

Morpho - Bug #2169

Table display in Morpho is slow when many columns involved

08/11/2005 04:07 PM - Callie Bowdish

Status:	Resolved	Start date:	08/11/2005
Priority:	Normal	Due date:	
Assignee:	Saurabh Garg	% Done:	0%
Category:	morpho - general	Estimated time:	0.00 hour
Target version:	Unspecified	Spent time:	0.00 hour
Bugzilla-Id:	2169		

Description

A table with 176 columns but only 45 records took so long to open that I thought Morpho had frozen. After 10 minutes or so it opened. The table I am working with is in a data project that has not been put on the network yet. It is part of the Data Package: "Data for the Habitat Conservation Planning for Endangered Species Study" table PQdata-final.txt

X-chat notes: you know what, it just opened up, took about 10 minutes. I could see the java working on it but I thought maybe it was stuck humm...

<sid> go to a faster machine?

<Callie> next time I'll time it. java was using about 50% processing power and 120 megs of ram

<sid> 50% probably because you have a dual proc machine

<Callie> is that a joke

<Callie> 3.19 GHz .99GB of RAM

just to help clarify, the 3.2 GHz machines have hyperthreading enabled, so 50% cpu usage is pretty much equal to 100% usage on a non-HT machine

<dan> I worked to support very long tables, but not very wide ones!

<matt_> ah

<matt_> well, it won't be uncommon

<Callie> yes this one had 176 tables all needing attributes... thanks nick I didn't realize that 50% CPU = 100 usage

<matt_> its not a good design

<matt_> but we'll have to deal

<matt_> callie -- can you enter a morpho bug to see if we can improve performance on wide tables

History

#1 - 08/12/2005 09:30 AM - Callie Bowdish

08/12/05 checked opening the table in Morpho again today and it took 10 minutes with the processor running at maximum. My task manager said Java2.exe was using about 49% CPU. Nick reports that "hyperthreading enabled, so 50% cpu usage is pretty much equal to 100% usage on a non-HT machine"

#2 - 08/29/2005 04:13 PM - Will Tyburczy

Increase in table load time seems to increase non-linearly as the number of columns increases. Some sample load times using different tables:

table1, 244 cols: 15 min 30 sec

table1, 122 cols: 2 min 10 sec

table2, 256 cols: 27 min 00 sec

table2, 128 cols: 3 min 50 sec

#3 - 08/30/2005 10:25 AM - Dan Higgins

A major cause of this problem is in the section of code that builds the header of the data table (i.e. column name, unit, etc). The problem is that AbstractDataPackage class is called to get that data and each column called a method to get an array of Attribute information for all columns. The result was that the attribute information was called ncolumns number of times and getting the data increased with ncolumns; so the amount of computation would increase with square of the number of columns.

A fix was made to the AbstractDataPackage class to save the last list of attribute nodes and used the saved data rather than regenerate it. This greatly speeds up the opening of tables with large number of columns. We need to check that this doesn't create problems (e.g. when a table is nmodified) before closing this bug.

Dan Higgins --- Aug 30, 2005

#4 - 09/01/2005 10:49 AM - Dan Higgins

Added some additional fixes to the AbstractDataPackage class to force a rebuild of the attributeArray after a column is added or deleted.

Marking this bug as 'fixed'. Tables now appear in a few seconds that previously required ~1 hr to display! (Dan Higgins)

#5 - 03/27/2013 02:19 PM - Redmine Admin

Original Bugzilla ID was 2169