Neo4j 2.1.7 - 70x Increase in database accesses, 1000x slowdown when traversing a path of length 10 vs a length of 9. #4921

Open **bunkat** opened this issue 11 days ago · 5 comments

bunkat commented 11 days ago

I posed this earlier in the Google Groups but got no reply. This is now becoming a blocking issue for our deployment. This is with 2.1.7, I'm unable to move to 2.2.x due to blocking index and query issues.

I have a list of history nodes attached to various objects that are connected together in a history hierarchy. I traverse the list for a particular object by following HISTORY relationships with the corresponding object id. I recently noticed that the queries are timing out and ran into the following during my investigation.

My query is as follows:

```
MATCH (i:file { id: "foobar" })
WITH i
MATCH (i)-[:HISTORY*1..10 {id: i.id}]->(a:activity)
WITH DISTINCT i, a SKIP 0 LIMIT 10

RETURN a
```

This query is just locating a particular starting node by schema index and then trying to follow the history chain, collecting up all of the activity nodes that it finds. If I look for a chain with a maximum of 2 nodes ([:HISTORY*1..2]) it returns quickly. Same for 3, 4, 5, 6, 7, 8, and 9. However, if I try to run the query above, it takes over 200s to actually complete. The crazy part is that this particular case only has two total activities in the chain.

Why does looking for a maximum of 9 return instantly but 10 take over 200s? Especially since there is only a maximum of 2 to be found in either case?

How should I be writing this query to be more perfomant?

I haven't tested with Neo4j 2.2.1 yet since I can't deploy that to production yet. I'm looking for a solution for 2.1.7.

PROFILE WITH 9

```
>   MATCH (i:file { id: "foobar" })
>   WITH i
>
>   MATCH (i)-[:HISTORY*1..9 {id: "foobar"}]->(a:activity)
>   WITH DISTINCT i, a SKIP 0 LIMIT 9
>
>   RETURN COUNT(a);
+----------+
| COUNT(a) |
+----------+
| 2        |
+----------+
```

Labels
None yet

Milestone
No milestone

Assignee
No one assigned

Notifications
Subscribe
You're not receiving notifications from this thread.

3 participants

```
1 row

ColumnFilter
  |
  +EagerAggregation
    |
    +Slice
      |
      +Distinct
        |
        +Filter
          |
          +PatternMatcher
            |
            +SchemaIndex

+------------------+------+--------+-------------------+------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|         Operator | Rows | DbHits |       Identifiers |                                                                                                                                                                  Other |
+------------------+------+--------+-------------------+------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|     ColumnFilter |    1 |      0 |                   |                                                                                                                                                 keep columns COUNT(a) |
| EagerAggregation |    1 |      0 |                   |                                                                                                                                                                        |
|            Slice |    2 |      0 |                   |                                                                                                                                              {  AUTOINT2}; {  AUTOINT3} |
|         Distinct |    2 |      0 |                   |                                                                                                                                                                        |
|           Filter |    2 |      8 |                   | (hasLabel(a:activity(18)) AND all(  UNNAMED-7541039006321584207 in   UNNAMED-2669272078697603796 where Property(  UNNAMED-7541039006321584207,id(6)) == {  AUTOSTRING1})) |
|   PatternMatcher |    2 |  72838 | i, a,   UNNAMED60 |                                                                                                                                                                        |
|      SchemaIndex |    1 |      2 |              i, i |                                                                                                                                            {  AUTOSTRING0}; :file(id) |
+------------------+------+--------+-------------------+------------------------------------------------------------------------------------------------------------------------------------------------------------------------+

Total database accesses: 72848
```

PROFILE WITH 10

```
>   MATCH (i:file { id: "foobar" })
>   WITH i
>
>   MATCH (i)-[:HISTORY*1..10 {id: "foobar"}]->(a:activity)
>   WITH DISTINCT i, a SKIP 0 LIMIT 10
>
>   RETURN COUNT(a);
+----------+
| COUNT(a) |
+----------+
| 2        |
+----------+
1 row

ColumnFilter
  |
  +EagerAggregation
    |
    +Slice
      |
      +Distinct
        |
        +Filter
          |
          +PatternMatcher
            |
            +SchemaIndex

+------------------+------+---------+-------------------+------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|         Operator | Rows |  DbHits |       Identifiers |                                                                                                                                                                  Other |
+------------------+------+---------+-------------------+------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|     ColumnFilter |    1 |       0 |                   |                                                                                                                                                 keep columns COUNT(a) |
| EagerAggregation |    1 |       0 |                   |                                                                                                                                                                        |
|            Slice |    2 |       0 |                   |                                                                                                                                              {  AUTOINT2}; {  AUTOINT3} |
|         Distinct |    2 |       0 |                   |                                                                                                                                                                        |
|           Filter |    2 |       8 |                   | (hasLabel(a:activity(18)) AND all(  UNNAMED-6855841532423436547 in   UNNAMED-4126485615317615815 where Property(  UNNAMED-6855841532423436547,id(6)) == {  AUTOSTRING1})) |
|   PatternMatcher |    2 | 5084198 | i, a,   UNNAMED60 |                                                                                                                                                                        |
|      SchemaIndex |    1 |       2 |              i, i |                                                                                                                                            {  AUTOSTRING0}; :file(id) |
+------------------+------+---------+-------------------+------------------------------------------------------------------------------------------------------------------------------------------------------------------------+

Total database accesses: 5084208
```

jexp commented 11 days ago Collaborator

Hi Bill, sorry that you didn't get an answer on the google group. Must have slipped through the cracks.

I think cypher chooses the PatternMatcher because of the relationship-property. And the "PatternMatcher" has some unhealthy exponential behavior on longer paths.

When you change the query to filter for the relationship-id later, it chooses the (bi-directional) traversal-matcher which is way more efficient.

```
profile
MATCH (i:file { id: "foobar" })-[rel:HISTORY*1..10]->(a:activity)
WHERE ANY (r in rel WHERE r.id = "foobar")
WITH DISTINCT i, a
SKIP 0 LIMIT 9
RETURN COUNT(a);

+------------------+------+--------+-------------+-----------------------------------------------------------------------------------------------+
|         Operator | Rows | DbHits | Identifiers |                                                                                         Other |
+------------------+------+--------+-------------+-----------------------------------------------------------------------------------------------+
|     ColumnFilter |    1 |      0 |             |                                                                       keep columns COUNT(a) |
| EagerAggregation |    1 |      0 |             |                                                                                             |
|            Slice |    0 |      0 |             |                                                                  {  AUTOINT2}; {  AUTOINT3} |
|         Distinct |    0 |      0 |             |                                                                                             |
|           Filter |    0 |      0 |             | (hasLabel(a:activity) AND any(r in rel where Property(r,id(0)) == {  AUTOSTRING1})) |
| TraversalMatcher |    0 |      1 |             |                                                                  a,   UNNAMED31, a, rel |
+------------------+------+--------+-------------+-----------------------------------------------------------------------------------------------+
```

Please try it out and report back.

bunkat commented 11 days ago

That allows the query to work up until 15 instead of 9. At 16, it falls over again.

In general, is it a best practice to avoid relationship-property matching? I assumed it would be more efficient since paths can be eliminated immediately but it seems like that is incorrect.

jexp commented 11 days ago — Collaborator

Do you have a dataset to share?

It depends, checking on relt-types is faster as it doesn't require additional reads. But in general and more so in 2.2+ it is eagerly pruning paths

AshishBarde commented 11 days ago

Ok I attached database with this mail
 database.rar
<https://drive.google.com/file/d/0B8G1DciINdbtSFhjZ214dzRiTm8/view?usp=drive_web>
please reply me when u will get database
...

AshishBarde commented 11 days ago

can u give mi your skype id
...

Write Preview — Markdown supported — Edit in fullscreen

Leave a comment

Attach images by dragging & dropping, selecting them, or pasting from the clipboard.

Comment