

GROMACS - Task #2448

should mdrun -multidir permit only one directory?

03/13/2018 04:33 PM - Viveca Lindahl

Status:	Accepted
Priority:	Normal
Assignee:	
Category:	mdrun
Target version:	
Difficulty:	uncategorized
Description	
<p>"mpirun -np 1 \$gmx mdrun -v -stepout 1 -nstlist 1 -npme 0 -ntomp 1 -maxh 0.0001 -multidir sim-0" gives error:</p> <pre>[tcb107:1906] *** An error occurred in MPI_comm_size [tcb107:1906] *** on communicator MPI_COMM_WORLD [tcb107:1906] *** MPI_ERR_COMM: invalid communicator [tcb107:1906] *** MPI_ERRORS_ARE_FATAL: your MPI job will now abort</pre>	
<p>while "-np 2" works fine.</p>	

Associated revisions

Revision e96bbcf - 03/20/2018 12:58 PM - Berk Hess

Disallow single simulation with multisim

To avoid many untested code paths, we, at least temporarily, disable a single simulation with mdrun -multisim and -multidir.

Refs #2448

Change-Id: I9befdb677f3c77bcc6dc0c77abc3478f0b83ba8d

History

#1 - 03/13/2018 04:59 PM - Mark Abraham

Not sure we should intend this to work, but anyway it shouldn't segfault

#2 - 03/13/2018 05:54 PM - Mark Abraham

Viveca Lindahl wrote:

It is likely not a common use case, but it can be useful for running in a directory without changing directory or for benchmarking.

```
(cd sim-0; mpirun ...)
```

also works with thread-MPI ;-)

#3 - 03/13/2018 06:07 PM - Viveca Lindahl

Mark Abraham wrote:

Viveca Lindahl wrote:

It is likely not a common use case, but it can be useful for running in a directory without changing directory or for benchmarking.

```
(cd sim-0; mpirun ...)
```

also works with thread-MPI ;-)

Suure.. I'd still consider this a bug though. Just like it's useful to allow 1-size arrays (even though one could just call it a number then) ;)

#4 - 03/13/2018 07:11 PM - Viveca Lindahl

Seems like the problem happens in runner.cpp:

checkHardwareOversubscription(numThreadsOnThisRank,*hwinfo->hardwareTopology, cr, mdlog); in which:

```
#if GMX_MPI
    if (PAR(cr) || MULTISIM(cr))
    {
        /* Count the threads within this physical node */
        MPI_Comm_size(cr->mpi_comm_physicalnode, &numRanksOnThisNode);
        MPI_Allreduce(&numThreadsOnThisRank, &numThreadsOnThisNode, 1, MPI_INT, MPI_SUM, cr->mpi_comm_physical
node);
    }
#endif
```

and for -multidir and a single rank mpi_comm_physicalnode=0x0. 1 rank without -multidir (mpirun -np 1 [...]) works fine however although I guess these two cases should behave the same here.

#5 - 03/14/2018 10:00 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2448](#).

Uploader: Berk Hess (hess@kth.se)

Change-Id: gromacs~release~2018~leffa1b3bcc23396751490639643502a7fc84a375

Gerrit URL: <https://gerrit.gromacs.org/7676>

#6 - 03/14/2018 10:02 AM - Berk Hess

- Status changed from New to Fix uploaded

- Target version set to 2018.1

Note that this code changed in master and I think -multidir with a single sim will not even start because the chdir is not done.

#7 - 03/14/2018 10:08 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2448](#).

Uploader: Berk Hess (hess@kth.se)

Change-Id: gromacs~master~lbc048af58ec60a5f058311193ba8a5c61a17a5a6

Gerrit URL: <https://gerrit.gromacs.org/7677>

#8 - 03/14/2018 10:51 AM - Szilárd Páll

Affected version shouldn't be 2018.1 as that's not been released yet, 2018 seems better, I think. I assume this is present in 2018 rather than introduced since.

#9 - 03/14/2018 11:12 AM - Viveca Lindahl

Berk Hess wrote:

Note that this code changed in master and I think -multidir with a single sim will not even start because the chdir is not done.

Yes it does. As the description says, if there are 2 ranks and 1 sim it works. There seems to be a resistance to having this work; does it really add so much complication to allow the "trivial" case?

#10 - 03/14/2018 11:13 AM - Viveca Lindahl

Szilárd Páll wrote:

Affected version shouldn't be 2018.1 as that's not been released yet, 2018 seems better, I think. I assume this is present in 2018 rather than introduced since.

Thanks. I just saw that version was 2018.1 dev and so I put 2018.1 there. I'll update.

#11 - 03/14/2018 11:13 AM - Viveca Lindahl

- Affected version changed from 2018.1 to 2018

#12 - 03/14/2018 12:17 PM - Mark Abraham

Viveca Lindahl wrote:

Berk Hess wrote:

Note that this code changed in master and I think -multidir with a single sim will not even start because the chdir is not done.

Yes it does. As the description says, if there are 2 ranks and 1 sim it works. There seems to be a resistance to having this work; does it really add so much complication to allow the "trivial" case?

We clearly can't do adequate testing of the cases we really intend users to use. If there's two ways to run a single simple simulation, then that doubles the amount of testing that in principle we should do.

We have a long history of adding things like -testverlet and -nsteps and -append and -cpi, perhaps testing them only in some subcases, and users adopt them for uses that were never intended, and now you're stuck with the problem. "It's nice because I know how to use it properly" isn't a sufficient argument.

#13 - 03/14/2018 12:38 PM - Viveca Lindahl

Mark Abraham wrote:

Viveca Lindahl wrote:

Berk Hess wrote:

Note that this code changed in master and I think -multidir with a single sim will not even start because the chdir is not done.

Yes it does. As the description says, if there are 2 ranks and 1 sim it works. There seems to be a resistance to having this work; does it really add so much complication to allow the "trivial" case?

We clearly can't do adequate testing of the cases we really intend users to use. If there's two ways to run a single simple simulation, then that doubles the amount of testing that in principle we should do.

We have a long history of adding things like -testverlet and -nsteps and -append and -cpi, perhaps testing them only in some subcases, and users adopt them for uses that were never intended, and now you're stuck with the problem. "It's nice because I know how to use it properly" isn't a sufficient argument.

It sounds like your argument is roughly: "if there's isn't a test defining the behaviour for this subcase, then it's not a bug and doesn't need fixing". I would have thought that even if a feature has been implemented without fully defining it's behaviour, there is still room later on for doing so. Or maybe the conclusion is simply "it could be considered a bug but it's not important enough" -- ok, sure.

#14 - 03/14/2018 01:38 PM - Mark Abraham

Viveca Lindahl wrote:

Mark Abraham wrote:

Viveca Lindahl wrote:

We clearly can't do adequate testing of the cases we really intend users to use. If there's two ways to run a single simple simulation, then that doubles the amount of testing that in principle we should do.

We have a long history of adding things like -testverlet and -nsteps and -append and -cpi, perhaps testing them only in some subcases, and users adopt them for uses that were never intended, and now you're stuck with the problem. "It's nice because I know how to use it properly" isn't a sufficient argument.

It sounds like your argument is roughly: "if there's isn't a test defining the behaviour for this subcase, then it's not a bug and doesn't need fixing". I would have thought that even if a feature has been implemented without fully defining it's behaviour, there is still room later on for doing so. Or maybe the conclusion is simply "it could be considered a bug but it's not important enough" -- ok, sure.

I already said the MPI error required a fix :-). I'm objecting to explicitly supporting single-simulation multi-sim. If there was ever a design intent for -multi or -multidir it's been lost to time. The only tests of them we have run multiple simulations, and are currently good only for seeing that they don't segfault.

We could add a similar test for a single simulation, but that doubles the number of cases we would eventually have to check work also with -deffnm, -append, -cpi, and all the custom output files written by pull, awh, FEP, ED. That's not valuable work to do.

Supporting two ways of doing the same thing ("running a single simulation in a subdirectory") means people only develop and test one of them (which we all know will only be normal mdrun), and the other one eventually produces a bug because the developers forget that there's this other way of doing it.

#15 - 03/14/2018 02:02 PM - Viveca Lindahl

Mark Abraham wrote:

Viveca Lindahl wrote:

Mark Abraham wrote:

Viveca Lindahl wrote:

We clearly can't do adequate testing of the cases we really intend users to use. If there's two ways to run a single simple simulation, then that doubles the amount of testing that in principle we should do.

We have a long history of adding things like -testverlet and -nsteps and -append and -cpi, perhaps testing them only in some subcases, and users adopt them for uses that were never intended, and now you're stuck with the problem. "It's nice because I know how to use it properly" isn't a sufficient argument.

It sounds like your argument is roughly: "if there's isn't a test defining the behaviour for this subcase, then it's not a bug and doesn't need fixing". I would have thought that even if a feature has been implemented without fully defining it's behaviour, there is still room later on for doing so. Or maybe the conclusion is simply "it could be considered a bug but it's not important enough" -- ok, sure.

I already said the MPI error required a fix :-). I'm objecting to explicitly supporting single-simulation multi-sim. If there was ever a design intent for -multi or -multidir it's been lost to time. The only tests of them we have run multiple simulations, and are currently good only for seeing that they don't segfault.

We could add a similar test for a single simulation, but that doubles the number of cases we would eventually have to check work also with -deffnm, -append, -cpi, and all the custom output files written by pull, awh, FEP, ED. That's not valuable work to do.

Supporting two ways of doing the same thing ("running a single simulation in a subdirectory") means people only develop and test one of them (which we all know will only be normal mdrun), and the other one eventually produces a bug because the developers forget that there's this other way of doing it.

I haven't stated that there should be an explicit test for this and (obviously) don't claim to know how to set up the test infrastructure of gromacs. There are many combinations of run parameters that would require fixing in gromacs if they broke, even if there is not (yet) sufficient test coverage.

After all this discussion I'm quite curious of what character the fix would be -- allowing or disallowing this case?

#16 - 03/14/2018 02:31 PM - Mark Abraham

Viveca Lindahl wrote:

Mark Abraham wrote:

I haven't stated that there should be an explicit test for this and (obviously) don't claim to know how to set up the test infrastructure of gromacs. There are many combinations of run parameters that would require fixing in gromacs if they broke, even if there is not (yet) sufficient test coverage.

I know. I'm here to apply counter-pressure to "wouldn't it be cool if we added" by reminding us that this adds to our workload. Also, see <https://gerrit.gromacs.org/#/c/7676/> for some consideration of flow-on effects in how we've written and documented the code.

After all this discussion I'm quite curious of what character the fix would be -- allowing or disallowing this case?

I've been proposing on Gerrit that we disable single-sim multi-sim.

#17 - 03/14/2018 02:46 PM - Viveca Lindahl

Mark Abraham wrote:

Viveca Lindahl wrote:

Mark Abraham wrote:

I haven't stated that there should be an explicit test for this and (obviously) don't claim to know how to set up the test infrastructure of gromacs. There are many combinations of run parameters that would require fixing in gromacs if they broke, even if there is not (yet) sufficient test coverage.

I know. I'm here to apply counter-pressure to "wouldn't it be cool if we added" by reminding us that this adds to our workload. Also, see <https://gerrit.gromacs.org/#/c/7676/> for some consideration of flow-on effects in how we've written and documented the code.

I think this was more like "wouldn't it be cool if we knew what the intended behaviour was".

After all this discussion I'm quite curious of what character the fix would be -- allowing or disallowing this case?

I've been proposing on Gerrit that we disable single-sim multi-sim.

#18 - 03/14/2018 02:48 PM - Mark Abraham

Viveca Lindahl wrote:

I think this was more like "wouldn't it be cool if we knew what the intended behaviour was".

Yeah, nobody wrote that down explicitly. But the state of the code suggests that people have only ever considered the multiple simulation case, so let's be explicit about that.

#19 - 03/19/2018 02:36 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2448](#).
Uploader: Berk Hess (hess@kth.se)
Change-Id: gromacs~release-2018~l9befdb677f3c77bcc6dc0c77abc3478f0b83ba8d
Gerrit URL: <https://gerrit.gromacs.org/7691>

#20 - 03/19/2018 03:04 PM - Berk Hess

It would be nice to get this case to work so users don't need to code special cases in scripts that deal with multi-runs with variable width. But then we need to decide if we want a multisim communicator or not with nsim=1 and what modules using this should expect and do. It could be that we think this is too much effort.

#21 - 03/21/2018 12:10 PM - Mark Abraham

- *Tracker changed from Bug to Task*
- *Subject changed from mdrun -multidir flag for 1 directory only works for >= 2 ranks to should mdrun -multidir permit only one directory?*
- *Status changed from Fix uploaded to Accepted*
- *Target version deleted (2018.1)*
- *Affected version - extra info deleted (Branched from: 717fd5b8c0b47ab8bff70cfbd189e52a452db543 (1 newer local commits))*
- *Affected version deleted (2018)*

I'm open to someone designing and implementing a way that this will work so that modules that use multisim will all do something sensible for their use case, and mean we can test that it stays working (e.g. all the modules that open files to the current working directory).

It isn't a priority for my development time, because we already have a way to run a simulation in a subdirectory.

Files

out.log	3.07 KB	03/13/2018	Viveca Lindahl
md.log	13.1 KB	03/13/2018	Viveca Lindahl
topol.tpr	312 KB	03/13/2018	Viveca Lindahl