stackoverfloooooooow

Questions   Tags   Users   Badges   Unanswered   Ask Question

GC overhead limit exceeded with Spring Data Neo4j

Problem: I am trying to import the DBLP database (from the publicly available XML file) into a Neo4j instance with a Spring application but after the application had imported about 39 000 publications into the Neo4j database I got a "GC overhead limit exceeded" message.

0

Note: I have thoroughly looked for answers at stackoverflow.com but I have not been able to solve the problem even though there are many good answers here.

Here is the entry point Application.java:

```java
package hello;

import java.io.File;

import org.neo4j.graphdb.GraphDatabaseService;
import org.neo4j.graphdb.Transaction;
import org.neo4j.graphdb.factory.GraphDatabaseFactory;
import org.neo4j.kernel.impl.util.FileUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.neo4j.config.EnableNeo4jRepositories;
import org.springframework.data.neo4j.config.Neo4jConfiguration;
import org.springframework.data.neo4j.core.GraphDatabase;

import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

import javax.xml.parsers.ParserConfigurationException;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.xml.sax.SAXException;
import org.apache.xerces.util.SecurityManager;

/**
```

Related

Hot Network Questions

```
 *
 * Inspiration: http://spring.io/guides/gs/accessing-data-neo4j/
 *
 */
@SpringBootApplication
public class Application implements CommandLineRunner {
```

Here is the SAX handler that Application.java makes use of:

```java
package hello;

import java.util.Stack;

import org.springframework.beans.factory.annotation.Autowired;
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

public class SaxHandler extends DefaultHandler {

    private Stack<String> elementStack = new Stack<String>();
    private Stack<Publication> objectStack = new Stack<Publication>();
    private String publicationType = null;
    private PublicationRepository publicationRepository = null;
    private Publication publication = null;
    private Author author = null;
    private String currentElement = null;
    private String value = null;

    private static int counter = 0;

    @Autowired
    public SaxHandler(PublicationRepository publicationRepository) {
        this.publicationRepository = publicationRepository;
    }

    public void startElement(String uri, String localName, String qName, Attribute

        elementStack.push(qName);

        publication = new Publication();

        if ("article".equals(qName)) {
            publication.setType("article");
        } else if ("inproceedings".equals(qName)) {
            publication.setType("inproceedings");
```

Here is PublicationRepository.java:

```java
package hello;

import org.springframework.data.repository.CrudRepository;

public interface PublicationRepository extends CrudRepository<Publication, String> {

    Publication findByTitle(String title);

}
```

Here is Author.java, one of the domain models:

```java
package hello;

import org.springframework.data.neo4j.annotation.GraphId;
import org.springframework.data.neo4j.annotation.NodeEntity;

@NodeEntity
public class Author {

    @GraphId
    private Long id;

    private String name;

    public Author() {

    }

    public Author(String name) {

    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;

        if (obj == null)
            return false;

        if (this.getClass() != obj.getClass())
            return false;

        Author other = (Author) obj;

        if (this.id != null && this.name != null && other.id != null && other.name
            if (this.id.equals(other.id) && this.name.equals(other.name))
                return true;
```

Here is Publication.java, the second domain model:

```java
package hello;

import java.io.Serializable;
import java.util.HashSet;
import java.util.Set;

import org.neo4j.graphdb.Direction;
import org.springframework.data.neo4j.annotation.GraphId;
import org.springframework.data.neo4j.annotation.NodeEntity;
import org.springframework.data.neo4j.annotation.RelatedTo;

@NodeEntity
public class Publication implements Serializable {

    private static final long serialVersionUID = -6393545300391560520L;

    @GraphId
    Long nodeId;

    private String type = "";
    private String key = "";
    private String mdate = "";
    private String publtype = "";
    private String reviewid = "";
    private String rating = "";
```

```java
    @RelatedTo(type = "WROTE", direction = Direction.INCOMING)
    private Set<Author> authors = new HashSet<Author>();
    private String editor = "";
    private String title = "";
    private String booktitle = "";
    private String pages = "";
    private String year = "";
    private String address = "";
    private String journal = "";
    private String volume = "";
    private String number = "";
    private String month = "";
```

The application uses Maven as build automation tool and here is pom.xml:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project    xmlns="http://maven.apache.org/POM/4.0.0"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apa

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.dblp</groupId>
    <artifactId>graphdbcreator</artifactId>
    <version>0.1.0</version>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.2.2.RELEASE</version>
    </parent>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-tx</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.data</groupId>
            <artifactId>spring-data-neo4j</artifactId>
        </dependency>
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-validator</artifactId>
```

Here is the central part of the error message:

```
java.lang.OutOfMemoryError: GC overhead limit exceeded
    at org.neo4j.collection.primitive.PrimitiveIntCollections.toPrimitiveIterator(Pr
    at org.neo4j.kernel.impl.api.StateHandlingStatementOperations.nodeGetLabels(Stat
    at org.neo4j.kernel.impl.api.ConstraintEnforcingEntityOperations.nodeSetProperty
    at org.neo4j.kernel.impl.api.LockingStatementOperations.nodeSetProperty(LockingS
    at org.neo4j.kernel.impl.api.OperationsFacade.nodeSetProperty(OperationsFacade.j
    at org.neo4j.kernel.impl.core.NodeProxy.setProperty(NodeProxy.java:254)
    at org.springframework.data.neo4j.fieldaccess.PropertyFieldAccessorFactory$Prope
    at org.springframework.data.neo4j.fieldaccess.DefaultEntityState.setValue(Defaul
    at org.springframework.data.neo4j.support.mapping.SourceStateTransmitter.setEnti
    at org.springframework.data.neo4j.support.mapping.SourceStateTransmitter.access$
    at org.springframework.data.neo4j.support.mapping.SourceStateTransmitter$3.doWit
    at org.springframework.data.neo4j.support.mapping.SourceStateTransmitter$3.doWit
    at org.springframework.data.mapping.model.BasicPersistentEntity.doWithProperties
    at org.springframework.data.neo4j.support.mapping.SourceStateTransmitter.copyPro
    at org.springframework.data.neo4j.support.mapping.Neo4jEntityConverterImpl.write
    at org.springframework.data.neo4j.support.mapping.Neo4jEntityPersister$CachedCon
    at org.springframework.data.neo4j.support.mapping.Neo4jEntityPersister.persist(N
    at org.springframework.data.neo4j.support.mapping.Neo4jEntityPersister.persist(N
    at org.springframework.data.neo4j.support.Neo4jTemplate.save(Neo4jTemplate.java:
    at org.springframework.data.neo4j.support.Neo4jTemplate.save(Neo4jTemplate.java:
    at org.springframework.data.neo4j.repository.AbstractGraphRepository.save(Abstra
    at sun.reflect.GeneratedMethodAccessor28.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.
    at java.lang.reflect.Method.invoke(Method.java:497)
    at org.springframework.data.repository.core.support.RepositoryFactorySupport$Que
    at org.springframework.data.repository.core.support.RepositoryFactorySupport$Que
    at org.springframework.data.repository.core.support.RepositoryFactorySupport$Que
    at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(Reflecti
    at org.springframework.data.repository.core.support.RepositoryFactorySupport$Def
    at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(Reflecti
    at org.springframework.transaction.interceptor.TransactionInterceptor$1.proceedW
    at org.springframework.transaction.interceptor.TransactionAspectSupport.invokeWi
```

java   maven   neo4j   garbage-collection   spring-data-neo4j

share improve this question

asked Mar 31 at 12:28

GCOverhead
3 ● 2

we had same problem, there is something wrong with neo when adding new nodes and relations. try to import using .csv file <-- that made us cry to create but eventually solved the problem – nafas Mar 31 at 12:32

add a comment

1 Answer

active   oldest   votes

▲

0

▼

✔

Your transaction is too large.

Try to create a service which only imports e.g. a batch of one thousand publications and annotated it with @Transactional (our use manual transactions within that method). Then call that method from your main class (get the service injected for spring-tx-handling) with batches of 1000 publications.

Alternatively, use this in your loop code

```
tx = db.beginTx();
while () {

// do stuff

if (++counter % 1000 == 0) {
   tx.success(); tx.close();
   tx = db.beginTx();
}
}
   tx.success(); tx.close();
```

share improve this answer

answered Mar 31 at 13:23

Michael Hunger
25.2k ● 2 ● 21 ● 41

add a comment

Your Answer

Sign up or log in

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.

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