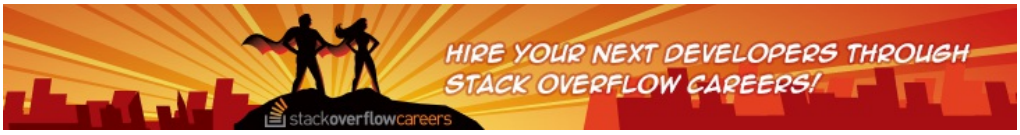
Neo4j: Fast query for getting relationships between a set of nodes



asked 4 months ago

viewed 131 times

active 4 months ago

1

I'm looking for a fast Cypher statement that returns all relationships between a known set of nodes (I have their Neo4j ID's), so that I can assemble the subgraph for that particular set of nodes. I'm working within a label called `label` which has around 50K nodes and 800K edges between these nodes.

I have several working approaches for this, but none are fast enough for my application, even at small set sizes (less than 1000 nodes).

For example, the following statement does the trick:

```
MATCH (u:label)-[r]->(v:label)
WHERE (ID(u) IN {ids}) AND (ID(v) IN {ids})
RETURN collect(r)
```

Where `{ids}` is a list of numeric Neo4j ids given as parameter to the Py2Neo `cypher.execute(statement, parameters)` method. The problem is that it takes around 34 seconds for a set of 838 nodes, which returns all 19K relationships between them. I realize the graph is kind of dense, but it takes 1.76 seconds for every 1000 edges returned. I just don't think that's acceptable.

If I use the START clause instead (shown below), the time is actually a little worse.

```
START u=node({ids}), v=node({ids})
MATCH (u:label)-[r]->(v:label)
RETURN collect(r)
```

I've found many similar questions/answers, however they all fall short in some aspect. Is there a better statement for doing this, or even a better graph schema, so that it can scale to sets of thousands of nodes?

UPDATE

Thanks for the fast replies. First, to run my current query for 528 nodes as input (`len(ids)=528`) it takes **32.1 seconds** and the query plan is below.

```
NodeByIdSeek: 528 hits
Filter      : 528 hits
Expand(All) : 73,773 hits
Filter      : 73,245 hits
Projection  : 0 hits
Filter      : 0 hits
```

Brian Underwood's query, with the same input, takes **27.8 seconds**. The query plan is identical, except for the last 2 steps (Projection and Filter), which don't exist on for his query. However the db hits sum is the same.

Michael Hunger's query takes **26.9 seconds** and the query plan is identical to Brian's query.

I've restarted the server between experiments to avoid cache effects (there's probably a smarter way to do it). I'm also querying straight from the web interface to by pass possible bottlenecks in my code and the libs I'm using.

Bottomline, Neo4j seems smart enough to optimize my query, however it's still pretty slow even with fairly small sets. Any suggestions?

performance neo4j cypher

Blog

Why Stack Overflow is a Good Workplace for Women

Mutually Human
based in Grand Rapids, MI

Who you'll work with...

Sam Bleckley

Aaron Christy

E.J. Dyksen
Software Craftsman
osx swift ruby

view company page

Related

0 Getting all relationships among neighbors of a node

1 Create relationship between nodes having same property value in common, using one Cypher query

1 Neo4j: query for all paths with specified relationships

0 Neo4j cypher getting nodes that are related to another node using ALL relations

0 neo4j performance with 4M nodes and 29M relations

1 Speedup relationship and node creation using cypher in Neo4j

share improve this question

edited Apr 10 at 5:17

asked Apr 8 at 3:28
LuamCT
6 ● 2

Is there any performance change if you change the first line to `MATCH (u:label)-[r]-(v:label)` ? – jag426 Apr 8 at 3:37

can you share your query plan? you can get it by prefixing your query with `PROFILE` – Michael Hunger Apr 8 at 8:11

Don't return collect(r) just r – Michael Hunger Apr 8 at 8:11

What's your use-case for this? (you certainly can't display this graph) Maybe you should change your strategy and only get the portion of the graph your processing? If you let me know, what your goal is, it will be easier to help – joe Apr 8 at 8:24

@jag426 It took over 7x longer for the same input of 528 nodes. I would also have to handle duplicates. – LuamCT Apr 9 at 20:21

add a comment

2 Answers

active oldest votes

1

I think the problem is that the query is doing a Cartesian product to get all combinations of the 838 node, so you end up searching 838*838=702,244 combinations.

I'm curious how this would perform:

```
MATCH (u:label)-[r]->(v:label)
WHERE (ID(u) IN {ids})
WITH r, v
WHERE (ID(v) IN {ids})
RETURN collect(r)
```

Also, why do the `collect` at the end?

share improve this answer

answered Apr 8 at 5:12
Brian Underwood
4,364 ● 4 ● 18

The `collect` made it easier to consume the results in the python's lib, but I've already dropped it. It seems Neo4j optimizes my query to be something similar to yours. Please check my update. – LuamCT Apr 9 at 20:09

Crazy thought: What if you returned all of the nodes and `collect`ed all of the relationships for those nodes and returned all of the results. Then try filtering out the relationships that you don't need. – Brian Underwood Apr 10 at 12:48

But that would be like doing Neo4j work for it, right? I moved to graph databases exactly to avoid this type of things. – LuamCT Apr 14 at 20:49

Yeah, fair enough point. I'm more curious of how it works. This is maybe an optimization that Neo4j should implement internally. – Brian Underwood Apr 15 at 13:08

add a comment

{ USE STACK OVERFLOW TO FIND THE BEST DEVELOPERS } stackoverflowcareers

0

How big are your id-lists?

Try this:

```
MATCH (u) WHERE ID(u) IN {ids}
WITH u
MATCH (v)-[r]->(v)
WHERE ID(v) IN {ids}
RETURN count(*)


MATCH (u) WHERE (ID(u) IN {ids})
WITH u
MATCH (v)-[r]->(v)
WHERE ID(v) IN {ids}
RETURN r
```

Also try to create a query plan by prefixing your query with `PROFILE` then you see where the cost is.

share improve this answer

answered Apr 8 at 8:13

1 Neo4j cypher query performance when ordering a large data set
1 how are neo4j caches speeding up queries?
2 Efficient cypher query matching subgraph connecting two groups of nodes
6 How can I optimise a Neo4j MERGE query on a node with many relationships?

Hot Network Questions

Name for a device purposefully put together from faulty parts
Noun form of "extant"?
Combining several symmetric ciphers using XOR
How does java.util.EnumSet<E> work?
Where did the nickname of 'Bones' for McCoy come from?
What are the advantages and disadvantages of an oblique wing over a swing wing
Draw horizontal lines like open and close brackets
Does adult neurogenesis occur only in the olfactory bulb and the hippocampus?
Does Lebanon ban the diary of Anne Frank?
Is it modest to compliment a woman on her modesty?
Religion After the Discovery of a Multiverse
Capture output without redirection and leave it on terminal too
Dullness vs. going overboard: Should I be calling people 'enfants terribles' in an academic paper?
How to use Piano pedals without using the foot?
Aerofoil that gives reasonably good lift for both flow directions: forward and backward
How to show pagination "go to page" text box?
How to ask dumb questions
Does touching the tent outer wall from the inside make it leak?
How can I grant a non-Super User group permission to force check-in of articles?
Construct a polynomial with given roots
A short, brutal riddle
Weird performance increase in simple benchmark
How do I know when my sauce is reduced enough?
Why are my tomatoes cracking and what can I do?

answered Apr 8 at 8:13

Michael Hunger
25.2k ● 2 ● 21 ● 41

I'm assuming you meant `MATCH (u)-[r]->(v)`. However the performance is not significantly better. Please check my update. – LuamCT Apr 9 at 20:11

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 performance neo4j cypher or ask your own question.

question feed

site design / logo © 2015 Stack Exchange Inc; user contributions licensed under cc by-sa 3.0 with attribution required