

Morpho - Bug #5823

Improve save-to-network performance

01/24/2013 11:37 PM - ben leinfelder

Status:	New	Start date:	01/24/2013
Priority:	Normal	Due date:	
Assignee:	Jing Tao	% Done:	0%
Category:	morpho - general	Estimated time:	0.00 hour
Target version:	2.1.0	Spent time:	0.00 hour
Bugzilla-Id:	5823		
Description			
Jing saved data packages with binary data entities from Morpho to a Metacat MN to test the performance of those save operations. There seems to be some overhead for this no matter how small the data file.			

File size	Total time (s)		
1.4 K	10		
395 k	10		
1.3 M	15		
5.8 M	26		
8.8 M	33		
50 M	115		

History

#1 - 01/24/2013 11:38 PM - ben leinfelder

I think in all cases we have the following calls:

- check if EML file exists
- generate new ID if so
- check if data file exists
- generate new ID if so
- submit EML + SystemMetadata bytes
- submit data + SystemMetadata bytes
- submit ORE +SystemMetadata bytes

For each call we do a lot of local work locating and setting up the client certificate and CA truststore for the SSL connection - each time starting from scratch. This can't come cheap and I wonder if there is room for improvement there. Perhaps the DataONE client can maintain a bit more state than it currently is doing. The CA truststore won't be changing from call to call, whereas the client certificate could be.

#2 - 01/25/2013 01:21 PM - ben leinfelder

With my NCEAS network connection I did a small and big data file:

big (50 MB) -- 18.5 s
small (119 bytes) -- 8.3 s

#3 - 02/20/2013 05:33 PM - Jing Tao

I did a test to save the almost same data packages (an eml document to describe the same data file) into the network and both locations.

It is interesting that the saving to the both location(10 seconds) took much less time than just saving to the network (14 seconds).

Here is some details:

It takes about 1 ~ 2 seconds to create a small size D1Object on the network.
It takes about 0.2 second to get a network-generated id.

The saving process saved 3 objects (a resource map, an eml document and a data file) and it took about 5 to 6 seconds.

The reason why the save-both action took the less time i guess is that it reads from the local copies to display the new data package.. But the save-network-only process reads from the network. If we can add the saved D1Objects into the cache in the d1_libclient_java module, it will improve the process.

#4 - 03/27/2013 02:31 PM - Redmine Admin

Original Bugzilla ID was 5823

