Ten. Million. Questions. Let's celebrate all we've done together.

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free. Take the 2-minute tour ×

Seeking Neo4J Cypher query for long but (nearly) unique paths

asked 4 months ago
viewed 45 times
active 4 months ago

3

We have a Neo4J database representing an evolutionary process with about 100K nodes and 200K relations. Nodes are individuals in generations, and edges represent parent-child relationships. The primary goal is to be able to take one or nodes of interest in the final generation, and explore their evolutionary history (roughly, "how did we get here?").

The "obvious" first query to find all their ancestors doesn't work because there are just too many possible ancestors and paths through that space:

```
match (a)-[:PARENT_OF*]->(c {is_interesting: true})
return distinct a;
```

So we've pre-processed the data so that some edges are marked as "special" such that *almost* every node has at most one "special" parent edge, although occasionally both parent edges are marked as "special". My hope, then, was that this query would (efficiently) generate the (nearly) unique path along "special" edges:

```
match (a)-[r:PARENT_OF* {special: true}]->(c {is_interesting: true})
return distinct a;
```

This, however, is still unworkably slow.

This is frustrating because "as a human", the logic is simple: Start from the small number of "interesting" nodes (often 1, never more than a few dozen), and chase back along the almost always unique "special" edges. Assuming a very low number of nodes with two "special" parents, this should be something like O(N) where N is the number of generations back in time.

In Neo4J, however, going back 25 steps from a unique "interesting" node where *every* step is unique, however, takes 30 seconds, and once there's a *single* bifurcation (where both parents are "special") it gets worse much faster as a function of steps. 28 steps (which gets us to the first bifurcation) takes 2 minutes, 30 (where there's still only the one bifurcation) takes 6 minutes, and I haven't even thought to try the full 100 steps to the beginning of the simulation.

Some similar work last year seemed to perform better, but we used a variety of edge labels (e.g., `(a)-[:SPECIAL_PARENT_OF*]->(c)` as well as `(a)-[:PARENT_OF*]->(c)`) instead of using data fields on the edges. Is querying on relationship field values just not a good idea? We have quite a few different values attached to a relationship in this model (some boolean, some numeric) and we were hoping/assuming we could use those to efficiently limit searches, but maybe that wasn't really the case.

Suggestions for how to tune our model or queries would be greatly appreciated.

Update I should have mentioned, this is all with Neo4J 2.1.7. I'm going to give 2.2 a try as per Brian Underwood's suggestion and will report back.

neo4j cypher

share improve this question edited Apr 6 at 14:59

asked Apr 6 at 5:41
Nic McPhee
16 ● 5

I'd really like to see an answer to this myself. I've definitely had some Cypher queries with open-ended path lengths which seemed like they shouldn't be traversing very much but which take a while. – Brian Underwood Apr 6 at 7:24

add a comment

Blog

Why Stack Overflow is a Good Workplace for Women

Looking for a job?

Software Engineer in Test
ReadyTalk
Denver, CO / relocation
javascript selenium

.NET Developer at software company
Bluesun
Mississauga, ON, Canada
c# .net

Support Engineer - Spanish Speaker
Incognito Software
Vancouver, BC, Canada
unix linux

Lead iOS developer at Hushmail — $100K - $150K
Hushmail
Vancouver, BC, Canada / remote / relocation
ios objective-c

Related

0	neo4j not reusing existing vertex in cypher create unique query
4	Cypher Query in Neo4j Returns 'undefined'
0	Neo4j crashes on 4th degree Cypher query
0	Slow neo4j Cypher Queries, using Baconator query for less-short paths than shortest path
0	Creating unique path in Neo4J using Cypher and SDN @Query annotation
2	Neo4j Cypher alternative paths
2	Cypher MATCH query speed

2 Answers

active oldest votes

2

I've had some luck with specifying a limit on the path length. So if you know that it's never more than 30 hops you might try:

```
MATCH (c {is_interesting: true})
WITH c
MATCH (a)-[:PARENT_OF*1..30]->c
RETURN DISTINCT a
```

Also, is there an index on the `is_interesting` property? That could also cause slowness, for sure.

What version of Neo4j are you using? If you are using or if you upgrade to 2.2.0, you get to use the new query profiling tools:

http://neo4j.com/docs/2.2.0/how-do-i-profile-a-query.html

Also if you use them in the web console you get a nice graph-ish tree thing (technical term) showing each step.

share improve this answer

edited Apr 6 at 18:01

answered Apr 6 at 7:28
Brian Underwood
4,364 ● 4 ● 18

We definitely would like to have long paths. We've been doing short paths as a workaround, but one definite goal is to be able to push all the way back to the beginning of the simulation. I think there is an index on the `is_interesting` property. Running `schema ls` lists it, at least. We are using 2.1.7 (sorry for not listing the version), but I'm definitely going to try 2.2 and see if that helps the situation, or if the profiling tools might help clarify what's going on. Thanks for the suggestions. – Nic McPhee Apr 6 at 15:14

Switching to 2.2 didn't improve the performance, but the profiling tools did help make it clear what the problem is. It is generating the full set of paths *without* using the `is_interesting` property, and *then* filtering with the `is_interesting` property. That initial set of paths quickly becomes huge, so the queries take forever. I'm guessing that if we had a special edge with `IS_INTERESTING` as its label, then it would be able to use that in pre-filtering the list of paths, thus speeding things up quite a bit. – Nic McPhee Apr 6 at 15:40

Maybe also use a WITH clause to filter first? – Brian Underwood Apr 6 at 15:50

I like the idea, but I'm not sure how to apply it. I know how to use a `WITH` to limit searching on *nodes*, but I'm not sure how to use it to limit searching on *edges*. Is there some way I can use `WITH` to say "only search paths with edges that have `is_interesting=true`? – Nic McPhee Apr 6 at 17:10

1 Well I was thinking about searching on nodes first. So you can see in my edit I filter out just nodes with `has_interesting` and then search for paths. If you're looking for relationships with that property you can't do that, but I thought node properties was your first use case – Brian Underwood Apr 6 at 18:02

show **3** more comments

$ git push origin stackoverflowcareers   Add repos

0

After exploring things with the profiling tools in Neo4J 2.2 (thanks to Brian Underwood for the tip) it's pretty clear that (at the moment) Neo4J doesn't do any pre-filtering on edge properties, which leads to nasty combinatorial explosions with long paths.

For example the original query:

```
match (a)-[r:PARENT_OF* {special: true}]->(c {is_interesting: true})
return distinct a;
```

finds *all* the paths from `a` to `c` and *then* eliminates the ones that have edges that aren't `special`. Since there are many millions of paths from `a` to `c`, this is totally infeasible.

If I instead add a `IS_SPECIAL` edge wherever there was a `PARENT_OF` edge that had `{special: true}`, then the queries become really fast, allowing me to push back around 100 generations in under a second.

This query creates all the new edges:

```
match (a)-[r:PARENT_OF {special: true}]->(b)
create (a)-[:IS_SPECIAL]->(b);
```

and takes under a second to add 91K relationships in our graph.

Then

```
match (c {is_interesting: true})
with c
match (a)-[:IS_SPECIAL*]->(c)
return distinct a;
```

0 Cypher query optimisation - Utilising known properties of nodes

3 Cypher query for shortest path with end node constraint

1 How to correctly use conditionals like IF or CASE in Cypher query language (Neo4J) to successfully create relationships?

Hot Network Questions

Is it modest to compliment a woman on her modesty?

Manager sounds upset every time I inform him of a (minor) obstacle

How to ask dumb questions

Dullness vs. going overboard: Should I be calling people 'enfants terribles' in an academic paper?

Prove the theorem on analytic geometry in the picture.

Using Emergency Fund to Sell Upside-down Car

The Programming Language Quiz

Double integral problem with sin

Aerofoil that gives reasonably good lift for both flow directions: forward and backward

Justification that 'good exposition' is uniquely defined

Proving that all integers are even or odd

`cond` with less redundancy

Password strength of random words and brainflayer

How to show pagination "go to page" text box?

Is it possible to automate tests for in progress sprint

Approximation of Borel sets by a countable collection of majorants

Select polygons one by one and export them using arcpy

Why are carpenter's pencils flat?

How can I grant a non-Super User group permission to force check-in of articles?

How to deal with people looking at their mobile phone during my presentation?

How does java.util.EnumSet<E> work?

Why do we still use keys to start cars? why not passwords?

Does adult neurogenesis occur only in the olfactory bulb and the hippocampus?

Is it 'Batmen' or 'Batmans'?

takes under a second to find the 112 nodes along the "special" path back from a unique target node `c`. Matching `c` first and limiting the set of nodes using `with c` seems to also be

important, as Neo4J doesn't appear to pre-filter on node properties either, and if there are several "interesting" target nodes things get a lot slower.

share improve this answer

answered Apr 6 at 18:53
Nic McPhee
16 ● 5

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			LIFE / ARTS	CULTURE / RECREATION	SCIENCE	OTHER
Stack Overflow	Programmers	Database Administrators	Photography	English Language & Usage	Mathematics	Stack Apps
Server Fault	Unix & Linux	Drupal Answers	Science Fiction & Fantasy	Skeptics	Cross Validated (stats)	Meta Stack Exchange
Super User	Ask Different (Apple)	SharePoint	Graphic Design	Mi Yodeya (Judaism)	Theoretical Computer Science	Area 51
Web Applications	WordPress Development	User Experience	Movies & TV	Travel	Physics	Stack Overflow Careers
Ask Ubuntu	Geographic Information Systems	Mathematica	Seasoned Advice (cooking)	Christianity	MathOverflow	
Webmasters	Electrical Engineering	Salesforce	Home Improvement	Arqade (gaming)	Chemistry	
Game Development	Android Enthusiasts	ExpressionEngine® Answers	Personal Finance & Money	Bicycles	Biology	
TeX - LaTeX	Information Security	**more (13)**	Academia	Role-playing Games	**more (5)**	
			more (9)	**more (21)**		

site design / logo © 2015 Stack Exchange Inc; user contributions licensed under cc by-sa 3.0 with attribution required
rev 2015.8.20.694