

GammaLib - Bug #1756

Segfault when using OpenMp

04/05/2016 03:49 PM - Mayer Michael

Status:	Closed	Start date:	04/05/2016
Priority:	High	Due date:	
Assigned To:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	1.2.0		

Description

I had this before (#1717) when running ctlike using openmp:

```
$ ctlike debug=yes
Input event list, counts cube or observation definition XML file [myinobs.xml]
Input model XML file [inmodel.xml]
Output model XML file [outmodel.xml]
...
2016-02-28T10:58:14: +=====+
2016-02-28T10:58:14: | Maximum likelihood optimisation |
2016-02-28T10:58:14: +=====+
[1] 106012 segmentation fault  ctlike debug=yes
```

Here is again the backtrace from gdb:

```
(gdb) backtrace
#0 0x0000003e8d86fac1 in fseeko64 () from /lib64/libc.so.6
#1 0x00007ffff7316bc5 in file_seek ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#2 0x00007ffff73087c0 in ffltrc ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#3 0x00007ffff730899b in ffmbyt ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#4 0x00007ffff7309f6d in ffgr8b ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#5 0x00007ffff73573ef in ffgcld.part.0 ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#6 0x00007ffff7351d48 in ffgcv ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#7 0x00007ffff7b1f3c6 in GFitsTableCol::load_column_fixed() () at GFitsTableCol.cpp:589
#8 0x00007ffff7b2c37f in GFitsTableDoubleCol::real(int const&, int const&) const () at GFitsTableDoubleCol.cpp:255
#9 0x00007ffff7c6dc88 in GCTAEventList::read_events(GFitsTable const&) const () at src/GCTAEventList.cpp:992
#10 0x00007ffff7c6eb12 in GCTAEventList::fetch() const () at src/GCTAEventList.cpp:672
#11 0x00007ffff7c6f699 in GCTAEventList::operator[](int const&) const () at src/GCTAEventList.cpp:215
#12 0x00007ffff7bb5179 in GObservation::likelihood_poisson_unbinned(GModels const&, GVector*, GMatrixSparse*, double*) const
    () at GObservation.cpp:924
#13 0x00007ffff7bb36be in GObservation::likelihood(GModels const&, GVector*, GMatrixSparse*, double*) const ()
    at GObservation.cpp:197
#14 0x00007ffff7bb1feb in GObservations::likelihood::eval () at GObservations_likelihood.cpp:270
#15 0x00007ffff7bb2279 in GObservations::likelihood::eval(GOptimizerPars const&) () at GObservations_likelihood.cpp:239
#16 0x00007ffff7b976fb in GOptimizerLM::optimize(GOptimizerFunction&, GOptimizerPars&) () at GOptimizerLM.cpp:225
#17 0x00007ffff7bb036f in GObservations::optimize(GOptimizer&) () at GObservations.cpp:720
#18 0x000000000040561a in ctlike::optimize_lm() () at ctlike.cpp:491
#19 0x0000000000406b81 in ctlike::run() () at ctlike.cpp:243
#20 0x0000000000408bc1 in ctool::execute() () at ctool.cpp:217
#21 0x0000000000405404 in main () at main.cpp:55
```

Apparently, this time the problem occurs when loading the events in GFitsTableCol::load_column_fixed.

Again recompiling without OpenMp makes the problem disappear. We might need to put the relevant into an omp_critical section,

History

#1 - 04/14/2016 10:45 AM - Mayer Michael

When running ctobssim, a similar problem occurs:

```
(gdb) backtrace
#0 0x0000003e8d86fac1 in fseeko64 () from /lib64/libc.so.6
#1 0x00007ffff7317bc5 in file_seek ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#2 0x00007ffff7317db2 in file_write ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#3 0x00007ffff7312404 in fwrite ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#4 0x00007ffff730941c in ffbwt ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#5 0x00007ffff730a890 in fflush ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#6 0x00007ffff7311968 in fclose ()
    from /afs/ihf.de/group/hess/scratch/software/stable/cfitsio/sl6/cfitsio-3.340-gcc-4.8.1/lib/libcfitsio.so
#7 0x00007ffff7b3295a in GFits::free_members() () at GFits.cpp:1436
#8 0x00007ffff7b366cc in GFits::~GFits() () at GFits.cpp:142
#9 0x00007ffff7c63e55 in GCTAObservation::save(GFilename const&, bool const&) const () at src/GCTAObservation.cpp:1291
#10 0x000000000040d003 in ctobssim::run () at ctobssim.cpp:419
#11 0x00007ffff70e035a in gomp_thread_start () at ../../gcc-4.8.1/libgomp/team.c:115
#12 0x0000003e8e407aa1 in start_thread () from /lib64/libpthread.so.0
#13 0x0000003e8d8e893d in clone () from /lib64/libc.so.6
```

#2 - 04/18/2016 12:00 AM - Knödlseider Jürgen

- Priority changed from Normal to High

#3 - 05/30/2016 11:41 PM - Knödlseider Jürgen

I was just inspecting the last problem and found that the relevant line in the code is protected:

```
#pragma omp critical(ctobssim_run)
{
    //obs_clone.save(outfile, clobber());
    obs->save(outfile, clobber());
}
```

I think that this was something changed during the coding sprint. Does the code that you are using have the same protection?

#4 - 05/30/2016 11:47 PM - Knödseder Jürgen

Also for you first problem there is already a critical zone (although unnamed):

```
#pragma omp critical
{
    try {

        // Open FITS file
        GFits fits(m_filename);

        // Initialise events extension name
        std::string extname = fits.filename().extname("EVENTS");

        // Get event list HDU
        const GFitsTable& events = *fits.table(extname);

        // Load event data
        read_events(events);

        // Close FITS file
        fits.close();

    }
    catch (...) {
        has_exception = true;
    }
}
```

#5 - 05/31/2016 09:37 AM - Mayer Michael

Does the code that you are using have the same protection?

Yes, I have the same code compiled, I usually try to keep as close to devel as possible.

Also for you first problem there is already a critical zone (although unnamed):

Is the naming important?

#6 - 05/31/2016 11:22 AM - Knödlseider Jürgen

user#77 wrote:

Is the naming important?

Normally this is only needed to avoid conflicts between different OMP CRITICAL zones (remember, we had a deadlock situation during the last coding sprint).

This seems hence to be more related to the details of OpenMP. I would need do to a bit of more reading to understand what's going on (did not find something evident on google). Maybe it's related to calling the cfitsio library, which by default is not thread save (can however be compiled to be thread save).

#7 - 05/31/2016 02:38 PM - Mayer Michael

- *Status changed from New to Resolved*

I just updated my code and recompiled (using OpenMP) again and cannot reproduce this problem now. I am therefore a bit puzzled what may have produce this problem in the first place. I can investigate a bit but for now we could close this issue.

Note that we have undergone a system upgrade to a new Scientific Linux version in Zeuthen just now (maybe this was a local issue).

In any case, maybe a unit test which runs an analysis for several observations in the container is a good thing to add (could only be executed if OpenMp is present).

#8 - 10/06/2016 05:02 PM - Knödlseider Jürgen

- *Target version set to 1.2.0*

#9 - 03/03/2017 10:14 AM - Knödlseider Jürgen

- *Status changed from Resolved to Closed*