Metacat - Bug #5518

Track down the performance issue of metacat query.

10/26/2011 03:47 PM - Jing Tao

Status:	In Progress	**Start date:**	10/26/2011
Priority:	Normal	**Due date:**	
Assignee:	ben leinfelder	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	2.x.y	**Spent time:**	0.00 hour
Bugzilla-Id:	5518		

Description

Matt queried knb with the word "water". It took one minute and half. We need to figured out why it so slow.

History

#1 - 01/13/2012 04:01 PM - ben leinfelder

Once we get a development/staging copy of KNB I can look into this.

#2 - 01/18/2012 11:48 AM - ben leinfelder

the indexPaths are all present in metacat.properties.
These are what the main KNB page are searching with:
-organizationName
-surName
-givenName
-keyword
-para
-geographicDescription
-literalLayout
-abstract/para
-abstract
-idinfo/citation/citeinfo/title
-idinfo/citation/citeinfo/origin
-idinfo/keywords/theme/themekey
-placekey
-originator/individualName/surName
-originator/individualName/givenName
-creator/individualName/surName
-creator/individualName/givenName
-originator/organizationName
-creator/organizationName
-dataset/title
-keyword

Plus these duplicates:
-idinfo/citation/citeinfo/title
-idinfo/citation/citeinfo/origin
-idinfo/keywords/theme/themekey

#3 - 01/18/2012 12:01 PM - ben leinfelder

Running the same simple search ("water") on http://dev2.nceas.ucsb.edu/knb returns in a reasonable (tens of seconds) amount of time -- this using the "default" skin. 20117 records returned.

There does not appear to be caching of repeated search results on the KNB - subsequent searches can take as long as the initial one, but sometimes not.

I also triggered a nagios alert ("swap usage") when running these simple queries.

#4 - 01/19/2012 12:26 PM - Margaret O'Brien

Hi -
I routinely get a timeout (or equivalent?) when querying the LNO or KNB production metacats, on indexed fields. Queries speed up on retries, presumably because they are cached.

Obviously, I cant use the same query statements on both metacats, because they are not indexed the same. But you can have my scripts to play with as you please.

#5 - 01/23/2012 01:05 PM - ben leinfelder

With all production data transfered to our new staging server (http://knb-mn-stage-1.dataone.org/knb) a search for "water" was indeed cached after the initial search (~ 5 seconds to return and start transferring 20705 data packages).

An initial search for "earth" returned 16235 records in ~16 seconds.

So it looks like this server is doing pretty well.

#6 - 05/02/2012 03:47 PM - ben leinfelder

There seems to be renewed concern that this is not resolved.
During the Kepler/sensor workshop Matt reported query performance issues when ~25 people were concurrently searching for 'Insects' via the Kepler data search interface.
His search finally completed in ~30 minutes during which time the server load was very high and nagios sent many emails warning such.
While the search index should have been utilized, I can see that for truly concurrent searches on 'Insects' if all of the queries were started before the first one finished (when the query results are cached by Metacat) then ALL of the individual queries would run concurrently and never be able to make use of the cached results.

I can write a multi-thread test that launches long-running queries on a test Metacat and see what kind of results we get on different servers. I don't think running it against production KNB is the best thing, but there are DataONE servers that have a similar amount of KNB data on them that will be useful. They are, however, running Metacat 2.0 so won't be a true reflection of current behavior, only of what's to come with the next release.

#7 - 05/04/2012 02:43 PM - ben leinfelder

I wrote a small test that issues the same Metacat query to a given server with parallel threads (25 for this example). Using the same query that Kepler would issue, the query for "Insects" does take some time, but not a half hour.

These are in ms:
---
query: 0 took: 315384
query: 1 took: 322531
query: 2 took: 305548
query: 3 took: 308005
query: 4 took: 315377
query: 5 took: 315468
query: 6 took: 304509
query: 7 took: 315469
query: 8 took: 317498
query: 9 took: 315381
query: 10 took: 316663
query: 11 took: 305278
query: 12 took: 317429
query: 13 took: 306272
query: 14 took: 315357
query: 15 took: 318126
query: 16 took: 315469
query: 17 took: 315343
query: 18 took: 306331
query: 19 took: 304750
query: 20 took: 305546
query: 21 took: 315555
query: 22 took: 308002
query: 23 took: 322503
query: 24 took: 315374

#8 - 05/04/2012 03:26 PM - ben leinfelder

Running the same query (well, with "Insect" (not plural) so the results were not cached) with 50 threads makes this problem crop up.
The threads took about 22 minutes to return and we got a nagios warning about load ("check-mk").
Thinking about how Kepler performs the data query, I think each different metadata type (EML version) that it searches uses a different query thread.
So each Kepler client was probably launching 2-3 queries which puts the demo workshop at around 75 concurrent Metacat queries.
Question is...what to do about it now?

#9 - 05/04/2012 04:29 PM - Jing Tao

Yeah. Kepler issues a query for different metadata type because the ecogrid could handle only one return doctype. I believe it can handle multiple return type types in a single query. So we can merge the three queries into one in kepler.

#10 - 01/24/2013 11:45 AM - ben leinfelder

Moving this along with the 2.1 bugs because we are still supporting old Metacat query. We do want people to start using SOLR-based query when we make it available, however, and we may just have to ignore this bug in order to implement the newer faster query mechanism.

#11 - 03/27/2013 02:30 PM - Redmine Admin

Original Bugzilla ID was 5518

#12 - 07/09/2013 02:22 PM - ben leinfelder

- Target version changed from 2.1.0 to 2.x.y