## GROMACS - Bug #1965

### Crash of mdrun with strange error messages

05/19/2016 02:37 PM - Semen Yesylevskyy

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>Affected version:</td>
<td>5.1.2</td>
</tr>
</tbody>
</table>

### Affected version -

**Description**

mdrun crashes with the following output. It happens randomly for different number of CPUs from 16 to 64. After restart it works nicely few hours and crashes again.

```
gmx_mpi:3879 terminated with signal 11 at PC=2b03ef1415a3 SP=7fff7e7acf60. Backtrace:
/Softs/lumiere/gromacs/5.1.2/lib/libgromacs_mpi.so.1(+0x1e35a3)[0x2b03ef1415a3]
/Softs/lumiere/gromacs/5.1.2/lib/libgromacs_mpi.so.1(_Z14spread_on_gridP9gmx_pme_tP14pme_atomcomm_tP10pmegrids_tiiPii+0x0f0f)[0x2b03ef140faf]
/Softs/easybuild/work/install/software/imkl/11.1.2.144/lib/intel64/libiomp5.so(__kmp_invoke_microtask+0x93)[0x2b03f0722233]
```

```
gmx_mpi:3881 terminated with signal 11 at PC=2b2c07e66110 SP=7fff50abbe50. Backtrace:
/Softs/lumiere/gromacs/5.1.2/lib/libgromacs_mpi.so.1(_Z8do_pairsiiPKiPK9t_iparamsPA3_KfPA3_fS8_PK5t_pbcPK7t_graphPfSF_PK9t_mdatomsPK10t_forcerecP17gmx_grppairener_tPi+0x710)[0x2b2c07e66110]
/Softs/lumiere/gromacs/5.1.2/lib/libgromacs_mpi.so.1(+0x15811c)[0x2b2c07e6a11c]
/Softs/lumiere/gromacs/5.1.2/lib/libgromacs_mpi.so.1(calc_listed+0xbd0)[0x2b2c07e6ae70]
/Softs/easybuild/work/install/software/imkl/11.1.2.144/lib/intel64/libiomp5.so(__kmp_invoke_microtask+0x93)[0x2b2c094f6233]
```

```
gmx_mpi:3877 terminated with signal 11 at PC=2b8d1f2e6110 SP=7fffce874d0. Backtrace:
/Softs/lumiere/gromacs/5.1.2/lib/libgromacs_mpi.so.1(_Z8do_pairsiiPKiPK9t_iparamsPA3_KfPA3_fS8_PK5t_pbcPK7t_graphPfSF_PK9t_mdatomsPK10t_forcerecP17gmx_grppairener_tPi+0x710)[0x2b8d1f2e6110]
/Softs/lumiere/gromacs/5.1.2/lib/libgromacs_mpi.so.1(+0x15811c)[0x2b8d1f2ea11c]
/Softs/lumiere/gromacs/5.1.2/lib/libgromacs_mpi.so.1(calc_listed+0xbd0)[0x2b8d1f2eae70]
/Softs/easybuild/work/install/software/imkl/11.1.2.144/lib/intel64/libiomp5.so(__kmp_invoke_microtask+0x93)[0x2b8d20956233]
```

```
gmx_mpi:3882 terminated with signal 11 at PC=2ac1a54a9110 SP=7fff62995150. Backtrace:
/Softs/lumiere/gromacs/5.1.2/lib/libgromacs_mpi.so.1(_Z8do_pairsiiPKiPK9t_iparamsPA3_KfPA3_fS8_PK5t_pbcPK7t_graphPfSF_PK9t_mdatomsPK10t_forcerecP17gmx_grppairener_tPi+0x710)[0x2ac1a54a9110]
/Softs/lumiere/gromacs/5.1.2/lib/libgromacs_mpi.so.1(+0x15811c)[0x2ac1a54ad11c]
/Softs/lumiere/gromacs/5.1.2/lib/libgromacs_mpi.so.1(calc_listed+0xbd0)[0x2ac1a54ade70]
/Softs/easybuild/work/install/software/imkl/11.1.2.144/lib/intel64/libiomp5.so(__kmp_invoke_microtask+0x93)[0x2ac1a6b19233]
```

### Related issues:

- Related to GROMACS - Bug #1958: Segfault with non-interacting atoms with Verl... Closed
- Related to GROMACS - Bug #2023: Segfault again with non-interacting atoms and... Closed

### History

**#1 - 05/20/2016 09:19 AM - Erik Lindahl**

- Status changed from New to Feedback wanted

This bug report contains very little information.

Please provide the raw input files (see instructions [http://redmine.gromacs.org/projects/gromacs](http://redmine.gromacs.org/projects/gromacs)) and the log file from the run.

1) Start by using the "-reprod" flag. Does this cause the simulation to crash at the same timestep every time?
2) If yes, can you get the simulation to crash quickly by restarting from the last checkpoint? If so, upload the checkpoint file too.

Cheers,

Erik

---

#2 - 05/20/2016 10:36 AM - Semen Yesylevskyy
- File bug1.zip added

Here are the input files, log and latest cpt. The crash seems to be at random time steps but I'll double check this.

---

#3 - 05/20/2016 02:30 PM - Erik Lindahl

A common reason for "random" crashes is that the best versions of load balancing and FFT algorithm selection are non-deterministic (it depends on the load on the machine).

For GPUs it's difficult (they are always a bit non-deterministic), but for CPU runs the "-reprod" flag to mdrun makes a pretty good job of making the run entirely deterministic, and then it should crash at the same time step every time.

---

#4 - 05/20/2016 06:06 PM - Semen Yesylevskyy

I've tried to run on other cluster and surprisingly it is much more stable with the same version 5.1.2!
I asked our admins to recompile without MKL to see if this is the reason. I'll write back when we have the results of this test.

---

#5 - 05/22/2016 10:57 AM - Semen Yesylevskyy

After the tests on four different machines I'm a bit lost. In one of the machines, which uses Intel MKL and MPI it crashes almost instantly. On shared-memory 4-cores workstation without MPI it works nicely. On the third machine with MPI but without MKL it also crashes but less frequently. Finally on the last one (the huge cluster) it works just fine on more than 200 cores.

Such inconsistency in behavior is very bad thing. What information should I provide to help to isolate the problem?

---

#6 - 05/23/2016 01:16 AM - Erik Lindahl

A normal CPU run (no GPUs) with the -reprod flag and FFTW/FFTPACK rather than MKL should always crash at the same timestep. Please try that on the "third" machine, ideally with a fairly small number for the -cpt argument; if you checkpoint every minute, you should be able to restart and have it crash again after a minute.

---

#7 - 05/26/2016 09:41 AM - Semen Yesylevskyy

Ok, the reason for this bug is apparently SIMD. Works Ok if recompiled without it.

---

#8 - 07/28/2016 03:15 PM - Mark Abraham
- File repro.tgz added

This is again a problem of missing exclusions between atoms that have interactions and end up with identical coordinates, even though the simulation is using the group scheme (so it will not have been fixed by the changes to the verlet scheme in 5.1.3). In release-5-1 (and presumably earlier) the problem is first noticed in the PME code at the step following the problematic one, but in release-2016 branch, the new check for infinite potential energy stops the simulation sooner. It has nothing to do with MKL, etc. I think that disabling SIMD should have had no effect, but clearly the problem case arises only rarely.

After a long simulation from Semen's inputs, I found a repro case (attached). Run gmx mdrun -cpi and there is a SIGFPE in the first step in nb_kernel_ElecEw_VdwLJ_GeomP1P1_VF_* (reproduced in release-5-1 HEAD with AVX_128_FMA and AVX_256; does not reproduce with GMX_SIMD=None; reproduces also in release-2016 as above). To get the SIGFPE to throw, one needs to e.g. hack runner.cpp to call 
gmx_feenableexcept() also for the group scheme.

I haven't worked out which atoms are overlapping (though f[2056] is the one that goes NaN), but extracting the positions from the .cpt can find that out if we need to. Presuming this system is similar to that of #1958, it seems likely that it's a pair of non-interacting SW "wall" atoms that end up interacting. I think a much more reliable topology setup is to have a [moleculeroype] that is all the atoms in the wall, and explicitly excluding all internal interactions. Now they can't end up in pair lists and this problem cannot arise. Previously, forces got computed, but I presume the zero masses meant that only updates of zero size occurred. However zero * NaN is a NaN, so it seems like Semen needs exclusions.

We could also consider a fix to the group scheme like that done for the Verlet scheme at #1958.

We could also consider a grompp warning if a user has mobile atoms that have no interaction parameters (Semen's SW-SW have none, if I've read correctly). I can't see what grompp did in this case, because Semen's tarball hasn't got an .mdp file.

---

#9 - 07/28/2016 03:21 PM - Mark Abraham
- Related to Bug #1958: Segfault with non-interacting atoms with Verlet scheme added

---

#10 - 07/30/2016 09:31 AM - Semen Yesylevskyy
I think a much more reliable topology setup is to have a [moleculetype] that is all the atoms in the wall, and explicitly excluding all internal interactions.

Thank you for suggestion, I'll try this out. However, changing topology like this is just a workaround, which makes things more complicated (one needs to generate another itp for dummies every time their number changes). The real fix would be much better.

We could also consider a fix to the group scheme like that done for the Verlet scheme at #1958.

Seems to be the best idea for consistency. If something works with Verlet one expects it to work with group as well.

We could also consider a grompp warning if a user has mobile atoms that have no interaction parameters (Semen's SW-SW have none, if I've read correctly). I can't see what grompp did in this case, because Semen's tarball hasn't got an .mdp file.

Sorry, I just forgot an mdp and can't find it now. SW-SW are not interacting for sure. Grompp warning only seems necessary if such corner case will continue to crash the simulation. If it will be fixed such warning didn't tell anything useful for the user.

#11 - 08/02/2016 01:11 AM - Mark Abraham
Semen Yesylevskyy wrote:

I think a much more reliable topology setup is to have a [moleculetype] that is all the atoms in the wall, and explicitly excluding all internal interactions.

Thank you for suggestion, I'll try this out. However, changing topology like this is just a workaround, which makes things more complicated (one needs to generate another itp for dummies every time their number changes). The real fix would be much better.

In a very real sense the proper fix is that particle pairs that should never have a short-range interaction should have exclusions between them - that's what exclusions are for. The general case is tricky for grompp to handle automatically. For example, typical water models have no VDW on hydrogen, relying on the oxygen VDW to prevent molecules from coming close enough at typical temperatures and time-step sizes. Your case is potentially recognizable as problematic because there were single-particle moleculetypes that were not in frozen groups that had zero parameters with other atoms not in frozen groups, so grompp could issue a diagnostic. But at some point a user wanting to express "this set of particles can move, but will never interact" will need to coordinate their own exclusions, because we can't write code that recognizes in advance all patterns that might want a diagnostic (or automatic generation of exclusions).

We could also consider a grompp warning if a user has mobile atoms that have no interaction parameters (Semen's SW-SW have none, if I've read correctly). I can't see what grompp did in this case, because Semen's tarball hasn't got an .mdp file.

Sorry, I just forgot an mdp and can't find it now. SW-SW are not interacting for sure. Grompp warning only seems necessary if such corner case will continue to crash the simulation. If it will be fixed such warning didn't tell anything useful for the user.

Yeah, but the implementations are very different, and the group scheme is about to be removed, so the value of such a fix is low.

We could also consider a grompp warning if a user has mobile atoms that have no interaction parameters (Semen's SW-SW have none, if I've read correctly). I can't see what grompp did in this case, because Semen's tarball hasn't got an .mdp file.

Sorry, I just forgot an mdp and can't find it now. SW-SW are not interacting for sure. Grompp warning only seems necessary if such corner case will continue to crash the simulation. If it will be fixed such warning didn't tell anything useful for the user.

It's not just about whether it will crash. Mobile atoms whose types have zero interaction parameters are intrinsically unusual. Grompp can't tell whether the error was the failure to declare exclusions, or the failure to declare parameters, or a misnamed atom type. IMO the latter two possibilities are rather more consistent with single-particle moleculetypes, but they're OK - even normal - in a water molecule. Even if I'm wrong, the default behaviour of grompp should not be to generate such a tpr. The user might judge where the problem lies, and that in practice exclusions aren't needed, and perhaps suppress a warning. If grompp had had such a warning, then you might have read that, and had reason to consult on e.g. gmx-users about how best to proceed, and perhaps had your simulations working well a while ago, without needing code changes.

#12 - 08/05/2016 12:39 AM - Mark Abraham
- Related to Bug #2023: Segfault again with non-interacting atoms and verlet cutoff added

#13 - 01/30/2017 03:06 PM - Szilárd Páll
Was this also fixed in fce7c4c4c?

#14 - 12/30/2017 11:01 PM - Erik Lindahl
- Status changed from Feedback wanted to Closed

02/23/2020
Yes, this should have been fixed in the verlet kernels, and we no longer touch the group kernels since they are deprecated.

<table>
<thead>
<tr>
<th>Files</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>topol.tpr</td>
<td>3.44 MB</td>
<td>05/19/2016</td>
<td>Semen Yesylevskyy</td>
</tr>
<tr>
<td>bug1.zip</td>
<td>9.43 MB</td>
<td>05/20/2016</td>
<td>Semen Yesylevskyy</td>
</tr>
<tr>
<td>repro.tgz</td>
<td>4.15 MB</td>
<td>07/28/2016</td>
<td>Mark Abraham</td>
</tr>
</tbody>
</table>