Metacat - Bug #2154

Metacat Performace: Configurable path condition indices

07/13/2005 06:16 PM - Saurabh Garg

Status:	Resolved	**Start date:**	07/13/2005
Priority:	Normal	**Due date:**	
Assignee:	Saurabh Garg	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	1.6	**Spent time:**	0.00 hour
Bugzilla-Id:	2154		

Description

From Matt's email...

Configurable path condition indices -- this would allow admins to configure specific XML paths in metacat to be replicated to their own table in the DB (which would be dynamically created) to make searching fast for those paths. This is effective because clients (e.g., Morpho, web) tend to use only a limited set of where clause restrictions (e.g., on title, surname, keyword, ...). In a typical EML document with 5000 nodes, our current search has to hit many records when searching for a path like "dataset/title". Given that each document has only one or a few titles, a dedicated table for title that is indexed might only have 3500 records in the "dataset/title" table (compared to ~106 in xml_nodes), which would be a significantly faster query. Also, by creating temporary tables for all commonly searched fields, we would possibly avoid searchcing the xml_nodes table altogether, and therefore avoid the whole recursive query issue. See Sid's point #2 below for more details. This has a lot of potential for speeding up structured queries (e.g., spatial), but xml_nodes would still be used for unstructured (anyfield) queries.

History

#1 - 07/13/2005 06:37 PM - Saurabh Garg

From my email:

Metacat can have a new index table for admin specified fields. This will be helpful as both xml_nodes and xml_index are huge tables. But most of the queries we run are on a few specific fields. Example: any search on NCEAS skin results in the following query:
((SELECT DISTINCT docid FROM xml_nodes WHERE UPPER LIKE '%NATIONAL CENTER FOR ECOLOGICAL ANALYSIS AND SYNTHESIS%' AND parentnodeid IN (SELECT nodeid FROM xml_index WHERE path LIKE 'organizationName'))) Instead, lets say we have a metacat where the user has specified in metacat.properties that organizationName should be indexed seperately. Then we can replace the above query with something like this:
SELECT DISTINCT docid FROM xml_indexed_path WHERE nodedata LIKE '%NATIONAL CENTER FOR ECOLOGICAL ANALYSIS AND SYNTHESIS%' AND path LIKE 'organizationName')
xml_indexed_path could be a table which has three simple fields - docid, nodedata and path where nodedata and path are indexed.
This is similar to adding nodedata column in xml_index table which I tried to do in release 1.5. But I had to remove the changes because 1) The performance gain was not that significant. 2) nodedata in xml_index was not an exact copy of nodedata from xml_nodes because of space constraints.
However a new index table should result in 1) significant performance gain as the query is one table and on a much smaller table. 2) nodedata in new table would be exact copy of nodedata in xml_nodes. 3) user can tune metacat for the queries paths which are important for him. e.g. organizationName, keywords, coverage information, terms used in advanced search etc
Number of documents in KNB database:
1600
Size of xml_index in current KNB database:
3877202

Size of xml_index in current KNB
database:
5675990
Size of new table assuming we index 100 values/document for 1600
documents: 160000
(I think, 100 is easily an assumption on the higher side as far as
values/document is concerned)

#2 - 07/19/2005 12:42 PM - Saurabh Garg

It will also speed up queries like the following which are used while creating
the doclist...

select xml_nodes.docid, xml_index.path, xml_nodes.nodedata,
xml_nodes.parentnodeid from xml_index, xml_nodes where
xml_index.nodeid=xml_nodes.parentnodeid and (xml_index.path
like 'originator/individualName/surName' or xml_index.path
like 'originator/individualName/givenName' or xml_index.path
like 'originator/organizationName' or xml_index.path
like 'creator/individualName/surName' or xml_index.path
like 'creator/organizationName' or xml_index.path like 'dataset/title' or
xml_index.path like 'keyword') AND xml_nodes.docid in
('tonelow13.1024','Sartwell.88','ArchivalTag.5') AND xml_nodes.nodetype = 'TEXT

#3 - 12/08/2005 10:11 AM - Saurabh Garg

Done. Closing the bug as it is working fine.

#4 - 03/27/2013 02:19 PM - Redmine Admin

Original Bugzilla ID was 2154