



This repository Search

Explore Features Enterprise Blog

Sign up

Sign in

neo4j / neo4j

Watch

238

Star

1,885

Fork

730

# Neo4j 2.2.1: COST and RULE planners produce different results when using HEAD, TAIL, and LAST operations

New issue



Open

bunkat opened this issue Apr 18, 2015 · 3 comments



bunkat commented Apr 18, 2015

I have a query that is calculating a minimum value across a set of related nodes. I was running into some odd results and it turns out that the two query planners produce different results for the following query:

```
MATCH (p:project { id: "foobar" })
WITH p

MATCH (p)-[:MATERIAL]->(materials)-[:MATERIAL]->(m:material)
WITH m, [s IN (:schedule)<-[:SCHEDULE]-(:task)-[:MATERIAL]->(m) | HEAD(NODES(s))] AS sch
RETURN m, schedules
```

The COST planner decides that the `schedule` nodes are the HEAD elements while the RULE planner decides that the `task` nodes are the HEAD elements. This causes queries to fail since unexpected nodes are being returned.

Rewriting the query this way makes both planners agree.

```
WITH m, [s IN (m)<-[:MATERIAL]-(:task)-[:SCHEDULE]->(schedule) | LAST(NODES(s))] AS sch
```

Although the query is slightly ambiguous, the COST planner and RULE planner should always return the same results especially since the engine can decide to bail out of the COST planner based on previous statements in the query.

Labels

None yet

Milestone

No milestone

Assignee

No one assigned

2 participants



bunkat commented Apr 18, 2015

Looks like the COST and RULE planners can't agree on even simpler examples:

```
[c IN (s)<-[:SCHEDULE]-(:week) | LAST(NODES(c))] AS contacts
```

COST likes LAST to return the `week` nodes, RULE likes HEAD. Re-writing the query slightly to this makes both planners agree that the HEAD nodes are the week nodes.

```
[c IN (:week)-[:SCHEDULE]->(s) | HEAD(NODES(c))] AS contacts
```



jexp commented Apr 19, 2015

Collaborator

Hi Bill,  
I tried it for a simple dataset and it worked for me.

can you try to run this and share the results:

```
match (s:schedule)
return [c IN (s)<-[:SCHEDULE]-(:week) | NODES(c)] AS paths
```

@pontusmelke could that be related to the "getDegree" changes, where we would have to potentially

reverse paths if we look from the other side first? left-to-right vs. right-to-left ?

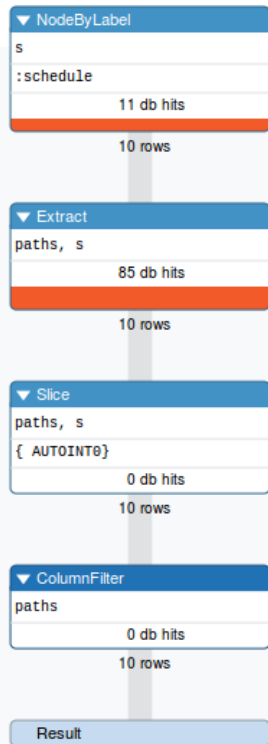


bunkat commented Apr 19, 2015

Yes, that query breaks as well between the two planners.

## RULE

```
PROFILE
PLANNER RULE
MATCH (s:schedule)
RETURN [c IN (s)<-[:SCHEDULE]-(:week) | NODES(c)] AS paths LIMIT 1
```



[[:week, :schedule], [:week, :schedule], ... ]

## COST

```
PROFILE
PLANNER COST
MATCH (s:schedule)
RETURN [c IN (s)<-[:SCHEDULE]-(:week) | NODES(c)] AS paths LIMIT 1
```



**Mats-SX** referenced this issue Apr 22, 2015

**Order of path elements is now always the written order of identifiers in the query. #4493**

**Merged**

[Sign up for free](#)

to join this conversation on [GitHub](#). Already have an account? [Sign in to comment](#)

