Metacat - Bug #25

nodeid should be generated in safe manner

06/19/2000 10:44 AM - Jivka Bojilova

Status:	Resolved	**Start date:**	06/19/2000
Priority:	Normal	**Due date:**	
Assignee:	Jivka Bojilova	**% Done:**	0%
Category:	metacat	**Estimated time:**	0.00 hour
Target version:	Sep2000	**Spent time:**	0.00 hour
Bugzilla-Id:	25		

Description

We're going to use sequence and db trigger before insert for nodeid generation,
because it is the fastest way.
But here comes the problem with getting the right generated number from the
sequence (select seq.currval ...) b' there time between the sequence generation
and selection
insert into xml_nodes ...
db trigger: put nextval from seq ...
select currval from the seq ...
If 2 users make a call to writeChildNodeToDb method and the second user
insert into xml_nodes before the first one get currval from the sequence
definitely the first user will get wrong number (the number that belongs to the
second user):
1 user: insert into xml_nodes (1 ...)
2 user: insert into xml_nodes (2 ...)
1 user: select seq.currval -> 2
2 user: select seq.currval -> 2
Approach 1/ every request to the servlet starts a new tread.
Our servlet creats a new DBWriter obj on every request.
Put insert into xml_nodes & select seq.curval in one method:
DBSAXNode.writeChildNodeToDB() and make it
static synchronized DBSAXNode.writeChildNodeToDB()
(should investigate)
Approach 2/ having single DBWriter obj for the servlet but everything inside
DBWriter should be synchronized and tread safe
(very complicated and unefficient)
Approach 3/ call to one DBWriter static method from the servlet.
This will avoid creating many obj instances for the DBWriter but will still
create instances of the other classes.
And still need static synchronized DBSAXNode.writeChildNodeToDB()

History

#1 - 06/19/2000 02:50 PM - Jivka Bojilova

DONE
Used different approach:
When next node to be inserted into xml_nodes table, get a next # from
sequence up in the Java code and use it for nodeid and return it to be used for
parentnodeid or roodnodeid values for subsequent nodes.
Requests are tread save and since we create new DBWriter obj insts on every
request and generate IDs inside them (not down in the db) then nodeid
generation is safe.
And we avoid using synchronization of treads.

#2 - 06/23/2000 04:26 PM - Jivka Bojilova

DONE
Used different approach:
When next node is to be inserted into xml_nodes table, get a next # from
sequence up in the Java code and use it for nodeid and return it to be used for
parentnodeid or roodnodeid values for subsequent nodes.
Requests are tread save and since we create new DBWriter obj insts on every
request and generate IDs inside them (not down in the db) then nodeid

generation is safe.
And we avoid using synchronization of treads.

#3 - 03/27/2013 02:13 PM - Redmine Admin

Original Bugzilla ID was 25