Removing and readding a relationship to a RelationshipIndex in a single transaction corrupts later queries using the node as a starting point

New issue

Open **JohnButterworth** opened this issue Mar 26, 2015 · 3 comments

JohnButterworth commented Mar 26, 2015

I am using Neo4j version 2.1.7 embedded in my Java application using JRE 1.8.0_32

I've written a short test that seems to indicate that once a Relationship is removed and readded to a RelationshipIndex queries against that index that specify the relationship's starting node as a starting node in the query no longer return the correct results.

I would appreciate if anyone can shed some light on this or if I have written the code incorrectly in the test.

Thank you.

Here is the output of the test:

```
Update 1: Create 2 nodes and index their relationship
Created a left node with nickname 1
Created a right node with nickname 2
Created a relationship from left node 1 to right node 2
Indexed the relationship from 1 to 2 with timestamp 1001
The relationship was returned by the query when the start node was NOT specified
The relationship was returned by the query when the start node was specified

Update 2: Reindex the relationship
Unindexed the relationship from 1 to 2
Indexed the relationship from 1 to 2 with timestamp 1002
The relationship was returned by the query when the start node was NOT specified
The relationship was NOT returned by the query when the start node was specified
```

Here is the code for the test:

```
package sample;

import org.neo4j.graphdb.Direction;
import org.neo4j.graphdb.DynamicLabel;
import org.neo4j.graphdb.GraphDatabaseService;
import org.neo4j.graphdb.Label;
import org.neo4j.graphdb.Node;
import org.neo4j.graphdb.Relationship;
import org.neo4j.graphdb.RelationshipType;
import org.neo4j.graphdb.ResourceIterable;
import org.neo4j.graphdb.Transaction;
import org.neo4j.graphdb.index.IndexManager;
import org.neo4j.graphdb.index.RelationshipIndex;
import org.neo4j.graphdb.schema.ConstraintDefinition;
import org.neo4j.graphdb.schema.IndexDefinition;
import org.neo4j.index.lucene.QueryContext;
import org.neo4j.index.lucene.ValueContext;
import org.neo4j.test.TestGraphDatabaseFactory;
import org.neo4j.tooling.GlobalGraphOperations;

public class SampleApp {
    GraphDatabaseService graphDatabaseService;
    final Label leftLabel = DynamicLabel.label("left");
    final Label rightLabel = DynamicLabel.label("right");
    final String nicknameProperty = "nickname";
    final String relationshipIndexProperty = "timestamp";
    RelationshipIndex leftToRightIndex;
    long currentTime = 1000;

    static enum SampleRelationshipType implements RelationshipType {
```

Labels

None yet

Milestone

No milestone

Assignee

No one assigned

2 participants

```java
    static enum SampleRelationshipType implements RelationshipType {
        LEFT_TO_RIGHT
    }

    void createANode(Label label, int nickname) {
        Node node = graphDatabaseService.createNode(label);
        node.setProperty(nicknameProperty, nickname);
        System.out.println("Created a " + label.toString()
                + " node with nickname " + nickname + "");
    }

    Node findNodeByLabelAndProperty(Label label, String propertyName, int value) {
        final ResourceIterable<Node> results = graphDatabaseService
                .findNodesByLabelAndProperty(label, propertyName, value);
        return results.iterator().hasNext() ? results.iterator().next() : null;
    }

    Relationship createRelationship(int leftNickname, int rightNickname) {
        final Node leftNode = findNodeByLabelAndProperty(leftLabel,
                nicknameProperty, leftNickname);
        final Node rightNode = findNodeByLabelAndProperty(rightLabel,
                nicknameProperty, rightNickname);

        Relationship relationship = leftNode.createRelationshipTo(rightNode,
                SampleRelationshipType.LEFT_TO_RIGHT);
        System.out.println("Created a relationship from " + leftLabel + " node "
                + leftNickname + " to " + rightLabel + " node "
                + rightNickname + "");
        return relationship;
    }

    Relationship getRelationship(int leftNickname, int rightNickname) {
        final Node leftNode = findNodeByLabelAndProperty(leftLabel,
                nicknameProperty, leftNickname);
        final Node rightNode = findNodeByLabelAndProperty(rightLabel,
                nicknameProperty, rightNickname);

        for (final Relationship relationship : leftNode.getRelationships(
                Direction.OUTGOING, SampleRelationshipType.LEFT_TO_RIGHT)) {
            if (relationship.getEndNode().getId() == rightNode.getId()) {
                return relationship;
            }
        }

        return null;
    }

    void indexRelationship(Relationship relationship) {
        ++currentTime;
        leftToRightIndex.add(relationship, relationshipIndexProperty,
                new ValueContext(currentTime).indexNumeric());

        System.out.println("Indexed the relationship from "
                + relationship.getStartNode().getProperty(nicknameProperty)
                + " to "
                + relationship.getEndNode().getProperty(nicknameProperty)
                + " with timestamp " + currentTime);
    }

    void unindexRelationship(Relationship relationship) {
        leftToRightIndex.remove(relationship);

        System.out
                .println("Unindexed the relationship from "
                        + relationship.getStartNode().getProperty(
                                nicknameProperty)
                        + " to "
                        + relationship.getEndNode().getProperty(
                                nicknameProperty));
    }

    void createEmptyDb() {
        graphDatabaseService = new TestGraphDatabaseFactory()
                .newImpermanentDatabase();

        try (Transaction tx = graphDatabaseService.beginTx()) {
            for (ConstraintDefinition constraint : graphDatabaseService
                    .schema().getConstraints()) {
                constraint.drop();
            }

            for (IndexDefinition index : graphDatabaseService.schema()
                    .getIndexes()) {
                index.drop();
            }

            for (Relationship relationship : GlobalGraphOperations.at(
            for (Relationship relationship : GlobalGraphOperations.at(
                    graphDatabaseService).getAllRelationships()) {
                relationship.delete();
            }

            for (Node node : GlobalGraphOperations.at(graphDatabaseService)
                    .getAllNodes()) {
                node.delete();
            }

            graphDatabaseService.schema().indexFor(leftLabel)
                    .on(nicknameProperty).create();
            graphDatabaseService.schema().indexFor(rightLabel)
                    .on(nicknameProperty).create();
            final IndexManager indexManager = graphDatabaseService.index();
            leftToRightIndex = indexManager.forRelationships("LEFT_TO_RIGHT");
            tx.success();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    void createAndIndexRelationship(int leftNickname, int rightNickname) {
        try (Transaction tx = graphDatabaseService.beginTx()) {
            createANode(leftLabel, leftNickname);
            createANode(rightLabel, rightNickname);
            Relationship relationship = createRelationship(leftNickname,
                    rightNickname);
            indexRelationship(relationship);
            tx.success();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    void reindexRelationship(int leftNickname, int rightNickname) {
        try (Transaction tx = graphDatabaseService.beginTx()) {
            Relationship relationship = getRelationship(leftNickname,
                    rightNickname);
            unindexRelationship(relationship);
            indexRelationship(relationship);
            tx.success();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    void verifyReturnedByQuery(int leftNickname, int rightNickname, boolean specifyStartNo
        try (Transaction tx = graphDatabaseService.beginTx()) {
            final Node leftNode = findNodeByLabelAndProperty(leftLabel,
                    nicknameProperty, leftNickname);
            final Node rightNode = findNodeByLabelAndProperty(rightLabel,
                    nicknameProperty, rightNickname);

            boolean theRelationshipWasFound = false;
            for (final Relationship relationship : leftToRightIndex.query(
                    QueryContext.numericRange(relationshipIndexProperty,
                            Long.MIN_VALUE, Long.MAX_VALUE).sortNumeric(
                            relationshipIndexProperty, true), specifyStartNode ? leftNode
                if (relationship.getStartNode().getId() == leftNode.getId() &&
                    relationship.getEndNode().getId() == rightNode.getId()) {
                    theRelationshipWasFound = true;
                    break;
                }
            }

            System.out.println("The relationship was " + (theRelationshipWasFound ? "" : "
            tx.success();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    void go() {
        createEmptyDb();
        int leftNickname = 1, rightNickname = 2;

        System.out.println("\nUpdate 1: Create 2 nodes and index their relationship");
        createAndIndexRelationship(leftNickname, rightNickname);
        verifyReturnedByQuery(leftNickname, rightNickname, false);
        verifyReturnedByQuery(leftNickname, rightNickname, true);

        System.out.println("\nUpdate 2: Reindex the relationship");
        reindexRelationship(leftNickname, rightNickname);
        verifyReturnedByQuery(leftNickname, rightNickname, false);
        verifyReturnedByQuery(leftNickname, rightNickname, true);
    }
    }

    public static void main(String[] args) {
        new SampleApp().go();
        System.exit(0);
    }
}
```

jexp commented Mar 26, 2015 Collaborator

Thanks a lot for your report, I reported it to the engineering to have a look at it.

JohnButterworth commented Apr 4, 2015

Thank you. Is there any update on this?

I think I may have found another RelationshipIndex issue as well. I will post more as I discover more.

jexp commented Apr 4, 2015 Collaborator

Not yet, I'll let you know when there is an update.