

QGIS Application - Bug report #4862

GRASS rasters are very slow to load and are causing crash on Windows

2012-01-23 08:57 AM - Giovanni Manghi

Status: Closed	
Priority: Normal	
Assignee:	
Category: GRASS	
Affected QGIS version: master	Regression?: No
Operating System:	Easy fix?: No
Pull Request or Patch supplied: No	Resolution: fixed
Crashes QGIS or corrupts data: No	Copied to github as #: 14692
Description	
Attached a raster and a location with the very same raster imported via r.in.gdal	
The GRASS raster takes quite a lot more to load if compared with the original one. Given that the raster is very small it is easy to understand that this is an issue for much larger ones. Seems to be a recent regression in master, as not long ago it wasn't that slow.	
http://ubuntuone.com/1Qx63r53iqcxLEU9OTd4UZ	

History

#1 - 2012-01-30 11:35 PM - Paolo Cavallini

See also #4421

#2 - 2012-02-01 04:08 AM - Giovanni Manghi

- Priority changed from Normal to High

One explanation is the following: now in QGIS when adding a GRASS raster the stats/stretching is always done, regardless of what option is saved in the raster properties. If the raster is "big" it takes sometime, moreover the stats are not saved like for other formats (in the .xml file) so there is no speed up when re-loading the same GRASS raster.

#3 - 2012-02-01 10:19 AM - Filipe Dias

This morning I imported a 360 MB DEM raster into grass using r.in.gdal, but when I tried to visualize in QGIS it crashed. On 1.7.3 this does not happen.

#4 - 2012-02-02 02:37 AM - Giovanni Manghi

Filipe Dias wrote:

This morning I imported a 360 MB DEM raster into grass using r.in.gdal, but when I tried to visualize in QGIS it crashed. On 1.7.3 this does not happen.

This is kind of confirmed. With this raster

http://www.faulnia.pt/downloads/mdt_aster_pt-tm06_GeoTIFF.tar.gz

a 360mb DEM (EPSG 3763) it takes a couple of minutes to load into canvas (every time) after being imported in a Mapset. But under Linux it does not crashes. Filipe can you test on Windows with my dataset?

#5 - 2012-02-02 03:06 AM - Filipe Dias

Yes, on Windows 7 QGIS crashes with your raster when I try to import it into the view.

#6 - 2012-02-02 03:13 AM - Giovanni Manghi

- *Crashes QGIS or corrupts data changed from No to Yes*

#7 - 2012-02-02 09:41 AM - Giovanni Manghi

- *Subject changed from GRASS rasters are very slow to load to GRASS rasters are very slow to load and are causing crash on Windows*

#8 - 2012-02-11 02:21 PM - Giovanni Manghi

- *Priority changed from High to 6*

#9 - 2012-04-15 08:46 AM - Giovanni Manghi

- *Priority changed from 6 to High*

#10 - 2012-04-16 03:57 AM - Giovanni Manghi

- *Priority changed from High to Severe/Regression*

#11 - 2012-04-16 04:00 AM - Giovanni Manghi

- *Priority changed from Severe/Regression to High*

#12 - 2012-05-21 04:51 AM - Radim Blazek

With current master on Linux and your `mdt_aster_pt-tm06_GeoTIFF.tif` imported to GRASS it loads in about 3s using by default colormap grayscale and contrast enhancement 'Stretch to MinMax'.

If statistics calculation is forced (by running 'Load min/max values' with 'Actual' checked) then it is really slow. The GRASS provider is using y block size calculated from maximum 10 MB memory block, in this case it results in 266 rows, the raster has 19217 and it is read in $19217/266=73$ blocks. I thought that reading data by launching GRASS module for each block could be a problem, but changing the block size to 50MB does not have any significant effect. It seems that reading data through the GRASS module is simply slow for large amounts of data, solution: write a GRASS module which collects and returns all the statistics and use it in `QgsGrassRasterProvider::bandStatistics()` (IIRC GRASS stores only min/max).

I have also found, that the blocks are read twice, that will be another problem in raster layer or `QgsRasterDataProvider`, not in GRASS provider. I don't know exactly where.

Why it crashes on Windows I don't know.

#13 - 2012-05-21 05:17 AM - Radim Blazek

Data are read twice to get average first and stddev in second pass.

I am not sure if it may be much faster even calculating stats in GRASS module. Try to run `r.stats` on that image.

We should probably consider separate methods for min/max (stored in GRASS) and stddev. The real problem is stddev which is not used for stretch to min/max.

#14 - 2012-05-22 01:34 AM - Giovanni Manghi

- Status changed from Open to Closed
- Resolution set to worksforme

Hi Radim,

the initial issue described in this ticket seems somehow solved. Not sure when this happened. Adding GRASS rasters is now fast and does not crash under Windows.

The problem of the slowness when computing stats seems a general one and not strictly related to GRASS rasters. Maybe we should create a new ticket with your last notes.

#15 - 2012-05-22 01:38 AM - Giovanni Manghi

| *The problem of the slowness when computing stats seems a general one and not strictly related to GRASS rasters.*

see for example #4421

checking right now if it is still true.

#16 - 2012-05-24 09:00 AM - Radim Blazek

- Status changed from Closed to Reopened
- Priority changed from High to Normal

I have implemented stats calculation in QGIS-GRASS module, it reduced the time for the example `mdt_aster_pt-tm06_GeoTIFF.tar.gz` from about 6min to less than 1min. It is available here:

<https://github.com/blazek/Quantum-GIS/commit/5c933cd6ea>

I'll merge it to master after 1.8 (reopening issue to remember)

I have used single pass stddev calculation, which, IMO, should be also used in `QgsRasterDataProvider::bandStatistics()`, it should reduce time to half. See http://en.wikipedia.org/wiki/Standard_deviation and http://en.wikipedia.org/wiki/Computational_formula_for_the_variance

Meeting Markus Neteler and Markus Metz on GRASS HF in Prague today, I suggested, that GRASS should calculate more stats in GRASS libs when raster is written. Computation is not heavy but reading data is.

#17 - 2012-05-24 09:05 AM - Giovanni Manghi

Radim Blazek wrote:

| *I have implemented stats calculation in QGIS-GRASS module, it reduced the time for the example `mdt_aster_pt-tm06_GeoTIFF.tar.gz` from about 6min to less than 1min. It is available here:*

| <https://github.com/blazek/Quantum-GIS/commit/5c933cd6ea>

| *I'll merge it to master after 1.8 (reopening issue to remember)*

| *Meeting Markus Neteler and Markus Metz on GRASS HF in Prague today, I suggested, that GRASS should calculate more stats in GRASS libs when raster is written. Computation is not heavy but reading data is.*

very cool Radim! thanks.

I have used single pass stddev calculation, which, IMO, should be also used in `QgsRasterDataProvider::bandStatistics()`, it should reduce time to half. See http://en.wikipedia.org/wiki/Standard_deviation and http://en.wikipedia.org/wiki/Computational_formula_for_the_variance

should we open a new ticket to suggest use this new method for rasters in general?

#18 - 2012-09-04 12:10 PM - Paolo Cavallini

- Target version set to Version 2.0.0

#19 - 2012-09-05 01:12 AM - Radim Blazek

- Resolution changed from worksforme to fixed

5c933cd6ea was applied to master as commit:eb93568f7 and consequently upgraded to new stats API.

Standard deviation calculation in `QgsRasterDataProvider` was changed to single pass.

Anyway, QGIS is now using data sample only to calc stats, which should be always fast.

I believe that it is fixed.

#20 - 2012-09-05 01:12 AM - Radim Blazek

- Status changed from Reopened to Closed