Neo4j Performance Challenge - How to Improve?

asked 4 months ago

viewed 96 times

active 4 months ago

3

2

I've been wrangling with Neo4J for the last few weeks, trying to resolve some extremely challenging performance problems. At this point, I need some additional help because I can't determine how to move forward.

I have a graph with a total of approx 12.5 Million nodes and 64 Million relationships. The purpose of the graph is going to be analyzing suspicious financial behavior, so it is customers, accounts, transactions, etc.

Here is an example of the performance challenge:

- This query for total nodes takes 96,064ms to complete, which is extremely long.

```
neo4j-sh (?)$ MATCH (n) RETURN count(n);
+----------+
| count(n) |
+----------+
| 12519940 |
+----------+
1 row
96064 ms
```

- The query for total relationships takes 919,449ms to complete, which seems silly.

```
neo4j-sh (?)$ MATCH ()-[r]-() return count(r);
+----------+
| count(r) |
+----------+
| 64062508 |
+----------+
1 row
919449 ms
```

- I have 6.6M Transaction Nodes. When I attempt to search for transactions that have an amount above $8,000, the query takes 653,637ms also way too long.

```
neo4j-sh (?)$ MATCH (t:Transaction) WHERE t.amount > 8000.00 return count(t);
+----------+
| count(t) |
+----------+
| 10696    |
+----------+
1 row
653637 ms
```

Relevant Schema

```
 ON :Transaction(baseamount)    ONLINE
 ON :Transaction(type)          ONLINE
 ON :Transaction(amount)        ONLINE
 ON :Transaction(currency)      ONLINE
 ON :Transaction(basecurrency)  ONLINE
 ON :Transaction(transactionid) ONLINE (for uniqueness constraint)
```

Profile of Query:

```
neo4j-sh (?)$ PROFILE MATCH (t:Transaction) WHERE t.amount > 8000.00 return count(t)
+----------+
| count(t) |
+----------+
| 10696    |
+----------+
```

Blog

Why Stack Overflow is a Good Workplace for Women

Related

0 Improve Neo4j Cypher Performance On Lengthy Match

0 How to improve Neo4j 2.0 cypher/ExecutionResult performance under heavy load?

1 neo4j improving cypher query performace

0 How to return random records in Neo4j using Cypher?

0 how to improve Neo4J performance in creating edges?

1 Neo4j: how does relationships concentration influence performance?

1 neo4j cypher - Unable to execute query [400], 'StackOverflowError' exception

1 Neo4j write performance improvement

1 Neo4j - don't know how to improve cypher query

1 Neo4j - how to improve queries response time?

Hot Network Questions

The Programming Language Quiz

Combining several symmetric ciphers using XOR

Does touching the tent outer wall from the inside make it leak?

Why are carpenter's pencils flat?

An "outside the box" sequence

Manager sounds upset every time I inform him of a (minor) obstacle

When does the designer pay for a reprint

Lunch with the CEO

```
1 row

ColumnFilter
  |
  +EagerAggregation
    |
    +Filter
      |
      +NodeByLabel

+------------------+---------+----------+-------------+------------------------------
|         Operator |    Rows |   DbHits | Identifiers |
+------------------+---------+----------+-------------+------------------------------
|     ColumnFilter |       1 |        0 |             |                    keep colu
| EagerAggregation |       1 |        0 |             |
|           Filter |   10696 | 13216382 |             | Property(t,amount(62)) > {
|      NodeByLabel | 6608191 |  6608192 |        t, t |
+------------------+---------+----------+-------------+------------------------------
```

- I am running these in the neo4j shell.
- The performance challenges here are starting to create substantial doubt about whether I can even use Neo4J, and seem opposite of the potential the platform offers.
- I am fully admit that I may have misconfigured something (I'm relatively new to Neo4J), so guidance on what to fix or what to look at is much appreciated.

Here are details of my setup:

System: Linux, Ubuntu, 16GB RAM, 3.5 i5 Proc, 256GB SSD HD

CPU

```
$ cat /proc/cpuinfo
processor   : 0
vendor_id   : GenuineIntel
cpu family  : 6
model       : 60
model name  : Intel(R) Core(TM) i5-4690K CPU @ 3.50GHz
stepping    : 3
microcode   : 0x12
cpu MHz     : 4230.625
cache size  : 6144 KB
```

Memory

```
$ cat /proc/meminfo
MemTotal:       16115020 kB
MemFree:          224856 kB
MemAvailable:    8807160 kB
Buffers:          124356 kB
Cached:          8429964 kB
SwapCached:         8388 kB
```

Disk

```
$ df -h
Filesystem                  Size  Used Avail Use% Mounted on
/dev/mapper/data1--vg-root  219G   32G  177G  16% /
```

Neo4J.properties

```
neostore.nodestore.db.mapped_memory=200M
neostore.relationshipstore.db.mapped_memory=1G
neostore.relationshipgroupstore.db.mapped_memory=200M
neostore.propertystore.db.mapped_memory=500M
neostore.propertystore.db.strings.mapped_memory=500M
neostore.propertystore.db.arrays.mapped_memory=50M
neostore.propertystore.db.index.keys.mapped_memory=200M
relationship_auto_indexing=true
```

Neo4J-Wrapper.properties

- Did a Kamikaze ever impact the hull of a ship, as opposed to the deck?
- How can I grant a non-Super User group permission to force check-in of articles?
- Which Star Trek character appeared on screen with the most different ranks?
- Funding government by only printing more money
- Why don't decoupling/bypass capacitors need resistors to perform their function, like regular filters?
- Weird performance increase in simple benchmark
- Using Emergency Fund to Sell Upside-down Car
- Sign that word!
- Why did Holly decide to simply wait out the radioactivity?
- Which type of drill bit for which jobs?
- If a stock doesn't pay dividends, then why is the stock worth anything?
- Carrying a handgun in other countries with a US concealed carry permit
- How to show pagination "go to page" text box?
- As a professional photographer, how can one handle a wedding photographer being rude or demanding in a way that will compromise overall image quality?
- Why moving fan seems transparent?
- Aerofoil that gives reasonably good lift for both flow directions: forward and backward

```
wrapper.java.additional=-Dorg.neo4j.server.properties=conf/neo4j-server.properties
wrapper.java.additional=-Djava.util.logging.config.file=conf/logging.properties
wrapper.java.additional=-Dlog4j.configuration=file:conf/log4j.properties

#********************************************************************
# JVM Parameters
#********************************************************************

wrapper.java.additional=-XX:+UseConcMarkSweepGC
wrapper.java.additional=-XX:+CMSClassUnloadingEnabled
wrapper.java.additional=-XX:-OmitStackTraceInFastThrow

# Uncomment the following lines to enable garbage collection logging
wrapper.java.additional=-Xloggc:data/log/neo4j-gc.log
wrapper.java.additional=-XX:+PrintGCDetails
wrapper.java.additional=-XX:+PrintGCDateStamps
wrapper.java.additional=-XX:+PrintGCApplicationStoppedTime
wrapper.java.additional=-XX:+PrintPromotionFailure
wrapper.java.additional=-XX:+PrintTenuringDistribution

# Java Heap Size: by default the Java heap size is dynamically
# calculated based on available system resources.
# Uncomment these lines to set specific initial and maximum
# heap size in MB.
wrapper.java.initmemory=4096
wrapper.java.maxmemory=6144
```

Other:

- Changed the open file settings for Linux to 40k

- I am not running anything else on this machine, no X Windows, no other DB server. Here is a snippet of top while running a query:

  ```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
  15785 neo4j     20   0 12.192g 8.964g 2.475g S 100.2 58.3 227:50.98 java
  1 root      20   0   33464   2132   1140 S   0.0  0.0   0:02.36 init
  2 root      20   0       0      0      0 S   0.0  0.0   0:00.01 kthreadd
  ```

- The total file size in the graph.db directory is:

  ```
  data/graph.db$ du --max-depth=1 -h
  1.9G    ./schema
  36K ./index
  26G .
  ```

- Data loading was extremely hit or miss. Some merges would take less than 60 seconds (Even for ~200 to 300K inserts), while some merges would last for over 3 hours (11,898,514ms for a CSV file with 189,999 rows merging on one date)

- I get constant GC thread blocking:

  ```
  2015-03-27 14:56:26.347+0000 WARN  [o.n.k.EmbeddedGraphDatabase]: GC Monitor: Ap
  2015-03-27 14:56:39.011+0000 WARN  [o.n.k.EmbeddedGraphDatabase]: GC Monitor: Ap
  2015-03-27 14:56:57.533+0000 WARN  [o.n.k.EmbeddedGraphDatabase]: GC Monitor: Ap
  2015-03-27 14:57:17.345+0000 WARN  [o.n.k.EmbeddedGraphDatabase]: GC Monitor: Ap
  2015-03-27 14:57:29.955+0000 WARN  [o.n.k.EmbeddedGraphDatabase]: GC Monitor: Ap
  2015-03-27 14:58:14.311+0000 WARN  [o.n.k.EmbeddedGraphDatabase]: GC Monitor: Ap
  ```

Please let me know if I should add anything else that would be salient to the discussion

---

Update 1

Thank you very much for your help, I just moved so I was delayed in responding.

1. Size of Neostore Files:

   ```
   /data/graph.db$ ls -lah neostore.*
   -rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.id
   -rw-rw-r-- 1 neo4j neo4j  110 Apr  2 13:03 neostore.labeltokenstore.db
   -rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.labeltokenstore.db.id
   -rw-rw-r-- 1 neo4j neo4j  874 Apr  2 13:03 neostore.labeltokenstore.db.names
   -rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.labeltokenstore.db.names.id
   -rw-rw-r-- 1 neo4j neo4j 200M Apr  2 13:03 neostore.nodestore.db
   -rw-rw-r-- 1 neo4j neo4j   41 Apr  2 13:03 neostore.nodestore.db.id
   -rw-rw-r-- 1 neo4j neo4j   68 Apr  2 13:03 neostore.nodestore.db.labels
   -rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.nodestore.db.labels.id
   -rw-rw-r-- 1 neo4j neo4j 2.8G Apr  2 13:03 neostore.propertystore.db
   -rw-rw-r-- 1 neo4j neo4j  128 Apr  2 13:03 neostore.propertystore.db.arrays
   -rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.propertystore.db.arrays.id
   -rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.propertystore.db.id
   -rw-rw-r-- 1 neo4j neo4j  720 Apr  2 13:03 neostore.propertystore.db.index
   -rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.propertystore.db.index.id
   -rw-rw-r-- 1 neo4j neo4j 3.1K Apr  2 13:03 neostore.propertystore.db.index.keys
   ```

```
-rw-rw-r-- 1 neo4j neo4j 3.1K Apr  2 13:03 neostore.propertystore.db.index.keys
-rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.propertystore.db.index.keys.
-rw-rw-r-- 1 neo4j neo4j 1.7K Apr  2 13:03 neostore.propertystore.db.strings
-rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.propertystore.db.strings.id
-rw-rw-r-- 1 neo4j neo4j  47M Apr  2 13:03 neostore.relationshipgroupstore.db
-rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.relationshipgroupstore.db.id
-rw-rw-r-- 1 neo4j neo4j 1.1G Apr  2 13:03 neostore.relationshipstore.db
-rw-rw-r-- 1 neo4j neo4j 1.6M Apr  2 13:03 neostore.relationshipstore.db.id
-rw-rw-r-- 1 neo4j neo4j  165 Apr  2 13:03 neostore.relationshiptypestore.db
-rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.relationshiptypestore.db.id
-rw-rw-r-- 1 neo4j neo4j 1.3K Apr  2 13:03 neostore.relationshiptypestore.db.nam
-rw-rw-r-- 1 neo4j neo4j    9 Apr  2 13:03 neostore.relationshiptypestore.db.nam
-rw-rw-r-- 1 neo4j neo4j 3.5K Apr  2 13:03 neostore.schemastore.db
-rw-rw-r-- 1 neo4j neo4j   25 Apr  2 13:03 neostore.schemastore.db.id
```

2. I read that mapped memory settings are replaced by another cache, and I have commented out those settings.

3. Java Profiler

```
   JvmTop 0.8.0 alpha - 16:12:59,  amd64,  4 cpus, Linux 3.16.0-33, load avg 0.3
   http://code.google.com/p/jvmtop

   Profiling PID 4260:            org.neo4j.server.Bootstrapper

    68.67% (    14.01s) org.neo4j.kernel.impl.nioneo.store.StoreFileChannel.read
    18.73% (     3.82s) org.neo4j.kernel.impl.nioneo.store.StoreFailureException
     2.86% (     0.58s) org.neo4j.kernel.impl.cache.ReferenceCache.put()
     1.11% (     0.23s) org.neo4j.helpers.Counter.inc()
     0.87% (     0.18s) org.neo4j.kernel.impl.cache.ReferenceCache.get()
     0.65% (     0.13s) org.neo4j.cypher.internal.compiler.v2_1.parser.Literals$
     0.63% (     0.13s) org.parboiled.scala.package$.getCurrentRuleMethod()
     0.62% (     0.13s) scala.collection.mutable.OpenHashMap.<init>()
     0.62% (     0.13s) scala.collection.mutable.AbstractSeq.<init>()
     0.62% (     0.13s) org.neo4j.kernel.impl.cache.AutoLoadingCache.get()
     0.61% (     0.13s) scala.collection.TraversableLike$$anonfun$map$1.apply()
     0.61% (     0.12s) org.neo4j.kernel.impl.transaction.TxManager.assertTmOk()
     0.61% (     0.12s) org.neo4j.cypher.internal.compiler.v2_1.commands.EntityP
     0.61% (     0.12s) scala.collection.AbstractTraversable.<init>()
     0.61% (     0.12s) scala.collection.immutable.List.toStream()
     0.60% (     0.12s) org.neo4j.kernel.impl.nioneo.store.NodeStore.getRecord()
     0.57% (     0.12s) org.neo4j.kernel.impl.transaction.TxManager.getTransacti
     0.37% (     0.08s) org.parboiled.scala.Parser$class.rule()
     0.06% (     0.01s) scala.util.DynamicVariable.value()
```

neo4j cypher

share improve this question

edited Apr 2 at 20:15

asked Mar 27 at 15:03
BC Smith
16 ● 3

can you show the individual file sizes for `neostore.*` – Michael Hunger Mar 28 at 21:11

all your queries are full scans and are so mostly bound by your disk speed, is each of those the first query you run? Can you test the disk performance? And the disk scheduler which should be noop or deadline. – Michael Hunger Mar 28 at 21:13

The garbage collection also doesn't look good. Can you take a thread dump (kill -3 <pid>) and connect a profiler to your neo instance and report back `e.g. jvmtop.sh --profile <pid>` github.com/mneedham/jvmtop/tree/master/jvmtop – Michael Hunger Mar 28 at 21:14

Also your merge and create performance doesn't sound right. There seems tob e something really off with your machine. – Michael Hunger Mar 28 at 21:15

but in general full scans is not what a graph database is built for and according to your description also not what your use-case is about. – Michael Hunger Mar 28 at 21:16

add a comment

1 Answer

active oldest votes

2

Unfortunately the schema indexes (aka those created using `CREATE INDEX ON :Label(property)`) do not yet support larger than/smaller than conditions. Therefore Neo4j falls back to scan all nodes with the given label and filter on their properties. This is of course expensive.

I do see two different approaches to tackle this:

1) If your condition does always have a pre-defined maximum granularity e.g. 10s of USDs, you can build up an "amount-tree" similar to a time-tree (see http://graphaware.com/neo4j/2014/08/20/graphaware-neo4j-timetree.html).

2) if you don't know the granularity upfront the other option is to setup a manual or auto index for the amount property, see http://neo4j.com/docs/stable/indexing.html. The most easy thing is probably using auto index. In `neo4j.properties` set the following options:

```
node_auto_indexing=true
node_keys_indexable=amount
```

Note that this will not automatically add all existing transaction into that index, it just puts those in the index that have been written to since auto indexing is enabled.

You can do a explicit range query on the auto index using

```
MATCH t=node:node_auto_index("amount:[6000 TO 999999999]")
RETURN count(t)
```

share improve this answer

answered Mar 28 at 11:57
Stefan Armbruster
23k ● 2 ● 28 ● 51

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 neo4j cypher or ask your own question.

question feed

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