Hi,

we have a serious problem with GROMACS 4.6.5 for GPU, which produces incorrect output. Comparing to the CPU version, calculated energies are totally different (see attached files from the official GROMACS gpu benchmark), what basically results in a fast collapse to a random coil structure. The system we are running on is 2 x 8-core 2.5 GHz Intel Xeon E5-2450 + 2 x NVIDIA Tesla K40m.

Could you help us with that?

Marcin

------
Computational Chemistry Group
UvA, Amsterdam

Associated revisions
Revision 92a2eb1a - 09/08/2014 09:36 AM - Berk Hess
Check for implicit solvent + Verlet scheme
Fixed #1570
Change-ID: l8734c2dc99d3bc3e0a79ae043d8685444f03b495

History
#1 - 07/30/2014 02:15 PM - Justin Lemkul
The CPU run reports version 4.6.4, while the GPU version reports 4.6.5; at least do a comparison with a consistent version. Many bugs get fixed between minor releases. Better yet, try again with 4.6.6 or 5.0; it is much more effective to troubleshoot the current release than it is an old version whose issues may have already been resolved.

#2 - 07/30/2014 10:18 PM - Marcin Nowosielski
In addition to 4.6.5 we have tested 4.5.7 and the latest 5.0. All of them give the wrong result.

Focusing on the 5.0, there are two major issues:
1) Incorrect Polarization and Nonpolar Sol. energies (here I mean orders of magnitude)
2) Presence of pressure (either positive or negative) with pcoupl = No

To be honest this is at least disturbing... . Don't get me wrong, I appreciate your work a lot, and use GROMACS all the time.

Nevertheless, how can it be that all three versions are giving such huge errors (i.e. they do not work)? Are we so unlucky with the system configuration?

Regards: Marcin

#3 - 07/30/2014 10:30 PM - Justin Lemkul
Well, to be fair, we advertise that implicit + GPU does not work:

http://www.gromacs.org/Developer_Zone/Roadmap/GROMACS_4.6#Features currently not supported by the new GPU and SSE kernels

The previous use of implicit solvent on GPU was totally reliant upon the OpenMM interface, which is no longer supported. It would be nice if the implicit code could be made more robust and work with GPU, but at present there are no developers with time to do it. There have been a number of discussions about this, but time is limited and implicit + GPU is a very low priority. We should probably just issue a fatal error in mdrun if anyone tries to do this.

#4 - 07/31/2014 05:05 AM - Roland Schulz
- Status changed from Rejected to In Progress

I agree we should add a fatal error for implicit+GPU. In fact I think it is important enough to not close the bug until we have done so.

#5 - 07/31/2014 05:05 AM - Roland Schulz
- Status changed from In Progress to Accepted

#6 - 07/31/2014 02:05 PM - Thomas Geenen

The gromacs runs that gave the wrong results where performed on our system.

from the release notes of the 4.6 version I understand that the 4.5 version using openmm should be able to compute the correct results. Marcin ran a model also with this version of gromacs and found that these results are also incorrect.

we build the 4.5 version of gromacs with these settings

```
export OPENMM_ROOT_DIR=/hpc/sw/gromacs-${ver}-gpu-sp/openmm
cmake -DCMAKE_INSTALL_PREFIX=$installdir
    -DGMX_MPI=no
    -DGMX_X11=OFF
    -DGMX_DEFAULT_SUFFIX=OFF
    -DGMX_THREADS=no
    -DGMX_OPENMMP=ON -DGMX_THREADS=OFF
    -DGMX_FFT_LIBRARY=FFTW3
    -DGMX_GPU=ON -DCUDA_TOOLKIT_ROOT_DIR=/hpc/sw/cuda/5.5/
    -DFFTW3F_INCLUDE_DIR=$SURFSARA_FFTW3_INCLUDE
    -DFFTW3F_LIBRARIES=$SURFSARA_FFTW3_LIB/libfftw3f.so
    ../$dir | tee ../log.cmake
```

should we expect to get the correct results from such a build with implicit calculations like the GPU benchmark examples impl 1nm and 2nm?

best
Thomas

#7 - 07/31/2014 04:09 PM - Justin Lemkul

The benchmarks were created using very old versions of everything, CUDA 3.0 and OpenMM 2.0, IIRC. Any results you get will almost certainly be sensitive to changes. The combination of Gromacs 4.5 + OpenMM 2.0 had lots of bugs and missing features (on both the Gromacs and OpenMM sides), so I would expect that only the same combination of software versions might generate the same exact results. How "wrong" are your results from 4.5? Side-by-side comparisons are needed here.

#8 - 08/03/2014 12:15 PM - Marcin Nowosielski

Hi,

Together with Thomas we have tried a whole bunch of things.

The setup has been taken from the official benchmark: dhfr-impl-2nm.bench (with amber96ff)

The results

1)

4.5 on CPU:

<table>
<thead>
<tr>
<th>Step</th>
<th>Time</th>
<th>Lambda</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

Energies (kJ/mol)
-7.27832e+04

2) 4.5 for GPU:

Energies (kJ/mol)

<table>
<thead>
<tr>
<th>Potential</th>
<th>Kinetic En.</th>
<th>Total Energy</th>
<th>Temperature</th>
<th>Constr. rmsd</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.86574e+04</td>
<td>6.36140e+03</td>
<td>-1.22960e+04</td>
<td>3.09693e+02</td>
<td>2.10775e-06</td>
</tr>
</tbody>
</table>

Error (potential) ca. 20%

3) 4.6 for GPU

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.74632e+03</td>
<td>3.15965e+03</td>
<td>4.17922e+01</td>
<td>-6.47831e+04</td>
<td>8.76577e+03</td>
</tr>
<tr>
<td>2.46211e+03</td>
<td>2.90393e+04</td>
<td>-5.15137e+03</td>
<td>-3.66485e+04</td>
<td>-6.13680e+04</td>
</tr>
<tr>
<td>1.08079e+02</td>
<td>-6.12599e+04</td>
<td>5.26483e+00</td>
<td>2.99131e+01</td>
<td>7.35318e-06</td>
</tr>
</tbody>
</table>

Error (potential) ca. 160%

4) 5.0 for GPU

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.51520e+03</td>
<td>4.10883e+03</td>
<td>5.02242e+02</td>
<td>-6.74749e+04</td>
<td>8.43644e+03</td>
</tr>
<tr>
<td>3.02422e+03</td>
<td>3.23932e+04</td>
<td>5.49163e+03</td>
<td>-7.27832e+04</td>
<td>-4.38033e+03</td>
</tr>
<tr>
<td>6.13076e+03</td>
<td>-6.66524e+04</td>
<td>2.98646e+02</td>
<td>-1.23727e+01</td>
<td>2.36228e-05</td>
</tr>
</tbody>
</table>

Error (potential) ca. 208%

Taking into account that a folded state is only slightly more stable than an unfolded (<100kJ/mol) none of the versions actually works (confirmed by analyzing trajectories)

Now on the good side - with the infinite cut-offs the 4.5 ver. works well

CPU Potential: -2.43293e+04
GPU Potential: -2.43574e+04

There is only one small note added to the release of 4.6v. which suggests that implicit solvent calculations may not work correctly and none in case of the 5.0. In case of 4.5 all configurations were suppose to work.
Reading the manuals (one small table at page 21 for ver.5.0), and looking at the benchmarks, one have an impression that it works. Especially that in the case of expl. solvent the speedup is marginal and may not justify costs of moving to GPU at all.

Please be more specific, that is a great software after all.

Getting back to bug hunting in my own code.

Cheers:
Marcin

#9 - 09/09/2014 02:38 PM - Gerrit Code Review Bot
Gerrit received a related patchset '1' for Issue #1570.
Uploader: Berk Hess (hess@kth.se)
Change-Id: I8734c2dc99d3bc3e0a79aa043d86854446f3b495
Gerrit URL: https://gerrit.gromacs.org/4021

#10 - 09/11/2014 12:14 PM - Berk Hess
- Category set to preprocessing (pdb2gmx,grompp)
- Status changed from Accepted to Resolved
The incorrect GB results with GPU were due to a missing check for GB with the Verlet scheme in grompp. A grompp check has been added for 5.0.2. We would like to have GB supported with the new SIMD and GPU kernels, but we currently don't have time to do this (but we do need it for 6.0 where the group cut-off scheme will no longer be present).

Were there more issues here?

#11 - 09/11/2014 05:47 PM - Marcin Nowosielski

That is awesome.
No more issues for the moment and I am keeping my fingers crossed for new releases.

Best:
Marcin

#12 - 10/01/2014 04:29 PM - Mark Abraham

- Status changed from Resolved to Closed
- Target version deleted (5.0.2)

<table>
<thead>
<tr>
<th>Files</th>
<th>Size</th>
<th>Date</th>
<th>Username</th>
</tr>
</thead>
<tbody>
<tr>
<td>dhfr.tar.gz</td>
<td>5.18 MB</td>
<td>07/30/2014</td>
<td>Marcin Nowosielski</td>
</tr>
</tbody>
</table>