Cypher MATCH query speed

2

1

I have Neo4j installed on a windows machine with 12 processors and 64GB ram. I did not change any of the memory settings that Neo4j allows for.

My database has 3.8m nodes, 210,000 of which are labeled as Geotagged and a total of 650,000 relationships. I am trying to run the following query and I am wondering if this is a really intensive query that will likely take quite a while.

Messages.csv is my relationship file. The relationships have already been created, but as I could not figure out how to combine the relationship creation with the below Distance generation, I am loading and running through the relationship file twice.

```
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
SET r.Distance = (2 * 6371 * asin(sqrt(haversin(radians(toFloat(b.statusLat) - toFlo
```

The initial relationship generation takes about 3-5 minutes. I let the above run for over an hour and it still was not complete. I ran a similar algorithm (though it had a few more trig calls in it) on the same initial db and let it run for over 18 hours and still had not completed.

My question: Is this a very intensive query? Am I not giving it enough time? And more importantly, is there a way I can optimize this?

I tried adding "WHERE NOT HAS(r.Distance)" to exclude node pairs that the algorithm has already set the Distance on, though I am unsure if the MATCH is a one-time match or if it will MATCH for each line in the CSV file?

Any thoughts on this would really be appreciated.

neo4j cypher

share improve this question

asked Apr 8 at 17:28
Brooks
113 ● 11

add a comment

2 Answers

active oldest votes

2

One way that I would start to debug is to put a limit on it using `WITH`:

```
USING PERIODIC COMMIT 15000
LOAD CSV WITH HEADERS FROM "file:d:/messages.csv" AS line
WITH line LIMIT 100
MATCH (a:Geotagged { username: line.sender }) - [r:MSGED] -> (b:Geotagged { username
SET r.Distance = (2 * 6371 * asin(sqrt(haversin(radians(toFloat(b.statusLat) - toFlo
```

With that you can change the `LIMIT` number to see how the performance degrades as the limit increases.

Also, is the `username` property indexes for the `Geotagged` label? If not it definitely should be, like this:

```
CREATE INDEX ON :Geotagged(username)
```

asked	4 months ago
viewed	52 times
active	4 months ago

Linked

0 Neo4j Index created by constraint

0 Big data import into neo4j

Related

1 Cypher query return related nodes as children

0 What is the appropriate cypher query?

2 Poor performance of Neo4j Cypher query for transitive closure

0 Neo4j Cypher query fails and return with an Unknown Error

3 Cypher match path based on previous relationship

0 Limiting the number of matches in Neo4j Cypher

2 Cypher query resultset growing over subsequent runs?

If it's unique and you want the database to enforce that:

```
CREATE CONSTRAINT ON (g:Geotagged) ASSERT g.username IS UNIQUE
```

share improve this answer

answered Apr 8 at 18:24
Brian Underwood
4,364 ● 4 ● 18

Sorry, yes I have already constrained the username to be unique, though I was under the impression the CONSTRAINT automatically creates the index....? – Brooks Apr 8 at 20:36

Yup, sorry, I should have made that clear! – Brian Underwood Apr 8 at 21:37

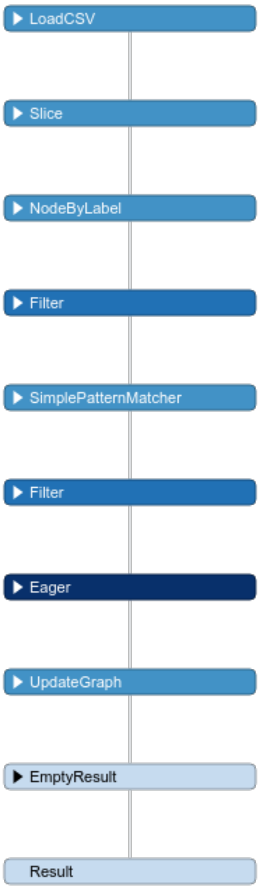
add a comment

2

This is additional to Brian's reply:

Your statement's query plan shows `EAGER`, to verify run

```
EXPLAIN explain LOAD CSV WITH HEADERS FROM "file:d:/messages.csv" AS line
WITH line LIMIT 100
MATCH (a:Geotagged { username: line.sender }) - [r:MSGED] -> (b:Geotagged { username
SET r.Distance = (2 * 6371 *asin(sqrt(haversin(radians(toFloat(b.statusLat) - toFloa
```



Eagerness in `LOAD CSV` is pretty bad, see the these blog posts why:

- http://www.markhneedham.com/blog/2014/10/23/neo4j-cypher-avoiding-the-eager/
- http://jexp.de/blog/2014/10/load-cvs-with-success/

Following Mark's suggested and replacing the `MATCH/SET` with a `MERGE ON MATCH SET` we can refactor that into:

1 Cypher - matching two different possible paths and return both

0 cypher query BadInputException

0 Cypher query optimisation - Utilising known properties of nodes

Hot Network Questions

How to ask dumb questions

Construct a polynomial with given roots

What are some techniques to avoid touching a wet tent wall?

Preventing a Print expression from generating a new cell each time it executed

Using siunitx, \ohm results in an italic Omega

The Programming Language Quiz

Two definitions of Green's function

How to use Piano pedals without using the foot?

Password strength of random words and brainflayer

Is there a single-word antoym for "Opportunity"?

Combining several symmetric ciphers using XOR

Has any Star Trek character appeared in all three division uniform colors?

Did a Kamikaze ever impact the hull of a ship, as opposed to the deck?

Can you open source firmware to closed hardware?

Straight line in TikZ with 'continued' marking in the middle

Using Emergency Fund to Sell Upside-down Car

Name for a device purposefully put together from faulty parts

Could a communication satellite that fails to reach GEO be repurposed?

Weird performance increase in simple benchmark

Is Let us = Let's?

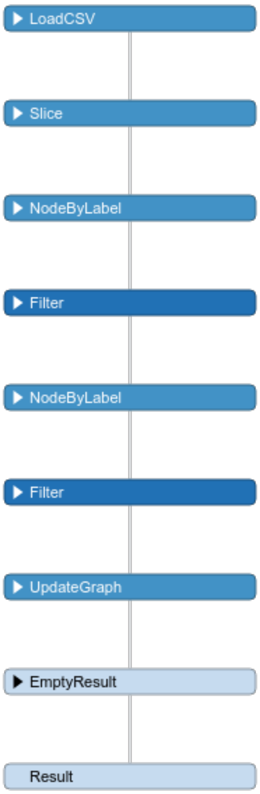
Is "user-select: none" bad UX?

My 4.5 yrs old son has no dominant hand

Double integral problem with sin

Does developing on ARM Processors from different vendors differ much?

```
explain LOAD CSV WITH HEADERS FROM "file:d:/messages.csv" AS line
WITH line LIMIT 100
MATCH (a:Geotagged { username: line.sender }), (b:Geotagged { username: line.recipie
MERGE (a)-[r:MSGED]->(b)
ON MATCH SET r.Distance = (2 * 6371 * asin(sqrt(haversin(radians(toFloat(b.statusLat
```



And `eager` has vanished.

share improve this answer

edited Apr 8 at 19:48

answered Apr 8 at 19:42
Stefan Armbruster
23.1k ● 2 ● 28 ● 51

Stefan, thank you, I was unaware of the PROFILE capability or even of the issue with eager. I've since changed the query as you suggested and ran with limits of 10 and 100. Both made 0 modifications (i.e. those lines did not represent messages sent from one geotagged user to another geotagged user). 10 lines took 11.5sec and 100 lines took 71 seconds. The file has 3,712,112 lines, so extrapolating based on the 71sec/100 lines, that's roughly 30 days. Is that surprising at all or is that normal? Just FYI, my original query ran 100 lines in 36 seconds (half that of the revised syntax). – Brooks Apr 8 at 21:35

that's way too slow. I guess you can gain a lot by switching over to Linux. – Stefan Armbruster Apr 9 at 8:00

Hi Stefan, is the windows implementation really that unreliable? I've also changed the pagecache with no change. I also posted another question on this, but the index created by the unique constraint doesn't show up in shell when I run 'index --indexes', is that normal? – Brooks Apr 9 at 17:09

1 By far the most of the production installations I'm aware of are running on Linux. In Neo4j <=2.1.x, the filebuffer cache is off-heap on Linux and on-heap in Windows - that is one important difference. In 2.2 the filebuffer cache has been superseded with the page cache which is off heap independent of the OS. I also have the impression (without knowing the facts) that memory management is far better in Linux vs. Windows - but I don't want to start a flame war here. – Stefan Armbruster Apr 9 at 18:45

You still have NodeByLabel scan stefan :) – Michael Hunger Apr 9 at 20:59

add a comment

Your Answer

Sign up or log in

Sign up using Google

Sign up using Facebook

Sign up using Stack Exchange

Post as a guest

Name

Email

required, but never shown

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.

question feed

tour help blog chat data legal privacy policy work here advertising info mobile **contact us** **feedback**

TECHNOLOGY

- Stack Overflow
- Server Fault
- Super User
- Web Applications
- Ask Ubuntu
- Webmasters
- Game Development
- TeX - LaTeX
- Programmers
- Unix & Linux
- Ask Different (Apple)
- WordPress Development
- Geographic Information Systems
- Electrical Engineering
- Android Enthusiasts
- Information Security
- Database Administrators
- Drupal Answers
- SharePoint
- User Experience
- Mathematica
- Salesforce
- ExpressionEngine® Answers
- **more (13)**

LIFE / ARTS

- Photography
- Science Fiction & Fantasy
- Graphic Design
- Movies & TV
- Seasoned Advice (cooking)
- Home Improvement
- Personal Finance & Money
- Academia
- **more (9)**

CULTURE / RECREATION

- English Language & Usage
- Skeptics
- Mi Yodeya (Judaism)
- Travel
- Christianity
- Arqade (gaming)
- Bicycles
- Role-playing Games
- **more (21)**

SCIENCE

- Mathematics
- Cross Validated (stats)
- Theoretical Computer Science
- Physics
- MathOverflow
- Chemistry
- Biology
- **more (5)**

OTHER

- Stack Apps
- Meta Stack Exchange
- Area 51
- Stack Overflow Careers

rev 2015.8.20.694