to not create a Follower type, but just create the relationship via the followers CSV load. To do this, I tried removing all follower nodes:
>
> MATCH (n:Follower) DELETE n

-> this very probably also results in a heap overrun, try

> MATCH (n:Follower) WITH n limit 1000000 DELETE n RETURN count(*)

until it returns 0

> This also took some time, until the browser page pointed at Neo4j reported that Neo4j was no longer responding. I was unable to connect via neo4j-shell as well (timeout), and the server didn't respond to SIGTERM or SIGINT, so I had to give it a SIGKILL. Before I killed it, I observed a lot of CPU load on part of the java process. Is there something I'm missing about why Neo4j would freeze up like this?

I'm really a novice at Neo4j, and I am probably improperly applying a lot of knowledge from the RDBMS world. Does anyone have any pointers or advice for what I'm doing?

I'm running Neo4j 2.2.0 on Arch Linux. Please let me know if you need more information from me.

Thanks,
Rob

Cheers, Michael