File leakage in neo4j spatial

5 posts by 3 authors

Josef Karthauser Mar 27

I'm importing a load of polygons into a neo4j spatial index (neo4j 2.1.6 / spatial 0.13-neo4j).

Each node is being added in it's own individual transaction.

The system appears to be leaking files whilst doing this:

```
| Error Error running script run-script  src/groovy/load_mastermap_
topographic_layer.groovy: java.lang.RuntimeException: java.io.FileNotFoundException:
/Users/joe/Documents/Wansdyke/Git/network-database/db/src/truedb/data/truedb.db/index/
lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__spatialNodeLookup__/
_2n9.prx (Too many open files in system) (Use --stacktrace to see the full trace)
| Error Exception in thread "Lucene Merge Thread #3085"
| Error org.apache.lucene.index.MergePolicy$MergeException: java.io.IOException:
directory '/Users/joe/Documents/Wansdyke/Git/network-database/
db/src/truedb/data/truedb.db/schema/index/lucene/51' exists and is a directory, but
cannot be listed: list() returned null
| Error         at org.apache.lucene.index.ConcurrentMergeScheduler.
handleMergeException(ConcurrentMergeScheduler.java:509)
| Error         at org.apache.lucene.index.ConcurrentMergeScheduler$MergeThread.run(
ConcurrentMergeScheduler.java:482)
| Error Caused by: java.io.IOException: directory ' tmp/thedb.db/
schema/index/lucene/51' exists and is a directory, but cannot be listed: list()
returned null
| Error         at org.apache.lucene.store.FSDirectory.listAll(FSDirectory.java:230)
| Error         at org.apache.lucene.store.FSDirectory.listAll(FSDirectory.java:241)
| Error         at org.apache.lucene.index.IndexFileDeleter.refresh(
IndexFileDeleter.java:335)
| Error         at org.apache.lucene.index.IndexWriter.merge(IndexWriter.java:3922)
| Error         at org.apache.lucene.index.ConcurrentMergeScheduler.doMerge(
ConcurrentMergeScheduler.java:388)
| Error         at org.apache.lucene.index.ConcurrentMergeScheduler$MergeThread.run(
ConcurrentMergeScheduler.java:456)
```

And, yes, there appear to be a load of open files:

```
joe$ lsof -p 89310 | grep thedb.db | wc -l
    6317
```

Loads of lucine index files...

```
thedb.db/schema/index/lucene/51/_4q3.cfs
thedb.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_2n5.cfs
thedb.db/schema/index/lucene/51/_4pp.cfs
thedb.db/schema/index/lucene/51/_4pw.cfs
thedb.db/schema/index/lucene/51/_4pq.cfs
thedb.db/schema/index/lucene/51/_4pr.cfs
thedb.db/schema/index/lucene/51/_4pd.cfs
thedb.db/schema/index/lucene/51/_4ph.cfs
thedb.db/schema/index/lucene/51/_4pv.cfs
thedb.db/schema/index/lucene/51/_4pf.cfs
thedb.db/schema/index/lucene/51/_4qb.cfs
thedb.db/schema/index/lucene/51/_4pg.cfs
thedb.db/schema/index/lucene/51/_4qo.cfs
```

```
thedb.db/schema/index/lucene/51/_4pt.cfs
thedb.db/schema/index/lucene/51/_4pj.cfs
thedb.db/schema/index/lucene/51/_4q5.cfs
thedb.db/schema/index/lucene/51/_4qk.cfs
thedb.db/schema/index/lucene/51/_4q0.cfs
thedb.db/schema/index/lucene/51/_4q8.cfs
thedb.db/schema/index/lucene/51/_4q6.cfs
thedb.db/schema/index/lucene/51/_4q7.cfs
thedb.db/schema/index/lucene/51/_4qa.cfs
thedb.db/schema/index/lucene/51/_4q9.cfs
thedb.db/schema/index/lucene/51/_4pu.cfs
thedb.db/schema/index/lucene/51/_4qg.cfs
thedb.db/schema/index/lucene/51/_4py.cfs
thedb.db/schema/index/lucene/51/_4ql.cfs
thedb.db/schema/index/lucene/51/_4px.cfs
thedb.db/schema/index/lucene/51/_4qs.cfs
thedb.db/schema/index/lucene/51/_4pz.cfs
thedb.db/schema/index/lucene/51/_4r2.cfs
thedb.db/schema/index/lucene/51/_4qe.cfs
thedb.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_2n2.cfs
thedb.db/schema/index/lucene/51/_4qn.cfs
thedb.db/schema/index/lucene/51/_4ra.cfs
thedb.db/schema/index/lucene/51/_4r3.cfs
[cut]
etc
```

Why would they be being leaked? Is this a known problem that been fixed in a later spatial?

Thanks,
Joe

Josef Karthauser Mar 27

> On 27 Mar 2015, at 19:39, Dr Josef Karthauser <joe.kar...@wansdyketele.com> wrote:
>
> I'm importing a load of polygons into a neo4j spatial index (neo4j 2.1.6 / spatial 0.13-neo4j).
>
> Each node is being added in it's own individual transaction.
>
> The system appears to be leaking files whilst doing this:
>
> ```
> [cut]
> ```
>
> Why would they be being leaked? Is this a known problem that been fixed in a later spatial?

Trying a simple test with just 13 spatially indexes nodes (polygons) I end up with the following files still open at the end of the process:

```
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3ji.fdt
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3ji.fdx
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3ji.frq
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3ji.nrm
```

```
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3ji.prx
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3ji.tis
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3ji.tis
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3jj.cfs
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3jk.cfs
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3jl.cfs
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3jm.cfs
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3jn.cfs
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3jo.cfs
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3jp.cfs
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3jq.cfs
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3jr.fdt
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/_3jr.fdx
test.db/index/lucene/node/topography__neo4j-spatial__LayerNodeIndex__internal__
spatialNodeLookup__/write.lock
test.db/schema/index/lucene/39/_6da.cfs
test.db/schema/index/lucene/39/_6db.cfs
test.db/schema/index/lucene/39/_6dc.cfs
test.db/schema/index/lucene/39/_6dd.cfs
test.db/schema/index/lucene/39/_6de.cfs
test.db/schema/index/lucene/39/_6df.cfs
test.db/schema/index/lucene/39/_6dg.cfs
test.db/schema/index/lucene/39/_6di.cfs
test.db/schema/index/lucene/39/_6dj.cfs
test.db/schema/index/lucene/39/_6dk.cfs
test.db/schema/index/lucene/39/_6dl.cfs
test.db/schema/index/lucene/39/_6dm.cfs
test.db/schema/index/lucene/39/_6dn.cfs
test.db/schema/index/lucene/39/_6dp.cfs
test.db/schema/index/lucene/39/_6dq.cfs
test.db/schema/index/lucene/39/_6dr.cfs
test.db/schema/index/lucene/39/_6ds.cfs
test.db/schema/index/lucene/39/_6dt.cfs
test.db/schema/index/lucene/39/_6du.cfs
test.db/schema/index/lucene/39/_6dw.cfs
test.db/schema/index/lucene/39/_6dx.cfs
test.db/schema/index/lucene/39/_6dy.cfs
test.db/schema/index/lucene/39/_6dz.cfs
test.db/schema/label/lucene/_6dq.cfs
test.db/schema/label/lucene/_6dr.cfs
test.db/schema/label/lucene/_6ds.cfs
```

That doesn't look right.

Joe

Michael Hunger Mar 27

Lucene needs many open files to work,

what's your ulimit? recommended is 40.000

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.

Josef Karthauser Mar 28

It doesn't appear to need lots of files for a read-only load, only when writing. The number of open files appears to be proportional to the number of nodes being saved, not the number of nodes in the database, and it retains the open files indefinitely. Makes me think it's a leaking file handle instead of a normal operational mode.

Joe
- show quoted text -

Michael Hunger Apr 4

Can you somehow reproduce this?

Michael
- show quoted text -