

Kepler - Bug #4362

component search performance

09/01/2009 01:27 PM - Derik Barseghian

Status:	Resolved	Start date:	09/01/2009
Priority:	Normal	Due date:	
Assignee:	Aaron Aaron	% Done:	0%
Category:	interface	Estimated time:	0.00 hour
Target version:	wrp-modules-2.0.0	Spent time:	0.00 hour
Bugzilla-Id:	4362		

Description

For awhile now I've noticed the first search of the library has become much slower than it used to be, subsequent searches seemed to be fine, at least in the same window.

Just recently I've noticed the more Kepler windows I have open, the slower component search becomes. If I have 2 windows open when using WRP suite w/ my eclipse build on mac w/ 4gb ram, doing a search for "string" in the second window will generally crash Kepler with errors about Java heap space:

```
Exception in thread "AWT-EventQueue-0" java.lang.OutOfMemoryError: Java heap space
at java.util.LinkedList.addBefore(LinkedList.java:634)
at java.util.LinkedList.add(LinkedList.java:196)
at ptolemy.kernel.CompositeEntity.entityList(CompositeEntity.java:788)
at ptolemy.moml.EntityLibrary.entityList(EntityLibrary.java:357)
```

I've also seen this occasionally, though I'm not sure if it's related:

```
2009-09-01 13:04:36.530 java[50021:80f] Apple AWT Startup Exception : *** Collection <NSCFArray: 0x19e18d10> was mutated
while being enumerated.
```

```
2009-09-01 13:04:36.531 java[50021:80f] Apple AWT Restarting Native Event Thread
```

On my linux cmd line build, just using kepler suite, I also notice performance issues, though it takes more windows for them to become severe. Once I have 7 kepler windows open, search for "string" takes about 12 seconds.

History

#1 - 09/09/2009 05:16 PM - Aaron Aaron

Yeah seems like this could be improved quite a bit. The OntLibrarySearcher class does the heavy lifting. I added a time statement to SimpleLibrarySearcher class that shows the time it takes to do the search. Turn it on by uncommenting the log4j line for the SimpleLibrarySearcher class. There seems to be a lot of bad results returned from the search as well. Most likely the cause of the slowdown is that the library is twice as big now (with the folder items in there as well as the Ontology items). The algorithm used to do the search needs some looking at, it may not have scaled well to twice the amount of items in the library.

#2 - 10/06/2009 12:21 AM - Aaron Aaron

Sean and I discussed two possibilities for implementing fast component search today.

One was to use an SQL table that indexed the entire component library using Preorder Tree Traversal technique of storing hierarchical data in a relational database. I have much experience with this and know that the machinery to implement such a solution would be quite time consuming. However it would make our searches extremely fast and likely be useful in many other tasks.

Another option was to index the search terms (name, classname, ontologies, classes, etc.) for the components by their KeplerLSIDs in an SQL table. An SQL query could then be used to perform the matching of the provided string with the indexed search terms and return a list of KeplerLSIDs. Then a quick walk through the tree to match the LSIDs would finish the job. We're pretty sure the speed of the SQL query will be fast even though hsql does not support a fulltext index the same way MySQL and PostgreSQL do. Since we only expect to have a few thousand rows for the existing size of the component library the lack of fulltext indexing should not be a problem. The other slowing factor would be the KeplerLSID matching in the component library after the results have been retrieved from the database. To demonstrate that this is likely a very quick procedure I have implemented KeplerLSID matching in the Component Library (see bug 4303). You can try it yourself by right clicking on a component, view the LSID, copy it, and paste it into the search field.

#3 - 10/06/2009 05:06 PM - Aaron Aaron

I started to design the second option above this morning to figure out how difficult it would be to do. Turns out it was super easy, so I just went ahead and did it. If the performance is not good enough and we can think of a better solution it will be easy to delete and update it to whatever newer system we want. For now the Component Library search is pretty fast again. I will wait a little while before remerging the results into a single tree since the individual results makes it easy to see what is happening. Once we're satisfied with it I'll remerge the results into one tree.

#4 - 10/07/2009 11:04 AM - Aaron Aaron

I have done some timing analysis of the new component search implementation.

There are 3 stages of the new component search:

- 1) Searching the Indexed search terms in the SQL table named CACHE_SEARCH and finding all KeplerLSIDs that match
- 2) Using the results from 1) to walk through the component library model and find all of the instances that match the KeplerLSIDs and build the new model that is to be displayed to the user
- 3) Refreshing the graphical AnnotatedPTree with the new model that was generated in stage 2)

Stage 1) turns out to be very fast <10ms

Stage 2) is quite fast <54ms

Stage 3) is where the most time is spent, 87ms for 14 results but up to 5 seconds for 838 results

Stage 3 is currently negatively impacted since the results are individually displayed, this time should be improved substantially when the results are merged back into one tree. I will redo this analysis after remerging the tree. Also, 838 results causes a heap overflow exception and essentially crashes Kepler unless the -Xmx option is set fairly high (80M seems OK).

My conclusion is that this solution for searching components is going to work fine for now. But our AnnotatedPTree implementation is insufficient for large tree models.

Below is the output I got on my Macbook pro, these results can be recreated by uncommenting the ...ComponentLibraryTab=DEBUG line in the kepler module log4j file and uncommenting the two System.out lines in SimpleLibrarySearcher.search(String) (approx line 140)

```
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:291) Search term is 'sdf'
Index search generated 8 results in 13ms
Model search generated 14 results in 15ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:296) Library search completed in 53ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:299) Library updated in 87ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:300) Total search time was 140ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:291) Search term is 'data input'
Index search generated 30 results in 4ms
Model search generated 71 results in 12ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:296) Library search completed in 16ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:299) Library updated in 272ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:300) Total search time was 289ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:291) Search term is 'components'
Index search generated 355 results in 5ms
Model search generated 838 results in 54ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:296) Library search completed in 59ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:299) Library updated in 5031ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:300) Total search time was 5090ms
```

#5 - 10/08/2009 02:59 PM - Aaron Aaron

From meeting with Sean and Derik.

NamedOntClass.getName() does not always return the name that appears in the component library. Sean has a new accessor to get at the label of the OntClass instance stored in the NamedOntClass that is the value that is used in the component library. He is going to flatten his override into the core so we can use it when building the search index.

Derik cannot add a WorkflowRun to the search index now because it only supports adding an ActorMetadata object. Aaron will add a method to allow including a named object in the search index.

Derik needs a second search table and a second search class and search interface for doing searches on the workflow runs in the workflowrunmanager. The reason is that the existing search only works with components that have been saved to an archive. The workflow runs are stored in the provenance tables and do not always appear in the component library.

The new search allows for specifying what type of search term you would like to match (Name,Java classname,ontology name, ontology classname). We agreed that adding an advanced button that allows the user to configure what type of search terms they would like to match would be an important addition. With this ability the default search would be set to only match on the Names of components. The user would have to select a checkbox if they also wanted to match java classname, ontology name or ontology class names. Aaron will implement that.

#6 - 10/12/2009 04:55 PM - Aaron Aaron

I've added an advanced button to the Component search that allows the user to configure what they are searching for. Needs some sprucing up graphics wise but it is working right now and serializing the configuration to disk.

#7 - 10/13/2009 08:24 PM - Aaron Aaron

Component search is working well now. Speed is very good, results are merged back into one tree, the search is now configurable through the advanced button, and Sean has flattened his overrides into the core module.

Getting the component search to work with objects other than actors and developing a separate search mechanism for the WorkflowRunManager are the only tasks left to do on this one.

#8 - 10/19/2009 12:07 PM - Aaron Aaron

After merging the tree the results of the timing analysis are below. The data set used was slightly different from before (I didn't have any workflows in my workflows directory). As expected stage 3 is significantly faster with the merged tree, ~830 results are displayed in 1.6 seconds rather than 5 seconds in the unmerged tree. Interestingly I got many more results this time around for 'data input'. Not exactly sure why but my guess is that the ontology class names used in the search indexing match correctly now the ontology class names that show up in the tree. Before Sean flattened his core overrides of NamedOntClass I had noticed that the search terms did not all exactly match what was being shown in the tree but many were slightly different. Sean explained that there is a label for the NamedOntClass and a name for the NamedOntClass and that they are similar but do not always match up.

```
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:291) Search term is 'sdf'
Index search generated 8 results in 7ms
Model search generated 14 results in 8ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:296) Library search completed in 15ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:299) Library updated in 50ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:300) Total search time was 66ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:291) Search term is 'data input'
Index search generated 98 results in 5ms
Model search generated 331 results in 33ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:296) Library search completed in 38ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:299) Library updated in 572ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:300) Total search time was 612ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:291) Search term is 'components'
Index search generated 324 results in 9ms
Model search generated 828 results in 50ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:296) Library search completed in 59ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:299) Library updated in 1599ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:300) Total search time was 1660ms
```

#9 - 10/19/2009 12:25 PM - Aaron Aaron

Subsequent searches increase library refresh times substantially for large result sets. This is probably due to objects not being garbage collected from old searches properly. Below is timing analysis showing subsequent searches for 'data input' taking .5 seconds to complete the first time, but 5 seconds to complete the 5th time.

```
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:291) Search term is 'data input'
Index search generated 98 results in 5ms
Model search generated 331 results in 24ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:296) Library search completed in 30ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:299) Library updated in 541ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:300) Total search time was 571ms

DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:291) Search term is ""
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:296) Library search completed in 0ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:299) Library updated in 2ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:300) Total search time was 3ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:291) Search term is 'data input'
Index search generated 98 results in 5ms
Model search generated 331 results in 24ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:296) Library search completed in 29ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:299) Library updated in 593ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:300) Total search time was 623ms

DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:291) Search term is ""
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:296) Library search completed in 0ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:299) Library updated in 3ms
DEBUG (org.kepler.gui.ComponentLibraryTab$SearchButtonActionHandler:actionPerformed:300) Total search time was 3ms
```

DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:291) Search term is 'data input'
Index search generated 98 results in 5ms
Model search generated 331 results in 28ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:296) Library search completed in 33ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:299) Library updated in 2258ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:300) Total search time was 2291ms

DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:291) Search term is "
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:296) Library search completed in 0ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:299) Library updated in 1ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:300) Total search time was 2ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:291) Search term is 'data input'
Index search generated 98 results in 5ms
Model search generated 331 results in 28ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:296) Library search completed in 33ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:299) Library updated in 3569ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:300) Total search time was 3603ms

DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:291) Search term is "
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:296) Library search completed in 0ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:299) Library updated in 2ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:300) Total search time was 3ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:280)
SearchButtonActionHandler.actionPerformed(ActionEvent)
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:291) Search term is 'data input'
Index search generated 98 results in 10ms
Model search generated 331 results in 26ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:296) Library search completed in 36ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:299) Library updated in 5269ms
DEBUG (org.kepler.gui.ComponentLibraryTab\$SearchButtonActionHandler:actionPerformed:300) Total search time was 5307ms

#10 - 01/24/2010 04:23 PM - Aaron Aaron

Since the search has been integrated into the Library Index and the Library no longer uses the ActorMetadata objects directly this problem has gone away. Closing out open a new bug if search performance is an issue in the future.

#11 - 03/27/2013 02:26 PM - Redmine Admin

Original Bugzilla ID was 4362