stackoverflooooooow

Questions | Tags | Users | Badges | Unanswered | Ask Question

Neo4j Index created by constraint

asked   4 months ago
viewed   54 times
active   4 months ago

▲

0

▼

★

I am still trying to resolve my speed issue (shown here: Cypher MATCH query speed).

One thing I noticed is that while I am importing the data with a unique constraint (proven by the below).

> neo4j-sh (?)$ create index on :Person(username);
>
> QueryExecutionKernelException: Label 'Person' and property 'username' have a unique constraint defined on them, so an index is already created that matches this.

When I try to view the indexes in shell, I get the following:

> neo4j-sh (?)$ index --indexes
> Node indexes:
>
> Relationship indexes:

Are autogenerated indexes not supposed to show up? How can I verify that the unique constraint is in-fact indexing the username?

The main problem (as shown in the above link) is that the below simple query is taking 36 seconds (with an eager call) and twice that time when switched to a non-eager call.

```
USING PERIODIC COMMIT 15000
LOAD CSV WITH HEADERS FROM "file:d:/messages.csv" AS line
MATCH (a:Geotagged { username: line.sender }) – [r:MSGED] –> (b:Geotagged { username
RETURN NULL;
```

Note, this is excluding the SET call I was originally trying to use, I removed it and the MATCH alone is taking forever.

Additionally, I have also increased the pagecache to several times what I should need and saw no change.

EDIT 1 The nodes labeled with 'Geotagged' are ALSO labeled as 'Person'. All nodes are 'Person', some just happen to also be 'Geotagged'.

neo4j   cypher

share improve this question

asked Apr 9 at 17:07
Brooks
113 ● 11

add a comment

2 Answers

active   oldest   votes

▲

2

▼

✓

Have you used an uniqueness constrain with the Geotagged label as well as the Person label? I found that a uniqueness constrain on both labels increased speed greatly.

share improve this answer

answered Apr 9 at 17:15
Mel18
75 ● 4

Linked

2 Cypher MATCH query speed

Related

1 How to create Index on already created node using Cypher Query Language without using label in Neo4j

2 Neo4j 2.0.0M04, using Cypher and new indexes

2 neo4j import slowing down

0 neo4j cypher Error: org.neo4j.graphdb.NotFoundException: Unknown identifier `b` after index created

My import/upload process is to set the unique constraint on Person:username and then upload everything, all under the Person label. In that same query, I have a series of CASE WHEN + FOREACH statements that examine that nodes properties and add a specific label to it. You're suggesting I might need to constrain on each individual label? – Brooks Apr 9 at 17:19

1   Yes. I had a similar issue. I had only 3 choices for my secondary label, not sure how many you are adding. I found that adding the uniqueness constraint for each of the three secondary labels increased speeds exponentially. – Mel18 Apr 9 at 19:45

add a comment

▲

1

▼

You are using the `index` command for legacy indexes, **use `schema` to list schema indexes and constraints**.

Also if you match by `:Geotagged(username)` you have to have an index for that combination:

```
create index on :Geotagged(username);
```

or match on `:Person(username)` instead.

share improve this answer

answered Apr 9 at 20:13
Michael Hunger
25.2k ● 2 ● 21 ● 41

Michael, thank you! I was struggling with that. Is there a way to create the index on already existing nodes? It seems like the create index command simply adds the index and doesn't actually index the nodes. – Brooks Apr 9 at 20:19

I started a writeup, can you provide me some feedback if that would have helped you? gist.asciidoctor.org/… – Michael Hunger Apr 9 at 22:50

An schema index will always also index existing nodes with that label and property combination. – Michael Hunger Apr 9 at 22:51

Hi Michael, i'd be happy to review it and provide feedback, but I couldn't access the document, I think the URL was incorrect? – Brooks Apr 9 at 23:03
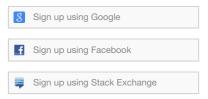
I ran "create index on :Geotagged(username)" and it was done in less than a second, as if there were no nodes to index. That should have indexed on all existing nodes? I assume that would have taken a few minutes? – Brooks Apr 9 at 23:04

add a comment

Your Answer

Post Your Answer

By posting your answer, you agree to the privacy policy and terms of service.

Not the answer you're looking for? Browse other questions tagged `neo4j` `cypher` or ask your own question.