

## Morpho - Bug #640

### Problem trying to download very large data sets

10/18/2002 10:58 AM - Dan Higgins

<b>Status:</b>	Resolved	<b>Start date:</b>	10/18/2002
<b>Priority:</b>	Immediate	<b>Due date:</b>	
<b>Assignee:</b>	Dan Higgins	<b>% Done:</b>	0%
<b>Category:</b>	morpho - general	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.3	<b>Spent time:</b>	0.00 hour
<b>Bugzilla-Id:</b>	640		
<b>Description</b>			
Chris Jones reports that attempts to move very large data files from Metacat to Morpho often apparently 'hang' after several hundred kilobytes has been transferred.			

### History

#### #1 - 11/05/2002 12:33 PM - Dan Higgins

I created a new data package containing one of the PISCO large CODAR data files (~47MB). Submitting it to Metacat seemed to work fine (~10 secs or so from within the NCEAS local net). Examination of the directory where data is stored on ecoinfo indicates that the entire file was copied. Thus the upload of very large files seems to work OK.

Download, however, is where the problem seems to occur. I deleted the local copy and then attempted to Synchronize from Metacat. The process seems to start but then hangs (with the CPU at near 100%) after 3-25MB have been sent. (Estimate based on size of file in cache). I waited 15-20 minutes and no recovery seemed to occur.

Could this be a problem with MetaCat or Apache?

#### #2 - 11/05/2002 12:46 PM - Matt Jones

Maybe. Easy to test. Try downloading the data file or package from Metacat using a web browser rather than Morpho.

#### #3 - 11/05/2002 02:03 PM - Dan Higgins

Download through browser works OK! --- apparently a Morpho problem  
- Dan Higgins -

#### #4 - 11/06/2002 09:08 AM - Dan Higgins

This problem with large data sets is apparently linked to the alternate HTTPClient package being used in Morpho. If one removes that code to sets the protocol handler and just uses the default http handler from Sun for downloads, there is no problem downloading large data files! (Of course, there is a problem uploading large files!)

There appears to have been no updates to the HTTPClient code since May 2001, so there is no a newer version.

Also, same problems are seen with Java 1.3 and 1.4

#### #5 - 11/08/2002 02:24 PM - Dan Higgins

A test using Morpho 1.1 indicates that the downloading of a large test dataset (a single 45MB Pisco data set) works fine with the current HTTPClient, but the same dataset hangs while trying to download with Morpho 1.2. An investigation which examined the differences between the two versions showed that the sychonize code in the newer version runs inside a SwingWorker threan. If one removes that thread and simply runs the synch code 'in-line' the download works OK!!!!

It would thus appear that there is a thread problem with the HTTPClient code (since the same problem does not occur if one uses Sun's version). One 'fix' would be to simply not run the metacat downloads in a separate thread, but that leaves Morpho 'unresponsive' during downloads. This is not a big problem if

working within NCEAS but with a T1 connection from the outside, one might have to wait a number of minutes to download a 50 MB data file.

**#6 - 11/12/2002 10:38 AM - Dan Higgins**

Some further investigation seems to confirm that we have a 'thread' problem in the HTTPClient package. Downloading a large data package seemed to hang at different places during the download. It finally occurred to me that this might be due to the background timer that checks for network connections by 'pinging' metacat.

And, sure enough, if one disables the periodic checks for network availability, then the download of large data file works!!!!!!!!!!!!!!

So a quick fix (hack) is to simply check for network availability at start up and not continually poll during a session.

Of course it would be better to figure out where the threading problem is inside HTTPClient, but that may take some time. [Note that since the system works with Sun's http handler, I assume there is no inherent reason that we cannot have two threads connecting to metacat at the same time.]

This problem indicates that any attempt to have two threads talking to metacat at the same time will probably cause a problem. Morpho threading allows, for example, one to start a query while data is being downloaded. This probably will not work right while this threading bug exists in HTTPClient.

**#7 - 12/04/2002 04:37 PM - Dan Higgins**

added a flag in the Morpho class that keeps track of when a connection to metacat is busy. The doPing method here checks that flag. Synchronized 'getMetacatInputStream' method in Morpho class to avoid threading problems with HTTPClient. Also added flags in MetacatDataStore methods to turn off the 'doPing' method while any streams from metacat are open.

Thus, problem with downloading large data sets has been eliminated, although the root cause of threading problem has not been determined.

**#8 - 03/27/2013 02:14 PM - Redmine Admin**

Original Bugzilla ID was 640