Neo4j's MERGE command on big datasets

asked 4 months ago
viewed 53 times
active 4 months ago

1

1

Currently, I am working on a project of implementing a Neo4j (V2.2.0) database in the field of web-analytics. After loading some samples, I'm trying to load a big data set (>1GB, >4M lines). The problem I am facing, is that the usage of the MERGE command takes exponentially more time as the data size grows. Online sources are ambiguous on what the best way is to load big sets of data when not every line has to be loaded as a node, and I would like some clarity on the subject. To emphasize, in this situation I am just loading the nodes; relations are the next step.

Basically there are three methods

i) Set a uniqueness constraint for a property, and create all nodes. This method was used mainly before the MERGE command was introduced.

```
CREATE CONSTRAINT ON (book:Book) ASSERT book.isbn IS UNIQUE
```

followed by

```
USING PERIODIC COMMIT 250
LOAD CSV WITH HEADERS FROM "file:C:\\path\\file.tsv" AS row FIELDTERMINATOR'\t'
CREATE (:Book{isbn=row.isbn, title=row.title, etc})
```

In my experience, this will return a error if a duplicate is found, which stops the query.

ii) Merging the nodes with all their properties.

```
USING PERIODIC COMMIT 250
LOAD CSV WITH HEADERS FROM "file:C:\\path\\file.tsv" AS row FIELDTERMINATOR'\t'
MERGE (:Book{isbn=row.isbn, title=row.title, etc})
```

I have tried loading my set in this manner, but after letting the process run for over 36 hours and coming to a grinding halt, I figured there should be a better alternative, as ~200K of my eventual ~750K nodes were loaded.

iii) Merging nodes based on one property, and setting the rest after that.

```
USING PERIODIC COMMIT 250
LOAD CSV WITH HEADERS FROM "file:C:\\path\\file.tsv" AS row FIELDTERMINATOR'\t'
MERGE (b:Book{isbn=row.isbn})
ON CREATE SET b.title = row.title
ON CREATE SET b.author = row.author
etc
```

I am running a test now (~20K nodes) to see if switching from method ii to iii will improve execution time, as a smaller sample gave conflicting results. Are there methods which I am overseeing and could improve execution time? If I am not mistaken, the batch inserter only works for the CREATE command, and not the MERGE command.

I have permitted Neo4j to use 4GB of RAM, and judging from my task manager this is enough (uses just over 3GB).

merge neo4j bigdata nodes graph-databases

share improve this question

asked Mar 25 at 13:55
Michiel van Zummeren
13 ● 5

AFAIK, method iii) will make use of the index to find the node, while method ii) will not (because of matching multiple properties). Have you also tried increasing the number of ops in each transaction to 1000 for example? – albertoperdomo Mar 25 at 16:32

POSTGRES OPEN
2015
The premier open source postgresql conference
September 16 - 18 2015 Dallas TX
Connect • Learn • Hack

Related

1	Neo4j how can I merge two nodes in Java?
0	Importing the datasets from the Book “Graph databases”
1	Neo4j's Java Algorithm binding does not work with big dataset but Cypher does
8	What is the status on Neo4j's horizontal scalability project Rassilon?
0	Intersecting 2 big datasets
0	Use Apache Giraph as Neo4j with Big Amount of Data
0	Creating relationships in neo4j

I lowered it initially, because I was thinking Neo4j wasn't handling the loading well when I had set it at 1000. After finding the real problem with that setup, I haven't raised it back to 1000, so I'll do that again – Michiel van Zummeren Mar 26 at 13:56

add a comment

2 Answers

active oldest votes

1

✓

Method iii) should be the fastest solution since you `MERGE` against a single property. Do you create the uniqueness constraint before you do the `MERGE`? Without an index (constraint or normal index), the process will take a long time with a growing number of nodes.

```
CREATE CONSTRAINT ON (book:Book) ASSERT book.isbn IS UNIQUE
```

Followed by:

```
USING PERIODIC COMMIT 20000
LOAD CSV WITH HEADERS FROM "file:C:\\path\\file.tsv" AS row FIELDTERMINATOR'\t'
MERGE (b:Book{isbn=row.isbn})
ON CREATE SET b.title = row.title
ON CREATE SET b.author = row.author
```

This should work, you can increase the `PERIODIC COMMIT`.

I can add a few hundred thousand nodes within minutes this way.

share improve this answer

answered Mar 25 at 17:47

Martin Preusse
2,731 ● 4 ● 25 ● 42

Awesome! This combination improves the execution time greatly (16 seconds vs 30 min). Only problem is that is doesn't appear to load all the data from the file. It loads 20000 nodes, where method ii) loaded 20506. Any idea how this is caused? EDIT: Combining the constraint with method ii returns errors regarding the unique constraint so that's not an option – Michiel van Zummeren Mar 26 at 14:09

If you create the uniqueness constraint on `isbn`, both methods should create the same number of nodes (even though ii) merges on multiple properties). Try to reduce to `PERIODIC COMMIT 1000` for testing. How many unique `isbn` do you have in the test data set? – Martin Preusse Mar 26 at 14:16

Apparently, the 20000 is correct, as adding a dummy line to the file does increase the amount of nodes to 20001. The 20506 then must be caused by inconsistency in the data – Michiel van Zummeren Mar 26 at 14:19

I haven't extracted the raw data myself. It should be 20000, but because of more inconsistencies in the data i had received, I was doubting this was correct. – Michiel van Zummeren Mar 26 at 14:20

Thanks a bunch! The >1GB file is now loaded in roughly half an hour, much better then my first attempts – Michiel van Zummeren Mar 26 at 15:22

add a comment

0

In general, make sure you have indexes in place. Merge a node first on the basis of the properties that are indexed (to exploit fast lookup) and then modify that node's properties as needed with `SET`.

Beyond that, both of your approaches are going through the transaction layer. If you need to jam a lot of data into the DB really quickly, you probably don't want to use transactions to do that, because they're giving you functionality you might not need, and they require overhead that's slowing you down. So a larger solution would be to not insert data with `LOAD CSV` but go another route entirely.

If you're using the 2.2 series of neo4j, you can go for thebatch inserter via java, or the neo4j-import tool sadly not available prior to 2.2. What they both have in common is that they don't use transactions.

Finally, either way you go you should readMichael Hunger's article on importing data into neo4j as it provides a good conceptual discussion of what's happening, and why you need to skip transactions if you're going to load big huge piles of data into neo4j.

share improve this answer

answered Mar 25 at 17:47

FrobberOfBits
8,397 ● 7 ● 29

Thanks for the explanation on the indices and transaction layer. I'll have a look into the tools somewhere next week, as further improvement on execution times is more then welcome. First got to finish up a proof of

14 Are neo4j's cypher transactions broken?

1 Should I use Neo4j's Import Tool or Load Command to Insert Several Million Rows?

0 Merging nodes with the same data using Neo4j

Hot Network Questions

- Password strength of random words and brainflayer
- Is there a single-word antoym for "Opportunity"?
- An "outside the box" sequence
- Why are carpenter's pencils flat?
- If a stock doesn't pay dividends, then why is the stock worth anything?
- Legacy application unreachable code
- How to ask dumb questions
- Straight line in TikZ with 'continued' marking in the middle
- How do I know when my sauce is reduced enough?
- Scraping High-Res image tiles from the MoMA website
- 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?
- Important formulas in Combinatorics
- Where did the nickname of 'Bones' for McCoy come from?
- Does developing on ARM Processors from different vendors differ much?
- Which Star Trek character appeared on screen with the most different ranks?
- Aerofoil that gives reasonably good lift for both flow directions: forward and backward
- Antonym of Overlap?
- For loop formatting
- The Programming Language Quiz
- A short, brutal riddle
- Is it 'Batmen' or 'Batmans'?
- Lunch with the CEO
- Religion After the Discovery of a Multiverse

week, as further improvement on execution times is more then welcome. I first got to finish up a proof of concept in a timely fashion, so I prefer to work with the tools I'm familiar with so far. – Michiel van Zummeren Mar 26 at 14:36

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 merge neo4j bigdata nodes graph-databases 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