neo4j-import and node_auto_indexing

5 posts by 3 authors 8+1

Jean Reptile   Apr 3

Hello,

For a project, I need to import 5 million nodes and 15 millions relations.
I tried to import by batch but it was very slow, so I used the new tool 'Neo4j-import' from Neo4j 2.2. We generate some specifics .csv and use the 'neo4j-import'. It is very fast, the whole database is created in 1mn30.
But the problem is that I need to do a regex query on one property (find a movie with only the beginning of his name). And the average response time is between 2.5 and 4 seconds, which is huge.

I read that with Lucene query it would be much more efficient. But with Neo4-import, nodes are created without the node_auto_indexing.

Is there a way to use Neo4j-import and have node_auto_indexingin order to use the Lucene query ?

Thanks a lot

Michael Hunger   Apr 4

It's a bit tricky.

I think the fastest way to achieve that is to use the neo4j batch-inserter APIs, (which you might want to use for this import of your store anyway as it is not that large), I think you can even use mygithub.com/jexp/batch-import tool for populating the auto-index.

If you want to keep your store and just add the auto-index entries:

- open the store,
- create a lucene-batch-inserter-impl configured for fulltext and named node_auto_index
- walk over all node-id's retrieve the property and add it to the index.

Michael

dav prz   Apr 10

Hey Michael,

Congrats the new import tool is blazing fast with very nice feedback.

Howevr I also need to populate the index
wihtout taking a look at the neo4j-import tool code:
would it be difficult to add this functionality? I would gladly contribute if that's the case.

Thanks

---

dav prz   Apr 10

On a side note:

Would it work for us(Jean and I) to use: https://github.com/jexp/neo4j-shell-tools ?

Michael Hunger   Apr 10

Other recipients: mat...@neotechnology.com

You can populate the index just by using the shell and executing the appropriate statements

create index on :Label(prop);
..

create constraint on (n:Label) assert n.prop is unique;
...

// and at the end do
schema await

and let it run.

We'll add functionality for that later to the index tool.