Application: gvSIG desktop - gvSIG bugs #510
Bad performance loading big vector layers (1mill features)

04/09/2012 01:43 AM - Manuel Madrid

Status:	Closed	**% Done:**	0%
Priority:	Low	**Spent time:**	0.00 hour
Assignee:	Juan Lucas Domínguez		
Category:	Document view		
Target version:	2.0.0-devel-2055		
Severity:		**Add-on version:**	
gvSIG version:	2.0.0	**Add-on build:**	
gvSIG build:	2045	**Add-on resolve version:**	
Operative System:		**Add-on resolve build:**	
Keywords:		**Proyecto:**	
Has patch:	No	**Hito:**	
Add-on name:	Unknown		

Description

I did some simple tests to compare the performance loading vector layer between gvSIG 1.11 and gvSIG 2.0. The tests showed that gvSIG 2.0 is quite slower than gvSIG 1.11 doing this operation.

I used highways.shp, available at:
http://downloads.cloudmade.com/europe/southern_europe/spain

Results:
Windows
- gvSIG 1.11 ~~> 9sec.~~
gvSIG 2.0 ~~> 45sec.~~
~~Linux~~
gvSIG 1.11 ~~> 14sec.~~
gvSIG 2.0 -> 2min 25sec. (it takes 50sec. start rendering)

Associated revisions

Revision 38884 - 09/12/2012 10:05 AM - Cesar Ordiñana

Optimization to simplify geometries to be drawn.
Refs #510

Revision 38889 - 09/13/2012 05:46 AM - Cesar Ordiñana

Remove unneeded check to see is a geometry will be drawn as a point, as it will already be optimized with the r38884 changes.
Refs #510

Revision 38890 - 09/13/2012 05:48 AM - Cesar Ordiñana

Only insert the drawn geometries in the spatial cache if it is in editing mode.
Refs #510

Revision 38891 - 09/13/2012 09:47 AM - Cesar Ordiñana

Solve error in the iteration which left out the last segment. Also replace recursive implementation with iterative one for much better performance.
Refs #510.

Revision 38893 - 09/13/2012 11:00 AM - Cesar Ordiñana

Small improvement in SHP Geometry creation.
Refs #510

History

#1 - 05/10/2012 07:59 PM - Manuel Madrid

- Target version set to 2.0.0-rc1

#2 - 06/22/2012 05:56 PM - Cesar Ordiñana

- Status changed from New to In progress

- Assignee set to Cesar Ordiñana

#3 - 06/26/2012 09:46 AM - Cesar Ordiñana

- gvSIG build changed from 2045 to 2046

#4 - 06/26/2012 10:25 AM - Cesar Ordiñana

- gvSIG build changed from 2046 to 2045

#5 - 06/27/2012 03:37 PM - Cesar Ordiñana

To have a base for how changes might affect performance, current numbers in my machine are, using the same layer:

- gvSIG 1.11: 3-4 seconds.
- gvSIG 2.0 (current development trunk after build 2049): 17-18 seconds.

From now on I will use the following project to test performance related changes:
https://devel.gvsig.org/svn/gvsig-plugintemplates/org.gvsig.example/trunk/org.gvsig.example/org.gvsig.example.viewer

This is a map viewer which gets the store parameters as command line parameters. I have added an option to the BaseTaskStatus that, in debug, shows how much time has taken the status, like the one created by the AbstractVectorialLegend draw method.

#6 - 09/12/2012 04:09 PM - Cesar Ordiñana

It seems we lost an optimization in the geometry draw process: when a geometry is going to be drawn as an AWT Shape, its coordinates are converted to integer and the geometry is simplified if its points are repeated.

This is the case of the highways layer, when the full layer is drawn there are geometries with a lot of points (100) which become much simpler geometries (10 points).

With the r38884 changeset it is readed. Now the drawing takes 15-16 seconds (12% - 17% improvement)

#7 - 09/13/2012 11:50 AM - Cesar Ordiñana

With the changesets r38889 and r38890 now it takes 14 - 15 seconds (7% improvement)

#8 - 09/13/2012 03:51 PM - Cesar Ordiñana

The changes in the r38891 leave it in 11-12 seconds (20% - 22% improvement!!)

#9 - 09/13/2012 06:25 PM - Cesar Ordiñana

Now the part taking most time is the creation of the geometries in the shp provider (50%). I think it could be greatly improved if the providers could directly create the simplified geometries for the view, when a scale is provided.

One posible way to do it could be to use a simplification algorithm, like the Douglas-Peucker one (see also an example usage in Openlayers), with the tolerance factor based on the scale.

Another part that could be improved is the iteration through the features. It takes almost 20% of the time, and from the data only the geometry envelop is read, as for drawing without labeling and the default legend no dbf data is loaded.

One way could be to avoid calling all the resource locking code for each iteration, doing it only once, at least when using a visitor to iterate through the features.

It would be interesting to compare also the performance when data from the dbf is loaded, by using a legend or labeling which needs alphanumeric data of the Feature.

#10 - 09/24/2012 09:54 AM - Joaquín del Cerro Murciano

- Assignee changed from Cesar Ordiñana to Juan Lucas Domínguez

- Priority changed from Normal to Low

#11 - 09/25/2012 10:04 AM - Juan Lucas Domínguez

gvsig-desktop:r38925

Fixed size of arrays (set to maximum needed from the start, they are "final")

#12 - 09/25/2012 10:49 AM - Juan Lucas Domínguez

- Status changed from In progress to Fixed

#13 - 10/01/2012 10:08 AM - Joaquín del Cerro Murciano

- Target version changed from 2.0.0-rc1 to 2.0.0-devel-2055

#14 - 10/16/2012 01:31 PM - Vicent Domenech

- Status changed from Fixed to Closed