

Infrastructure - Bug #3065

cn.listObjects slicing broken in production

2012-07-13 08:26 - Rob Nahf

Status:	Closed	Start date:	2012-12-02
Priority:	Normal	Due date:	2013-01-05
Assignee:	Ben Leinfelder	% Done:	100%
Category:	Metacat	Estimated time:	0.00 hour
Target version:	Sprint-2012.50-Block.6.4		
Milestone:	CCI-1.1	Story Points:	
Product Version:	*		
Description			
When retrieving the full list of objects 'page at a time' using slicing (start and count parameters), I get duplicate entries, and by inference, am not getting all identifiers.			
For example, for one run against cn-orc-1, there are 101247 identifiers, but when put into a hash table, only 89635 unique keys exist. In another test run, the number of unique keys (identifiers) was 95760. The phenomenon was noticed Java, and confirmed in perl script running curl. When I pull the objectList in one gulp, there are 101247 unique identifiers.			
Most likely retrieval from the database table does not guarantee the same order on repeat calls.			
Related issues:			
Related to Infrastructure - Bug #3128: Production CN listObjects slow response		Closed	2012-08-13 2013-01-05
Related to Infrastructure - Task #3465: Improve Metacat listObject slicing		New	2012-12-07

History

#1 - 2012-07-13 08:55 - Rob Nahf

- Subject changed from *cn.listObjects slicing broken* to *cn.listObjects slicing broken in production*
- File *parseObjectList.pl* added
- File *parseORCresults.txt* added
- Category set to Metacat
- Assignee set to Ben Leinfelder

#2 - 2012-07-13 14:51 - Ben Leinfelder

- Status changed from New to In Progress

Now ordering the results from the db by identifier.

This is fine for a static system, but when objects are continually being added, there's really no guarantee that the slicing won't then include additional identifiers in the results and throw off the original paging counts.

This will be included in the Metacat 2.0.2 release (currently in RC testing)

#3 - 2012-07-13 16:54 - Rob Nahf

if the database table contained a sequence # field, then additional entries would be tacked onto the end of the list, and it would be possible for an iterating listObjects routine to pick up those new entries with the final iteration. The downside compared to the current fix is that clients are more likely to want to sort after retrieval.

#4 - 2012-07-19 22:02 - Ben Leinfelder

- Position deleted (3)

- Target version changed from Sprint-2012.27-Block.4.2 to Sprint-2012.29-Block.4.3
- Position set to 1

#5 - 2012-08-14 23:46 - Ben Leinfelder

We'd then need a way for all CNs to have the same sequence number for each pid since we never know which CN will actually return the next batch of objects.

Rob Nahf wrote:

if the database table contained a sequence # field, then additional entries would be tacked onto the end of the list, and it would be possible for an iterating listObjects routine to pick up those new entries with the final iteration. The downside compared to the current fix is that clients are more likely to want to sort after retrieval.

#6 - 2012-08-14 23:49 - Ben Leinfelder

Other than for diagnostic purposes, is CN.listObjects() ever used in a manner that requires exactly consistent paging? I can't think of another product or project that actually utilizes that service method. For the MN, sure, synchronization calls it all day long. But on the CN?

#7 - 2012-08-31 03:05 - Chris Jones

- Target version changed from Sprint-2012.29-Block.4.3 to Sprint-2012.37-Block.5.3

#8 - 2012-10-11 15:26 - Dave Vieglais

- Milestone changed from None to CCI-1.1
- Due date set to 2012-09-22
- Start date set to 2012-09-09
- translation missing: en.field_remaining_hours set to 0.0

Is this still an issue?

#9 - 2012-10-11 15:26 - Dave Vieglais

- Due date changed from 2012-09-22 to 2012-10-27
- Target version changed from Sprint-2012.37-Block.5.3 to Sprint-2012.41-Block.6.1

#10 - 2012-10-24 16:08 - Ben Leinfelder

- Status changed from In Progress to Testing

needs to be tested on the dev cns

#11 - 2012-10-24 18:07 - Ben Leinfelder

- Due date changed from 2012-10-27 to 2012-11-10
- Target version changed from Sprint-2012.41-Block.6.1 to Sprint-2012.44-Block.6.2

#12 - 2012-12-07 18:51 - Ben Leinfelder

Added DB-based slicing to Metacat 2.0.5 so that it is not done in memory.

I think the final thing needed for this is an artificial id on each row of SM so that the slice order is guaranteed even when new content is added (always

to the end of the list).

#13 - 2012-12-12 18:39 - Ben Leinfelder

- Target version changed from Sprint-2012.44-Block.6.2 to Sprint-2012.50-Block.6.4
- Due date changed from 2012-12-07 to 2013-01-05

#14 - 2013-01-08 18:38 - Ben Leinfelder

- Status changed from Testing to Closed

Given the limits of our current CN architecture, we have done all we can to improve the performance of listObject slicing. And for paged requests that occur in rapid succession on a perfectly in-synch unchanging system the current paging mechanism will work (sorted by guid). But we have a volatile, imperfect system of multiple CNs on our hands and cannot guarantee consistent listObject slicing.

Please see the new task [#3468](#) for a continuation of this issue.

Files

parseObjectList.pl	1.1 KB	2012-07-13	Rob Nahf
parseORCresults.txt	469 KB	2012-07-13	Rob Nahf