**GROMACS - Bug #2318**

**Exact checkpoint restarts are not exact in 2018 beta1**

12/01/2017 08:27 PM - Erik Lindahl

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>High</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Berk Hess</td>
</tr>
<tr>
<td>Category:</td>
<td>mdrun</td>
</tr>
<tr>
<td>Target version:</td>
<td>2018-beta2</td>
</tr>
<tr>
<td>Affected version - extra info:</td>
<td>Difficulty:    uncategorized</td>
</tr>
<tr>
<td>Affected version:</td>
<td>2016</td>
</tr>
</tbody>
</table>

**Description**

Commit 76920a4f breaks binary exact restarts, which in turn prevents debugging of other patches.

I presume this is because the dynamic pruning keeps a state across the checkpointing that is not saved. To reproduce, make two copies of the attached tpr, use the -reprod flag, run one of them to completion (2000 steps) and interrupt/continue the other one. Prior to 76920a4f, gmx check will confirm the contents is identical, but not after.

There are three obvious options:

1) Reset the pruning so no state is kept at checkpoint steps
2) Save the state to the checkpoint
3) Disable dynamic pruning on CPUs

Hopefully (1) should not be too difficult, since I assume (2) is expensive.

However... it is quite embarrassing that we have now broken exact restarts in every single beta the last ~3 versions. This really shows the importance of testing checkpointing restarts for binary identity **ANY** time we do anything related either to any state or checkpoint I/O.

**Associated revisions**

**Revision 50a7265f - 12/04/2017 09:27 AM - Berk Hess**

Only stop at nstlist steps with -reprod

Stopping mdrun with two INT or TERM signals would always happen right after the first global communication step. But this breaks exact continuation. Now with mdrun -reprod a second signal will still stop at a pair-list generation step, like with the first signal, so we can still have exact continuation.

Refs #2318

Change-Id: If65c1215d2509d60c1c5a6444769e7809288e798

**Revision 700d6f38 - 12/05/2017 01:31 AM - Berk Hess**

Fix DD exact continuation bug

With domain decomposition the local atom density, used for setting the search grid for sorting particles, was based on the local atom count including atoms/charge groups that would be moved to neighboring cells. This lead do a different density value, which in turn could result in a different number of search grid cells and thus a different summation order during a run versus when continuing a run from checkpoint, when no atoms would be moved. Now exact continuation is guaranteed for the domdec module with the mdrun -reprod option.

Refs #2318

Change-Id: I78452c7d1fc3ca6bdee63ece3795efc7e4ac467f
How do you conclude this is the dynamic pruning and change 76920a4f? Checkpointing happens at search steps, do dynamic pruning should not affect exact continuation.

I get exact continuation with a single thread-mpi rank, -reprod and no GPU. I do not get exact continuation when changing to -ntmpi 6. I also don't get reproducibility with c5923ece, one change before 76920a4f. I don't know why we don't have reproducibility with DD, but it doesn't seem to be related with the dynamic pruning.

I we think exact reproducibility is important, we should have a regression test for that.

For a water box I get exact continuation with c5923ece with RF, but not with PME. This indicates the issue might be in PME, in which case FFTW (optimization) is the likely culprit.

Now I do seem to have a run with RF (and DD) with reproducibility before dynamic pruning and not after. I'll investigate.

Note that there still seems to be a FFTW reproducibility issue.

I also get non-exact continuation with the release-2016 branch on a water box with RF and -ntmpi 6. So it already is broken there. The dynamic pruning change only seems to increase the chance of differences occurring.

This seems to be a general issue in the domain decomposition that has always been present. Because dynamic pruning increases the cut-off, it is somewhat more likely to trigger the issue.

I uploaded a fix to release-2018, but we could also consider this for release-2016.

Thanks Berk - sorry, I had to go offline to prepare a PRACE scientific case presentation yesterday evening. My only reason for blaming dynamic pruning is that it bisection to that change, and the comment didn't include anything else :-(

It would indeed be good to test continuation more, but to do it properly we should interrupt a simulation rather than quit gracefully, and I'm not sure if that is possible in the present unit tests. There is also the issue that we have a ton of different setups that should be tested. For the future, I think the only way to handle that is to require that all data structures are classes where we can test invariance by saving/restoring the state and testing the class for identity.

I still don't get binary reproducibility even for a single thread when testing on dev-purley02. I've uploaded a tpr for a shorter run, just so you have exactly the same file as I tried, and the completion settings are in the corresponding log.

Steps to repeat:

1. Copy short.tpr to cont.tpr
2. Run one gmx mdrun -v -deffnm short -nt 1 -reprod // let this finish
3. gmx mdrun -v -deffnm cont -nt 1 -reprod -cpi cont // interrupt and restart a few times
4. gmx check -f short.trr -f2 cont.trr
Update: Did more testing with single-threaded runs, and it appears earlier changes are affected there too. Will do a new bisect, but for now: Don't focus on 76920a4f.

#10 - 12/02/2017 12:48 PM - Erik Lindahl
- Subject changed from Domain decomposition breaks exact checkpoint restarts to Dynamic pruning change breaks exact checkpoint restarts for non-parallel runs
- Affected version - extra info deleted (probably any version with DD)

There seem to be several (separate?) issues.

I did the original bisect on my Mac, and that was correct, but I should have clarified that it was intentionally single-threaded to avoid MPI issues.

I just reproduced this on dev-purley02, using gcc-7.1 and compiling with "cmake -DGMX_MPI=OFF -DGMX_OPENMP=OFF -DGMX_FFT_LIBRARY=FFTPACK". The change just before the dynamic pruning works perfectly fine and produces binary exact checkpoint restarts.

This breaks with the dynamic pruning change (76920a4f), and should not have anything to do with domain decomposition since there is no parallelism whatsoever involved.

Let's focus on that first, and then worry about the reproducibility issues caused by MPI/OpenMP/FFTW.

#11 - 12/02/2017 02:13 PM - Erik Lindahl
For earlier changes running single-threaded, the issue seems to be FFTW. I'll try to look into that tonight!

#12 - 12/02/2017 11:19 PM - Berk Hess
I fixed #2016, which was vsite related. I think that bug should not have affected reproducibility, but I'm not 100% sure.

I think we only need to guarantee exact continuation with -reprod with clean termination. But we now only ever terminate at nstlist steps, IIRC. So we should always get exact continuation (unless there are bugs, of course).

#13 - 12/02/2017 11:44 PM - Erik Lindahl
- Status changed from Fix uploaded to In Progress

Again: This whole bug report was NOT related to domain decomposition or any parallel run, so let's leave that aside for now.

The whole point of binary exact continuation was to enable restarts no matter how the run was interrupted. Later we have extended things so we try and write an extra checkpoint if we catch a signal, but that is merely a convenience feature, not a reason why extra states thingies can be kept - any checkpoint continuation should be exact with -reprod, unless there are fundamental algorithm reasons that make it impossible.

Have you been able to confirm the bug appearing with this patch for the example tpr file when only using a single thread and FFTPACK?

#14 - 12/03/2017 11:32 AM - Berk Hess
I tried your recipe on Haswell with the vsite fix, FFTW and a single thread and there I can't reproduce it. Later today I will try to use exactly your setup on the purley machine.

#15 - 12/03/2017 08:34 PM - Berk Hess
Tried on purley01 no with fftpack, gcc7.1, but with tMPI and OpenMP (which shouldn't matter with -nt 1), interrupted the run 6 times, but always get identical results.
But what do you do to interrupt the run?
I press ctrl-C once, which makes the run stop at an nstlist step. If you press ctrl-C twice, it might stop at a non-nstlist step and you will then not get exact continuation.

#16 - 12/03/2017 09:15 PM - Berk Hess
I just realized we should disable quitting and checkpointing before the next nstlist step (when pressing ctrl-C twice) with -reprod.

#17 - 12/03/2017 10:23 PM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #2318.
Uploader: Berk Hess (hess@kth.se)
Change-Id: gromacs~release-2018~If65c1215d2509d60c1c5a6444769e7809288e798
Gerrit URL: https://gerrit.gromacs.org/7270

#18 - 12/04/2017 08:34 AM - Berk Hess
Erik, is this still an issue, or was it the double signal issue?

#19 - 12/04/2017 01:06 PM - Berk Hess
- Subject changed from Dynamic pruning change breaks exact checkpoint restarts for non-parallel runs to Exact checkpoint restarts are not exact in 2018 beta1
- Status changed from In Progress to Resolved

#20 - 12/05/2017 03:33 AM - Mark Abraham
- Target version changed from 2018 to 2018-beta2

#21 - 12/05/2017 02:29 PM - Erik Lindahl
- Status changed from Resolved to Closed

Berk & I already talked offline, but just for the record: The cause here was likely the much longer neighbor list intervals introduced with the dynamic pruning patch (or something else enabled there), which made exact restarts fail. It is now fixed in release-2018.

It is not entirely trivial to introduce a unit test for this, since we would need signals or process interruption, but one way we could do it is to enable a debug-build only feature where we check for specific environment variables, and just call exit/abort at a specific step to emulate a crash or signal being caught.

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>run.tpr</td>
<td>354 KB</td>
<td>12/01/2017</td>
<td>Erik Lindahl</td>
</tr>
<tr>
<td>short.tpr</td>
<td>10.4 MB</td>
<td>12/02/2017</td>
<td>Erik Lindahl</td>
</tr>
<tr>
<td>short.log</td>
<td>24.7 KB</td>
<td>12/02/2017</td>
<td>Erik Lindahl</td>
</tr>
</tbody>
</table>