

Alpine Linux - Bug #9190

imagemagick: disable OpenMP for stability

08/05/2018 09:20 PM - Alexander Edland

Status:	Closed	Start date:	08/05/2018
Priority:	Normal	Due date:	
Assignee:	Natanael Copa	% Done:	100%
Category:	Aports	Estimated time:	0.00 hour
Target version:	3.8.2	Security IDs:	
Affected versions:			

Description

ImageMagick uses OpenMP for improved performance. For some software linking ImageMagick, this appears to cause segfaults in OpenMP-related code, but seemingly only when running on certain processors. My testcase (non-trivial, an uncommitted aport) exhibits consistent crashes on one laptop but runs fine on another. Both laptops perform the same minimal test in a clean chroot with the same packages from edge. The laptop exhibiting the crashes has an i5-6200U while the seemingly immune laptop has an i5-6300HQ. The following two stacks have been observed in gdb:

```
Program received signal SIGSEGV, Segmentation fault.
```

```
#0 0x00007ffff4ae0415 in QueueAuthenticPixels () from /usr/lib/libMagickCore-7.Q16HDRI.so.6
#1 0x00007ffffe92e7398 in ?? () from /usr/lib/ImageMagick-7.0.8/modules-Q16HDRI/coders/bmp.so
#2 0x00007ffff4b17892 in ReadImage () from /usr/lib/libMagickCore-7.Q16HDRI.so.6
#3 0x00007ffff526d5c2 in Magick::Image::read(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&) () from /usr/lib/libMagick+-7.Q16HDRI.so.4
#4 0x00007ffff526d61f in Magick::Image::Image(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&) () from /usr/lib/libMagick+-7.Q16HDRI.so.4
#5 0x00007ffff54978ef in readCreate (in=in@entry=0x5555557cd580, out=out@entry=0x555555a156e0, userData=<optimized out>, core=0x5555557a2560, vsapi=0x7ffff6ebbe40 <vs_internal_vsapi>) at src/filters/imwri/imwri.cpp:784
#6 0x00007ffff6c9442d in VSPlugin::invoke (this=this@entry=0x5555558d5e20, funcName=..., args=...) at src/core/vscore.cpp:1694
#7 0x00007ffff6c8aaed in invoke (plugin=0x5555558d5e20, name=<optimized out>, args=0x5555557cd580) at src/core/vsapi.cpp:386
#8 0x00007ffff6f30177 in __pyx_pf_11vapoursynth_8Function_2__call__ (__pyx_v_kwargs=0x7ffff58d33a8, __pyx_v_args=0x7ffff7e996a0, __pyx_v_self=0x7ffff58aafc0) at src/cython/vapoursynth.c:39415
#9 __pyx_pw_11vapoursynth_8Function_3__call__ (__pyx_v_self=0x7ffff58aafc0, __pyx_args=0x7ffff7e996a0, __pyx_kws=<optimized out>) at src/cython/vapoursynth.c:38521
#10 0x00007ffff71e4660 in PyObject_FastCallDict () from /usr/lib/libpython3.6m.so.1.0
#11 0x00007ffff71e4acc in PyObject_FastCallKeywords () from /usr/lib/libpython3.6m.so.1.0
#12 0x00007ffff727184d in ?? () from /usr/lib/libpython3.6m.so.1.0
#13 0x00007ffff7277626 in PyEval_EvalFrameDefault () from /usr/lib/libpython3.6m.so.1.0
#14 0x00007ffff72715a5 in ?? () from /usr/lib/libpython3.6m.so.1.0
#15 0x00007ffff72718d5 in PyEval_EvalCodeEx () from /usr/lib/libpython3.6m.so.1.0
#16 0x00007ffff72718f6 in PyEval_EvalCode () from /usr/lib/libpython3.6m.so.1.0
#17 0x00007ffff6f229d6 in __Pyx_PyExec3 (locals=0x7ffff7f25e10, globals=0x7ffff7f25e10, o=0x7ffff58ef810) at src/cython/vapoursynth.c:79365
#18 vpy_evaluateScript (__pyx_v_se=__pyx_v_se@entry=0x5555557d6960, __pyx_v_script=<optimized out>, __pyx_v_scriptFilename=<optimized out>, __pyx_v_scriptFilename@entry=0x7ffff6e4d8 "red.vpy", __pyx_v_flags=__pyx_v_flags@entry=1) at src/cython/vapoursynth.c:41408
#19 0x00007ffff6f2501b in vpy_evaluateFile (__pyx_v_se=0x5555557d6960, __pyx_v_scriptFilename=<optimized out>, __pyx_v_flags=1) at src/cython/vapoursynth.c:42406
#20 0x00007ffff7b6a198 in vsscript_evaluateFile (handle=0x55555575c210 <se>, scriptFilename=0x7ffff6e4d8 "red.vpy", flags=1) at src/vsscript/vsscript.cpp:160
#21 0x00005555555556bda in main (argc=3, argv=<optimized out>) at src/vspipe/vspipe.cpp:669
```

```
Program received signal SIGSEGV, Segmentation fault.
```

```
#0 0x00007ffff345bfab in omp_get_max_threads () from /usr/lib/libgomp.so.1
#1 0x00007ffff3ae3bb0 in ?? () from /usr/lib/libMagickCore-7.Q16HDRI.so.6
#2 0x00007ffff3a97540 in MagickCoreGenesis () from /usr/lib/libMagickCore-7.Q16HDRI.so.6
#3 0x00007ffff417a63c in Magick::InitializeMagick(char const*) () from /usr/lib/libMagick+-7.Q16HDRI.so.4
#4 0x00007ffff43af835 in initMagick (vsapi=0x7ffff6ebce40 <vs_internal_vsapi>, core=0x55555584f0c
```

```

0) at src/filters/imwri/imwri.cpp:73
#5 0x00007ffff43b058f in readCreate (in=in@entry=0x555555a9b120, out=out@entry=0x555555a99f80, userData=<optimized out>, core=0x55555584f0c0, vsapi=0x7ffff6ebce40 <vs_internal_vsapi>) at src/filters/imwri/imwri.cpp:748
#6 0x00007ffff6c9542d in VSPlugin::invoke (this=this@entry=0x5555558eb4c0, funcName=..., args=...) at src/core/vscore.cpp:1694
#7 0x00007ffff6c8baed in invoke (plugin=0x5555558eb4c0, name=<optimized out>, args=0x555555a9b120) at src/core/vsapi.cpp:386
#8 0x00007ffff6f31137 in __pyx_pf_11vapoursynth_8Function_2__call__ (__pyx_v_kwargs=0x7ffff58d0af8, __pyx_v_args=0x7ffff7e96898, __pyx_v_self=0x7ffff58d0ab0) at src/cython/vapoursynth.c:39415
#9 __pyx_pw_11vapoursynth_8Function_3__call__ (__pyx_v_self=0x7ffff58d0ab0, __pyx_args=0x7ffff7e96898, __pyx_kwds=<optimized out>) at src/cython/vapoursynth.c:38521
#10 0x00007ffff71e8410 in _PyObject_FastCallDict () from /usr/lib/libpython3.6m.so.1.0
#11 0x00007ffff71e887c in _PyObject_FastCallKeywords () from /usr/lib/libpython3.6m.so.1.0
#12 0x00007ffff72755fd in ?? () from /usr/lib/libpython3.6m.so.1.0
#13 0x00007ffff727b3d6 in _PyEval_EvalFrameDefault () from /usr/lib/libpython3.6m.so.1.0
#14 0x00007ffff7275355 in ?? () from /usr/lib/libpython3.6m.so.1.0
#15 0x00007ffff7275685 in PyEval_EvalCodeEx () from /usr/lib/libpython3.6m.so.1.0
#16 0x00007ffff72756a6 in PyEval_EvalCode () from /usr/lib/libpython3.6m.so.1.0
#17 0x00007ffff6f23996 in __Pyx_PyExec3 (locals=0x7ffff7f26ab0, globals=0x7ffff7f26ab0, o=0x7ffff58ee780) at src/cython/vapoursynth.c:79365
#18 vpy_evaluateScript (__pyx_v_se=__pyx_v_se@entry=0x5555558a2cc0, __pyx_v_script=<optimized out>, __pyx_v_scriptFilename=<optimized out>, __pyx_v_scriptFilename@entry=0x7ffff5c8 "red.vpy", __pyx_v_flags=__pyx_v_flags@entry=1) at src/cython/vapoursynth.c:41408
#19 0x00007ffff6f25fdb in vpy_evaluateFile (__pyx_v_se=0x5555558a2cc0, __pyx_v_scriptFilename=<optimized out>, __pyx_v_flags=1) at src/cython/vapoursynth.c:42406
#20 0x00007ffff7b6a198 in vsscript_evaluateFile (handle=0x55555575c210 <se>, scriptFilename=0x7ffff5c8 "red.vpy", flags=1) at src/vsscript/vsscript.cpp:160
#21 0x0000555555556bda in main (argc=3, argv=<optimized out>) at src/vspipe/vspipe.cpp:669

```

...in addition to the following assert when executing the test outside of gdb:

```
Assertion failed: id < (int) cache_info->number_threads (MagickCore/cache.c: QueueAuthenticPixels: 4323)
```

while I cannot find any similar issues where effort has been made to uncover the underlying problem, most people seem satisfied with the "fix" of disabling OpenMP:

- <https://www.imagemagick.org/discourse-server/viewtopic.php?t=32516>
- <https://github.com/termux/termux-packages/issues/1314>
- <https://stackoverflow.com/questions/2838307/why-is-this-rmagick-call-generating-a-segmentation-fault>

contrary to what is mentioned in the termux issue, setting OMP_NUM_THREADS=1 does not prevent the segfault in my scenario, however building ImageMagick with --disable-openmp does.

Associated revisions

Revision a23a75a8 - 08/06/2018 10:19 AM - Natanael Copa

main/imagemagick: upgrade to 7.0.8.8 and disable openmp

openmp appears to cause problems on some machines. We disable it everywhere.

fixes #9190

Revision edd37627 - 11/21/2018 02:37 PM - Natanael Copa

main/imagemagick: disable openmp

openmp appears to cause problems on some machines. We disable it everywhere.

fixes #9190

History

#1 - 08/06/2018 10:21 AM - Natanael Copa

- Status changed from New to Resolved
- % Done changed from 0 to 100

Applied in changeset [a23a75a8962b419f92fe63ba60317496bbfe712](#).

#2 - 10/19/2018 10:48 AM - Chingis S

I also get this error on some applications like a default image uploader in WordPress core. However, I get this randomly, sometimes docker image rebuild helps to resolve it although the ImageMagick pkg was the same. Maybe the issue in depended packages.

Are there any plans to merge this change to 3.8? Because if you install imagemagick from edge it brings a ton of dependencies that causes all sorts of issues when 3.8 and edge packages used together.

#3 - 10/23/2018 06:48 AM - Natanael Copa

- Status changed from Resolved to Assigned
- Target version set to 3.8.2
- % Done changed from 100 to 50

#4 - 11/22/2018 04:08 PM - Natanael Copa

- Status changed from Assigned to Resolved
- % Done changed from 50 to 100

Applied in changeset [edd37627b316f881325df6073fb379a4671d366e](#).

#5 - 12/20/2018 08:45 PM - Natanael Copa

- Status changed from Resolved to Closed