GROMACS - Bug #2769

mdrun freezes when running too many threads

11/19/2018 07:24 PM - Magnus Lundborg

Status:	Closed		
Priority:	Normal		
Assignee:	Magnus Lundborg		
Category:	mdrun		
Target version:	2019		
Affected version - extra info:		**Difficulty:**	uncategorized
Affected version:	2019-beta2		

Description

When this tpr on a machine with 4 physical cores (8 virtual threads) it hangs if running too many threads.
gmx mdrun -nt 32 -maxh 0.0005
hangs without any message whereas
gmx mdrun -nt 16 -maxh 0.0005
works as expected.

I am aware that there is no point in running that many threads. However, it is routinely done by Copernicus to check how many threads is possible for each job (unless specified by the user).

Associated revisions

Revision 707a94f6 - 11/29/2018 10:36 AM - Magnus Lundborg

Make pull with COM from previous step work with MPI

There was no communication between the ranks, which caused
crashes with MPI and tMPI. This fixes that.
Minor clean-ups of pull with COM from previous step as well.
Fixes #2769

Change-Id: I3b321872ffd4b295c4e97029d8d54872b3674ac4

History

#1 - 11/19/2018 08:27 PM - Berk Hess

Which tpr?
I assume it's using thread-MPI. I can't think of an explanation for this.

#2 - 11/19/2018 08:31 PM - Magnus Lundborg

- File topol.tpr added

Yes, it's thread-MPI. Hopefully the tpr is attached this time.

#3 - 11/19/2018 08:48 PM - Berk Hess

I can't reproduce this on my machine with 6 cores, also not when only using 4 physical cores.
Could you compile a version with debug symbols, run it in the debugger and press ctrl-c when it hangs to find out where it hangs?

#4 - 11/20/2018 10:19 AM - Magnus Lundborg

I can reproduce it on my desktop machine when running with "-pme cpu -nb cpu -bonded cpu". The machine on which I first had the problem does not have a GPU, so it is probably related to that.

#5 - 11/20/2018 10:26 AM - Magnus Lundborg

This is the stack trace:

```
#0  tMPI_Event_wait (ev=0x7a5330) at /home/magnusl/gromacs/src/external/thread_mpi/src/event.cpp:71
#1  0x00007ffff58376ef in tMPI_Wait_for_others (cev=0x7a5150, myrank=0) at
/home/magnusl/gromacs/src/external/thread_mpi/src/collective.cpp:522
#2  0x00007ffff5836047 in tMPI_Bcast (buffer=0xb56530, count=4, datatype=0x7ffff7db62e0 <tmpi_byte>, root=0, comm=0x6d7b00) at
/home/magnusl/gromacs/src/external/thread_mpi/src/bcast.cpp:98
```

```
#3  0x00007ffff4d72703 in gmx_bcast (nbytes=4, b=0xb56530, cr=0x6d8a80) at /home/magnusl/gromacs/src/gromacs/gmxlib/network.cpp:265
#4  0x00007ffff56b2a99 in nblock_bc<char> (cr=0x6d8a80, numElements=4, data=0xb56530 "CAL") at
/home/magnusl/gromacs/src/gromacs/mdlib/broadcaststructs.h:65
#5  0x00007ffff56af84c in bc_symtab (cr=0x6d8a80, symtab=0x7fffffffc518) at /home/magnusl/gromacs/src/gromacs/mdlib/broadcaststructs.cpp:175
#6  0x00007ffff56b1dcc in bcast_ir_mtop (cr=0x6d8a80, inputrec=0x7fffffffc560, mtop=0x7fffffffc320) at
/home/magnusl/gromacs/src/gromacs/mdlib/broadcaststructs.cpp:797
#7  0x00007ffff56b217e in init_parallel (cr=0x6d8a80, inputrec=0x7fffffffc560, mtop=0x7fffffffc320) at
/home/magnusl/gromacs/src/gromacs/mdlib/broadcaststructs.cpp:851
#8  0x00007ffff577f094 in gmx::Mdrunner::mdrunner (this=0x7fffffffc9a0) at /home/magnusl/gromacs/src/gromacs/mdrun/runner.cpp:658
#9  0x000000000040e5c7 in gmx::gmx_mdrun (argc=9, argv=0x7fffffffd640) at /home/magnusl/gromacs/src/programs/mdrun/mdrun.cpp:292
#10 0x00007ffff48aa8d5 in gmx::(anonymous namespace)::CMainCommandLineModule::run (this=0x6abc20, argc=9, argv=0x7fffffffd640) at
/home/magnusl/gromacs/src/gromacs/commandline/cmdlinemodulemanager.cpp:133
#11 0x00007ffff48ac5c0 in gmx::CommandLineModuleManager::run (this=0x7fffffffd510, argc=9, argv=0x7fffffffd640) at
/home/magnusl/gromacs/src/gromacs/commandline/cmdlinemodulemanager.cpp:589
#12 0x000000000040c02b in main (argc=10, argv=0x7fffffffd638) at /home/magnusl/gromacs/src/programs/gmx.cpp:60
```

#6 - 11/20/2018 09:24 PM - Berk Hess

It doesn't actually hang there, MPI communication is just very slow.
It actually seems to hang at step 0 in initPullComFromPrevStep(). My guess is that your code causes a deadlock when not all ranks are participating in pulling.

#7 - 11/20/2018 09:30 PM - Berk Hess

- Assignee set to Magnus Lundborg

The issue is simpler. The initPullComFromPrevStep() is within a MASTER conditional, but this call does MPI communication. I hope you can fix this yourself.

#8 - 11/23/2018 02:18 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #2769.
Uploader: Magnus Lundborg (magnus.lundborg@scilifelab.se)
Change-Id: gromacs~release-2019~I3b321872ffd4b295c4e97029d8d54872b3674ac4
Gerrit URL: https://gerrit.gromacs.org/8746

#9 - 11/23/2018 03:44 PM - Mark Abraham

Might this fix relate to that for #2776?

#10 - 11/23/2018 05:14 PM - Magnus Lundborg

The commit that fixes this (Gerrit commit 8746) does not fix #2776. I'm looking into if there's any relationship to the pull prev step COM (Gerrit commit 8060) and #2776. Right now I cannot see why they would be related, but I cannot say that they are not.

#11 - 11/29/2018 10:40 AM - Mark Abraham

- Status changed from New to Fix uploaded

#12 - 11/29/2018 10:40 AM - Mark Abraham

- Status changed from Fix uploaded to Resolved

#13 - 11/29/2018 10:40 AM - Mark Abraham

- Status changed from Resolved to Closed

Files

topol.tpr	1.08 MB	11/19/2018	Magnus Lundborg