GROMACS - Bug #2085

mdrun crashes with segmentation fault if started with more than 32 OpenMP threads.

11/30/2016 11:15 AM - Jiri Kraus

Status: Closed
Priority: Normal
Assignee: Berk Hess
Category: mdrun
Target version: 2016.2
Affected version - extra info:
Affected version: 2016.1
Difficulty: uncategorized

Description

I build GROMACS 2016.1 with CUDA 8 and GCC 5.4.0 using the following configure command:

CC=gcc  CXX=g++  cmake ../gromacs-2016.1 -DGMX_GPU=ON -DGMX_BUILD_OWN_FFTW=ON -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX=...

With more than 32 OpenMP threads mdrun crashes on water/0000.96:

```shell
[0000.96]$ gmx mdrun -pin on -ntmpi 1 -ntomp 33 -nb gpu -noconfout -resethway -v -maxh 0.08333 -nsteps 100000
```

```shell
[0000.96]$ gmx mdrun -pin on -ntmpi 1 -ntomp 32 -nb gpu -noconfout -resethway -v -maxh 0.08333 -nsteps 100000
```

:-( GROMACS - gmx mdrun, 2016.1 (-:  

[...] snip

starting mdrun 'Water'

100000 steps, 200.0 ps.

Segmentation fault

With 32 threads it runs:

```shell
[0000.96]$ gmx mdrun -pin on -ntmpi 1 -ntomp 32 -nb gpu -noconfout -resethway -v -maxh 0.08333 -nsteps 100000
```

```shell
[0000.96]$ gmx mdrun -pin on -ntmpi 1 -ntomp 32 -nb gpu -noconfout -resethway -v -maxh 0.08333 -nsteps 100000
```

:-( GROMACS - gmx mdrun, 2016.1 (-:  

[...] snip

Core t (s)  Wall t (s)  (%)
Time: 437.159  13.661  3200.0  
(ns/day)  (hour/ns)
Performance: 632.459  0.038

GROMACS reminds you: "Hangout In the Suburbs If You've Got the Guts" (Urban Dance Squad)

```shell
[0000.96]$
```

Even it if does not make sense to use that many OpenMP threads mdrun should not crash.

Associated revisions

Revision c0a90b8b - 12/14/2016 02:14 PM - Berk Hess

Add bonded #thread runtime check

Replaced a debug assertion on the number of OpenMP threads not being larger than GMX_OPENMP_MAX_THREADS by fatal error.

But since the listed forces reduction is actually not required with listed forces, these are now conditional and mdrun can run with more than GMX_OPENMP_MAX_THREADS threads.

Fixes #2085.

Change-Id: I7a6049d727924cd0b4df10a3525f9e7aeec49c3dc

History
Indeed, it should not crash. By design GMX_OPENMP_MAX_THREADS needs to be set at compile time to be able to use >32 threads/rank (this sets #bits used by the the masked sparse reduction in the bondeds). I'm not sure if the check broke recently or something else is wrong. ( IIRC this worked last time I tested on Power 8 a few weeks ago. )

I can't reproduce this with GMX_OPENMP_MAX_THREADS=64 on x86. Could you run a debugger to see where it segfaults?

As Szilard said in comment #1 I would say its expected that the error does not reproduce with GMX_OPENMP_MAX_THREADS=64. I did not file this issue because I expected mdrun to work with more than 32 threads if GMX_OPENMP_MAX_THREADS is not changed. But if the number of OpenMP threads requested on the command line exceeds GMX_OPENMP_MAX_THREADS mdrun should either exit with a understandable error message or cap the number of OpenMP threads to GMX_OPENMP_MAX_THREADS and print a warning. Does that make sense?

Ah, I misunderunderstood Szilard's comment. Only some parts of the code that use OpenMP have a thread count limitation. I'll check where checks are missing.

I uploaded a fix that not only adds the check, but also skips the OpenMP bonded thread reduction for cases without bondeds. So now the water system can run on 33 threads with GMX_OPENMP_MAX_THREADS=32.

Applied in changeset c0a90b8bc7f22ba11e46616b2da4c439946e0786.

- Status changed from Resolved to Closed