GROMACS - Bug #900

crash in OpenMP code.

03/17/2012 09:50 AM - David van der Spoel

Status:	Closed		
Priority:	High		
Assignee:	Berk Hess		
Category:	mdrun		
Target version:	4.6		
Affected version - extra info:		**Difficulty:**	uncategorized
Affected version:			

Description

When trying to run a simulation on two cores on my MacBook Pro running OSX 10.7 mdrun crashes at the initiation step. The stack dump is here:

```
(gdb) where
#0  0x00000001001552b2 in gomp_team_start ()
#1  0x00000001001bd150 in fft5d_plan_3d ()
#2  0x000000010021008c in gmx_parallel_3dfft_init ()
#3  0x0000000100256acb in gmx_pme_init ()
#4  0x000000010000d5e7 in mdrunner ()
#5  0x000000010000b031 in mdrunner_start_fn ()
#6  0x0000000100750095 in tMPI_Thread_starter ()
#7  0x00007fff8587b8bf in _pthread_start ()
#8  0x00007fff8587eb75 in thread_start ()
```

Associated revisions

Revision 002c4985 - 04/29/2012 05:12 PM - Roland Schulz

Disable OpenMP for llvm-gcc 4.2.x

Fixes #900

Change-Id: Ibd80b2e3768e25f5091441a65785f539dc3b7050

Revision 002c4985 - 04/29/2012 05:12 PM - Roland Schulz

Disable OpenMP for llvm-gcc 4.2.x

Fixes #900

Change-Id: Ibd80b2e3768e25f5091441a65785f539dc3b7050

Revision a6fee0ba - 01/15/2013 09:47 AM - Szilárd Páll

disable OpenMP with all OS X gcc 4.2-based compilers

Although gcc 4.2 should have OpenMP support, the gcc 4.2.1-based
compilers on Mac OS X (defaults in 10.6.x) all claim to support OpenMP,
but generates segfaulting code.

This change reworks the llvm-specific check and moves it out from the
C/CXX flag generation module.

This compiler is the default on many BSD os-es, but as no other gcc
4.2.x has been tested yet (to my knowledge), for now the limitation is
introduced only for the Mac OS X and gcc 4.2.x.

Refs #900

Change-Id: I1c2a27f6fc1162cf8999c65ff6173121109cfbad

History

#1 - 03/17/2012 09:52 AM - David van der Spoel

This is with the release-4-6 branch, freshly fetched and compiled with the default compiler:

```
[artemisiabmcuuse:charmm-pol-2/test] % gcc -version
i686-apple-darwin11-llvm-gcc-4.2: no input files
```

#2 - 03/17/2012 10:01 AM - David van der Spoel

One more thing, it was built with:

```
cmake -DGMX_DOUBLE:BOOL=ON -DCMAKE_BUILD_TYPE:STRING=Debug
```

#3 - 03/17/2012 11:22 AM - Berk Hess

- Assignee changed from Sander Pronk to Berk Hess

Which version is this?
We pushed a new version to the repository yesterday evening. If you didn't use this, please try this version and report back.

PS I assumed you manually assigned this issue to Sander. But this code is Roland's and Szilard and I have been looking at this as well.

#4 - 03/17/2012 11:26 AM - David van der Spoel

It is release-4-6, uploaded this morning. It seems to work fine on my Linux server though, so it seems to be Apple specific.

#5 - 03/17/2012 11:34 AM - Berk Hess

Or compiler specific. Some gcc versions don't behave properly with OpenMP.

Is this run actually using OpenMP threads?
I assume you started without options, which would fill your cores with tMPI threads with each one OpenMP thread. Then it would surprise me if you get a crash in OpenMP.

#6 - 03/17/2012 12:26 PM - David van der Spoel

As you can see from the stack dump, it crashes in the initialization of OMP from fft5d.c.

#7 - 03/28/2012 09:58 AM - David van der Spoel

Maybe cmake does not check whether the FFTW library is built with the -fomp flag (it is not when installing fftw-3 using macports).

The fft5d.c code says:

```
#ifdef FFT5D_THREADS
#include <omp.h>
/* requires fftw compiled with openmp /
/ #define FFT5D_FFTW_THREADS (now set by cmake) */
#endif
```

If I compile gromacs with -DGMX_OPENMP:BOOL=OFF it works fine on two threads on my mac.

#8 - 03/30/2012 02:11 PM - Roland Schulz

The code isn't using the OpenMP from FFTW. We implemented the OpenMP ourselves.

#9 - 03/31/2012 01:58 AM - Roland Schulz

I can't reproduce this on Linux with GCC 4.2.1 and FFTW 3.2.2.1. The Jenkins Mac doesn't have FFTW installed so I can't test it there.

#10 - 03/31/2012 02:32 AM - Szilárd Páll

I can reproduce the issue on the Mac OS Lion Jenkins build machine with current realease-4-6, mdrun build with clang and linked against both the fftw from MacPorts (@Roland: now it's installed!) and one that I compiled myself.

The crash happens with nt > 1.

```
Program received signal EXC_BAD_ACCESS, Could not access memory.
Reason: KERN_INVALID_ADDRESS at address: 0x0000000000000010
[Switching to process 16852 thread 0x1a03]
0x00000001000d0bf2 in gomp_team_start ()
(gdb) bt
#0  0x00000001000d0bf2 in gomp_team_start ()
#1  0x000000010010811e in fft5d_plan_3d (NG=12061024, MG=12061024, KG=12061024, comm=0x100b80960, flags=12061024,
rlin=0x100b80960, rlout=0x102081420, rlout2=0x100b809b0, rlout3=0x100b809a8, nthreads=1) at fft5d.c:158
```

#2 0x000000010012b341 in gmx_parallel_3dfft_init (pfft_setup=0x102081440, ndata=0x3800000038, real_data=0x102081400, complex_data=0x100b809f0, comm=0x5, slab2index_major=0x102081400, slab2index_minor=0x100d03a50, bReproducible=0, nthreads=1) at gmx_parallel_3dfft.c:71
#3 0x0000000100150c62 in gmx_pme_init (pmedata=0x100b80aa0, cr=0x100d010f0, nnodes_major=12061344, nnodes_minor=34084376, ir=0x102081228, homenr=12061344, bFreeEnergy=0, bReproducible=0, nthread=1) at pme.c:3081
#4 0x0000000100007c22 in mdrunner (fplog=0x0, nthreads_requested=34057728, cr=0x100d010f0, nfile=12062032, fnm=0x100b80d50, oenv=0x100b80d50, bVerbose=1, bCompact=1, nstglobalcomm=-1, ddxyz=0x100b80e4c, dd_node_order=1, rdd=1.69025069e-38, rconstr=1.69025069e-38, dddlb_opt=0x10001f54e "auto", dlb_scale=1.69025069e-38, ddcsx=0x0, ddcsy=0x0, ddcsz=0x0, nstepout=100, resetstep=-1, nmultisim=0, repl_ex_nst=0, repl_ex_seed=-1, pforce=1.69025069e-38, cpt_period=1.69025069e-38, max_hours=1.69025069e-38, deviceOptions=0x10001aed0 "", Flags=0) at runner.c:848
#5 0x0000000100006949 in mdrunner_start_fn (arg=0x100b80ef0) at runner.c:177
#6 0x000000010039be7e in tMPI_Thread_starter (arg=0x100a03458) at tmpi_init.c:378
#7 0x00007fff8e3de8bf in _pthread_start ()
#8 0x00007fff8e3e1b75 in thread_start ()

#11 - 03/31/2012 02:43 AM - Roland Schulz

In your case something else might be going wrong. Your NG,MG,KG values are way too large. For me it crashes too on Jenkins-Mac but I get reasonable values:

```
#0  0x0000000100155412 in gomp_team_start ()
#1  0x00000001001bd1d0 in fft5d_plan_3d (NG=13, MG=24, KG=24, comm=0x101f80118, flags=5, rlin=0x10581ad48, rlout=0x10581ad68, rlout2=0x101f80100, rlout3=0x101f800f8, nthreads=1) at fft5d.c:158
#2  0x000000010020fedc in gmx_parallel_3dfft_init (pfft_setup=0x10581ad88, ndata=0x101f80304, real_data=0x10581ad48, complex_data=0x10581ad68, comm=0x10581ac28, slab2index_major=0x101802350, slab2index_minor=0x101802390, bReproducible=0, nthreads=1) at gmx_parallel_3dfft.c:71
```

I used gcc 4.2 with the MacPorts FFTW latest 4.6 with Debug build type. But maybe your wrong values are actually not a separate problem. The error occurs in the first line of the functions, so maybe in your trace the values aren't loaded yet.

How are you using clang with OpenMP? I think Clang doesn't support OpenMP (yet)?
http://www.phoronix.com/scan.php?page=news_item&px=MTA0Mzc

To me this looks like a GCC 4.2 bug. I suggest we close it as WONTFIX. We could auto-detect gcc 4.2 and tell the user that we disable OpenMP and that a newer GCC is required for OpenMP.

#12 - 03/31/2012 02:56 AM - Szilárd Páll

Well, it could still be related, to me it seems that it's not those funky values are causing the issue as the crash seems to happen at fft5d.c:158 which is the entry point of fft5d_plan_3d() where none of those variables is used yet.

#13 - 03/31/2012 02:58 AM - Szilárd Páll

Btw, it works with gcc from MacPorts so I suspect it is a clang specific issue.

#14 - 03/31/2012 03:04 AM - Roland Schulz

I suppose you mean llvm with gcc frontend not clang (see my updated message #11). It is a good question whether this is a bug in libgomp of gcc 4.2 which only affects Mac or of llvm. I would suspect it is the first, because according to http://lists.cs.uiuc.edu/pipermail/llvmdev/2011-May/039912.html the OpenMP constructs are just passed though to the gomp library. But I suppose we would need to have a normal (with gcc backend) gcc 4.2 on Mac to test that. But either way it would be enough to disable OpenMP with gcc 4.2 on Mac.

#15 - 03/31/2012 04:00 AM - Szilárd Páll

Yes, I meant llvm.

Unfortunately:
Error: gcc42 does not build on Snow Leopard or later.
through MacPorts.

#16 - 03/31/2012 04:28 AM - Roland Schulz

I just tried llvm-gcc (GCC) 4.2.1 on Linux (jenkins master) and it gives:

```
$ gcc test.c -fopenmp
llvm-gcc: libgomp.spec: No such file or directory
```

In this case it already works that it automatically disables OpenMP.
For llvm-gcc 4.5 (Ubunut 11.10 my machine) OpenMP works correctly.

I would say we should disable OpenMP for gcc 4.2 on Mac or with LLVM to be sure. Or simply all gcc 4.2. People shouldn't use that old gcc anyhow but since it is the default on Mac it might be useful to not have it crash.

#17 - 04/03/2012 03:43 PM - Szilárd Páll

Roland Schulz wrote:

> I would say we should disable OpenMP for gcc 4.2 on Mac or with LLVM to be sure. Or simply all gcc 4.2. People shouldn't use that old gcc anyhow but since it is the default on Mac it might be useful to not have it crash.

I would say we can disable OpenMP for all gcc 4.2, but not silently. The new GPU acceleration doesn't mnake much sense without OpenMP.

#18 - 04/16/2012 12:01 PM - Berk Hess

Does anyone know if the OpenMP performance is good in gcc 4.1 and/or 4.2?
If it's not, it's not much of an issue to disable it (with a warning).

#19 - 04/16/2012 12:40 PM - David van der Spoel

Which OS and machines are still using gcc4.2, apart from Mac OS X?
Does disabling OpenMP mean that GPU computing is useless too? In that case it might be slightly important, but then again, remember that the only feasible Mac+GPU platform is the MacPro and those haven't been updated for a few years now. So in practice disabling OpenMP on Mac with gcc <= 4.2 should not be a problem.

Did anyone test it on a mac with newer gcc through macports?
I can try that otherwise.

#20 - 04/16/2012 02:29 PM - Szilárd Páll

David van der Spoel wrote:

> Which OS and machines are still using gcc4.2, apart from Mac OS X?

The 5.x RHEL/CentOS uses gcc 4.1.x which is anyway too old as it doesn't have OpenMP support. Otherwise I don't know of any reasonably recent OS that has gcc 4.2 as the default.

> Does disabling OpenMP mean that GPU computing is useless too? In that case it might be slightly important, but then again, remember that the only feasible Mac+GPU platform is the MacPro and those haven't been updated for a few years now. So in practice disabling OpenMP on Mac with gcc <= 4.2 should not be a problem.

GPUs acceleration is not entirely useless without OpenMP, but it greatly affects the performance especially with PME.

> Did anyone test it on a mac with newer gcc through macports?
> I can try that otherwise.

Yes, newer gcc versions through Macpors work fine on (on Lion).

#21 - 04/16/2012 02:34 PM - Szilárd Páll

Btw, does this happen to be at fft5d.c:505? If that's the case it's probably a (known) stupid OpenMP issue: the ordered OpenMP statement causes issues with various compilers.

#22 - 04/24/2012 05:28 PM - Roland Schulz

- If I replace ordered with "critical" it still segfaults. Also the segfault is at:

  ```
  #0  0x0000000100155412 in gomp_team_start ()
  #1  0x00000001001bd1d0 in fft5d_plan_3d (NG=15, MG=28, KG=28, comm=0x101980118, flags=5, rlin=0x10380e948,
   rlout=0x10380e968, rlout2=0x101980100, rlout3=0x1019800f8, nthreads=1) at fft5d.c:158
  ```

  This is the first line of fft5d_plan_3d.

- I fixed the spec error with llvm-gcc on Linux (creating a symlink from lib64/*spec to lib) and I also get an error. The complex/nacl test gives:

  ```
  ==16323== Invalid read of size 4
  ==16323==    at 0x43FB73: pme_calc_pidx_wrapper.omp_fn.2 (in /var/lib/jenkins/testing/gromacs/llvm/src/ker
  nel/mdrun_d)
  ==16323==    by 0x443DB6: gmx_pme_do (in /var/lib/jenkins/testing/gromacs/llvm/src/kernel/mdrun_d)
  ==16323==    by 0x4365A4: do_force_lowlevel (in /var/lib/jenkins/testing/gromacs/llvm/src/kernel/mdrun_d)
  ==16323==    by 0x499826: do_force (in /var/lib/jenkins/testing/gromacs/llvm/src/kernel/mdrun_d)
  ==16323==    by 0x42A1F0: do_md (in /var/lib/jenkins/testing/gromacs/llvm/src/kernel/mdrun_d)
  ==16323==    by 0x420E67: mdrunner (in /var/lib/jenkins/testing/gromacs/llvm/src/kernel/mdrun_d)
  ```

```
==16323==    by 0x42F57B: main (in /var/lib/jenkins/testing/gromacs/llvm/src/kernel/mdrun_d)
==16323==  Address 0x7d71cfc is 4 bytes before a block of size 8 alloc'd
==16323==    at 0x4C267CC: calloc (vg_replace_malloc.c:467)
==16323==    by 0x518B15: save_calloc (in /var/lib/jenkins/testing/gromacs/llvm/src/kernel/mdrun_d)
==16323==    by 0x44721D: init_atomcomm (in /var/lib/jenkins/testing/gromacs/llvm/src/kernel/mdrun_d)
==16323==    by 0x44945A: gmx_pme_init (in /var/lib/jenkins/testing/gromacs/llvm/src/kernel/mdrun_d)
==16323==    by 0x420B71: mdrunner (in /var/lib/jenkins/testing/gromacs/llvm/src/kernel/mdrun_d)
==16323==    by 0x42F57B: main (in /var/lib/jenkins/testing/gromacs/llvm/src/kernel/mdrun_d)
==16323==

-------------------------------------------------------
Program mdrun_d, VERSION 4.6-dev-20120423-6a4e6-unknown
Source code file: /var/lib/jenkins/testing/gromacs/src/mdlib/pme.c, line: 767

Fatal error:
17 particles communicated to PME node 0 are more than 2/3 times the cut-off out of the domain decompositio
n cell of their charge group in dimension x.
This usually means that your system is not well equilibrated.
For more information and tips for troubleshooting, please check the GROMACS
website at http://www.gromacs.org/Documentation/Errors
-------------------------------------------------------
```

I get this error only with Release build. The debug build runs correctly. Thus I assume this is also a bug in gcc.

- With Release build type on Mac (with the default llvm based gcc 4.2) I get:

```
/Users/jenkins/testing/gromacs/src/mdlib/genborn_sse2_double.c:930: internal compiler error: Segmentation
fault: 11
Please submit a full bug report,
with preprocessed source if appropriate.
See <URL:http://developer.apple.com/bugreporter> for instructions.
make[3]: *** [src/mdlib/CMakeFiles/md.dir/genborn_sse2_double.c.o] Error 1
make[3]: *** Waiting for unfinished jobs....
```

- Thus we have 3 errors with llvm-gcc 4.2 on either Linux or Mac: OpenMP bug. SSE bug. And Optimization bug. Thus I suggest we completely disable llvm-gcc 4.2 in CMake saying that it is broken.

#23 - 04/27/2012 12:26 AM - Szilárd Páll

I agree with the suggested solution, thanks for the extensive testing.

How about the non-llvm gcc 4.2? Has anyone tested it?

#24 - 04/27/2012 03:03 AM - Roland Schulz

non-llvm works (see comment #9)

#25 - 04/27/2012 03:11 AM - Szilárd Páll

Roland Schulz wrote:

> non-llvm works (see comment #9)

AFAIK on Linux if a newer gcc is the default, newer libgomp (and libstdc) is used. Wouldn't be good to test on an older distribution with gcc 4.2 as default, like RHEL/CentOS 5.x? I don't have any suitable OS at hand.

#26 - 04/27/2012 03:21 AM - Roland Schulz

I just checked. The binary I created for #9 is linked against libgomp from 4.2.1.

#27 - 06/23/2012 06:46 PM - Roland Schulz

- Status changed from New to Closed

Fixed by 002c4985c1d839810816b5c1ba347634b7d7cabb

Files

topol.tpr	54.6 KB	03/17/2012	David van der Spoel