

QGIS Application - Bug report #17117

Composer crashes with large number of raster images loaded

2017-09-09 01:36 PM - Patrick Dunford

Status:	Closed	
Priority:	High	
Assignee:		
Category:	Map Composer/Printing	
Affected QGIS version:	2.18.12	Regression?: Yes
Operating System:	Debian 9.1 Xubuntu 17.04 Xubuntu 17.10	Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: no timely feedback
Crashes QGIS or corrupts data:	Yes	Copied to github as #: 25016
Description		
affected 2.18 and master Some sample data will be attached shortly.		

History

#1 - 2017-09-09 02:39 PM - Giovanni Manghi

How about memory consumption by the time the crash occur?

#2 - 2017-09-09 03:42 PM - Patrick Dunford

It depends on whether you think the software should behave nicely and give some sort of error message or just crash.

#3 - 2017-09-09 08:31 PM - Giovanni Manghi

- Status changed from Open to Feedback
- Priority changed from Normal to High
- Crashes QGIS or corrupts data changed from No to Yes

Patrick Dunford wrote:

| It depends on whether you think the software should behave nicely and give some sort of error message or just crash.

I think you misunderstood: have you monitored the memory consumption at the time of the crash? did the program eat up all the memory/swap or when it crashed memory consumption was still normal?

The sample project and data would help a lot here.

#4 - 2017-09-10 07:27 AM - Patrick Dunford

OK so I would need to know how to monitor this

and these days with modern OS we expect virtual memory to handle this so I need to know if there is a particular configuration of the computer or in this case VM that is needed to ensure there is enough virtual memory available.

As you can edit the project in the main gui window with that number of layers open (over 1000 in this case), it is composer that is crashing.

sample project here, this is purely a sample, it is not real world because this project is far too complex, it would take me all day to archive the data and would be probably 10 GB of data. This sample works by loading the same group of rasters 5 times.

https://drive.google.com/drive/folders/0B_NF_W8CCnSgY1pCQlZuZTNmYk0?usp=sharing

#5 - 2017-09-10 03:09 PM - Patrick Dunford

I think the question is going to rise around whether composer actually needs the huge amount of memory it demands to produce an image (the issue here is the Save as Image export from Composer).

I have been monitoring the performance in top using a 12 GB virtual machine running Debian 9.1

The editor sits there using 5 GB of virtual memory with this project open. I have turned off a few layers that aren't needed in the composer.

I can use Save as Image from the editor and see no memory spike. The output image is 1326x816 pixels in a PNG.

However if I ask composer to do virtually the same thing it asks for 12 GB of virtual memory just for this task.

Whilst there may well be 12 GB of JPEG images in total in the project it does not need to load every single one of them to render the output.

The output is a PNG image 3425x2362 pixels i.e. 17 megapixels.

As we know the default DPI in composer for outputting to an image is 300 dpi. I imagine I could drop this down to 200 or 100 without losing too much quality for the purpose I am using these maps for.

However I think the question of it having an efficient usage of memory remains relevant.

#6 - 2017-09-10 04:12 PM - Patrick Dunford

- File Tau-100.png added

Attached is an image which for test purposes was output at 100, 200 and 300 dpi. (The 100 dpi version attached)

The 100 dpi image took about 1.5 GB of extra memory. The resulting PNG occupies 1.8 MB of disk space.

The 200 dpi image took about 5 GB of extra memory. The result on disk is 5.7 MB.

The 300 dpi image took about 12 GB of extra memory. The output uses 9.5 MB.

There were something like 700 raster layers loaded and 433 were being displayed in the editor at the time.

I don't use composer with a specific layer set, it always uses the layers displayed in the editor.

The images needed which form the background in this case as geoJpegs total 2 in number and take up 18 MB of disk space.

Everything else you see there is rendered from shapefiles.

If I delete all of the rasters so that I only have the two loaded that are needed to render this image I can render it at much greater dpi without it gobbling up all the memory. At 1000 dpi it took several minutes to render the image but using almost no extra memory therefore without the large amount of disk thrashing for paging.

At 1745 dpi (20 000 x 14 000 pixels i.e. 280 MP) extra memory usage peaked at 3 GB and then dropped back to 1 GB for most of the time that the image file was being written. The image creation took around 7 minutes to complete for a 60 MB final disk size.

Larger resolutions were not tested due to error message dialogs claiming the dpi requested would crash the software.

Clearly composer is hugely wasteful of memory in choosing to load every raster that is turned on for display in the editor when it should only be loading the ones needed to render the background of the actual image region it is outputting to the image or PDF.

#7 - 2017-09-11 07:24 PM - Giovanni Manghi

Thanks for all the details. So the question stand: does the program crash when all the available memory (plus paging/swap) is used?

#8 - 2017-09-12 04:57 AM - Patrick Dunford

I don't see why crash is the only criteria, I don't see why I could not have legitimate concerns that the software is a resource hog, or have an option to configure this behaviour in the software.

#9 - 2017-09-12 08:57 AM - Giovanni Manghi

Patrick Dunford wrote:

I don't see why crash is the only criteria, I don't see why I could not have legitimate concerns that the software is a resource hog, or have an option to configure this behaviour in the software.

I didn't say the contrary, I just asked a "simple" question: does it crashes when it eats up all the memory/swap or before?

#10 - 2017-09-12 12:13 PM - Patrick Dunford

Sorry, I don't know enough about linux virtual memory to test that.

#11 - 2017-09-12 12:44 PM - Giovanni Manghi

Patrick Dunford wrote:

Sorry, I don't know enough about linux virtual memory to test that.

use "top", "htop" or "system monitor" application (name can change depending if you use Gnome/KDE/etc.). Then just have a look how memory consumption grows.

#12 - 2017-09-12 01:48 PM - Giovanni Manghi

- Status changed from Feedback to Open

See also #17119

#13 - 2017-09-12 06:25 PM - Patrick Dunford

This issue does not occur in 2.14

Using the same project and rasters, in 2.14 the 280 MP image (20 000 x 14 000, or 1750 dpi) required about 1 GB of extra memory and took about 3 minutes to render. This was with all the 600 rasters loaded and 433 displayed. At that resolution the 2.18/master composer would have an astronomical memory demand. (I must test it to see how long it takes to crash)

#14 - 2017-09-12 06:29 PM - Giovanni Manghi

- Regression? changed from No to Yes

Patrick Dunford wrote:

This issue does not occur in 2.14

Using the same project and rasters, in 2.14 the 280 MP image (20 000 x 14 000, or 1750 dpi) required about 1 GB of extra memory and took about 3 minutes to render. This was with all the 600 rasters loaded and 433 displayed. At that resolution the 2.18/master composer would have an astronomical memory demand. (I must test it to see how long it takes to crash)

"good". We now know is a regression and as such has usually priority when bug fixing efforts are in place.

#15 - 2017-09-13 08:40 AM - Patrick Dunford

Attempting a 1750 dpi image in master build 313ec55 resulted in a demand for 415 GB of memory (0.415 TB as displayed in top) and the system crashed very rapidly. This VM running Debian 9.1 has 12 GB of physical memory and 8 GB of swap.

Testing on the latest masters shows a very efficient composer rendering resource optimisation (whether through reusing the loaded layer buffers for the canvas or reverting to the 2.14 composer behaviour of only loading the actually required images) at the cost of the high resource use for buffering all of the layers in canvas.

Ideally the preferred solution for working with large projects of this nature (hopefully a per project setting) is going to be the 2.14 behaviour with only the minimum number of layers buffered in canvas and only the minimum number of layers actually needed loaded by the composer to render the output. Whilst it is possible to take down the DPI and reduce the memory footprint for an image render, there is no option to do this for PDFs which appear to render at some preconfigured number of DPI.

#16 - 2019-01-21 12:34 AM - Jürgen Fischer

- Status changed from Open to Feedback

Please test with QGIS 3.4 - QGIS 2.18 reached it's end of life.

#17 - 2019-02-11 12:25 AM - Nyal Dawson

- Status changed from Feedback to Closed

- Resolution set to no timely feedback

Files

Tau-100.png	1.74 MB	2017-09-10	Patrick Dunford
-------------	---------	------------	-----------------