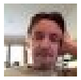
Stack overflow with large number of arguments

4 posts by 3 authors



Mike Holdsworth Apr 29

Neo4j 2.1.7 Java 8

When executing a cypher query with ~3000 arguments, passed as an in clause, the query fails. Reducing the java stack space (-Xss256) causes the error to occur with ~500 args

```
match ( e:Entity) where e.key in [ "", "some 3k","different args", "", .......]
```

Internals of Neo4j are doing some a recursive call and running out of stack space. Is this by design? Is there a better way to pass large arrays to use as args?

```
        at org.neo4j.cypher.internal.compiler.v2_1.bottomUp$BottomUpRewriter.apply(Rewritable.scala:137)
        at org.neo4j.cypher.internal.compiler.v2_1.bottomUp$BottomUpRewriter$$anonfun$1.
apply(Rewritable.scala:137)
        at scala.collection.Iterator$$anon$11.next(Iterator.scala:328)
        at scala.collection.Iterator$class.foreach(Iterator.scala:727)
        at scala.collection.AbstractIterator.foreach(Iterator.scala:1157)
        at scala.collection.generic.Growable$class.$plus$plus$eq(Growable.scala:48)
        at scala.collection.mutable.ListBuffer.$plus$plus$eq(ListBuffer.scala:176)
        at scala.collection.mutable.ListBuffer.$plus$plus$eq(ListBuffer.scala:45)
        at scala.collection.TraversableOnce$class.to(TraversableOnce.scala:273)
        at scala.collection.AbstractIterator.to(Iterator.scala:1157)
        at scala.collection.TraversableOnce$class.toList(TraversableOnce.scala:257)
        at scala.collection.AbstractIterator.toList(Iterator.scala:1157)
        at org.neo4j.cypher.internal.compiler.v2_1.bottomUp$BottomUpRewriter.apply(Rewritable.scala:137)
        at org.neo4j.cypher.internal.compiler.v2_1.bottomUp$BottomUpRewriter$$anonfun$1.
apply(Rewritable.scala:137)
        at scala.collection.Iterator$$anon$11.next(Iterator.scala:328)
        at scala.collection.Iterator$class.foreach(Iterator.scala:727)
        at scala.collection.AbstractIterator.foreach(Iterator.scala:1157)
        at scala.collection.generic.Growable$class.$plus$plus$eq(Growable.scala:48)
        at scala.collection.mutable.ListBuffer.$plus$plus$eq(ListBuffer.scala:176)
        at scala.collection.mutable.ListBuffer.$plus$plus$eq(ListBuffer.scala:45)
        at scala.collection.TraversableOnce$class.to(TraversableOnce.scala:273)
        at scala.collection.AbstractIterator.to(Iterator.scala:1157)
        at scala.collection.TraversableOnce$class.toList(TraversableOnce.scala:257)
        at scala.collection.AbstractIterator.toList(Iterator.scala:1157)
        at org.neo4j.cypher.internal.compiler.v2_1.bottomUp$BottomUpRewriter.apply(Rewritable.scala:137)
        at org.neo4j.cypher.internal.compiler.v2_1.bottomUp$BottomUpRewriter$$anonfun$1.
apply(Rewritable.scala:137)
```

Michael Hunger Apr 29

The parser was never built to handle kilobytes of query text :)
It will be terribly inefficient to parse and plan that query.

You should use parameters anyway and not literal values.

```
match ( e:Entity) where e.key in {keys}
```

Also I don't think this query will be very efficient as it has to go to the index 3000 times.

What is the actual use-case?

Michael

- show quoted text -
- show quoted text -
--
You received this message because you are subscribed to the Google Groups "Neo4j" group.
To unsubscribe from this group and stop receiving emails from it, send an email to neo4j+un...@googlegroups.com.
For more options, visit https://groups.google.com/d/optout.

Mike Holdsworth Apr 29

Our use case is to extract a subset of data of the graph from the millions of nodes for sample analysis. I would have thought performing lookups on a few thousand keys would be pretty straightforward.

The key values are already resolved, known and indexed in the graph and query performance is otherwise perfectly acceptable.

The issue seems to be with the recursive way in which Neo is working.

- show quoted text -
- show quoted text -
You received this message because you are subscribed to a topic in the Google Groups "Neo4j" group.
To unsubscribe from this topic, visit https://groups.google.com/d/topic/neo4j/EZq_lc-yBow/unsubscribe.

Michael Hunger Apr 29

Just use parameters.

Michael

- show quoted text -