GROMACS - Bug #1822

mdrun writes broken energy group values to .edr file

09/15/2015 11:37 AM - Mark Abraham

Status:	Closed		
Priority:	Low		
Assignee:	Erik Lindahl		
Category:	mdrun		
Target version:	2018		
Affected version - extra info:	4.6 through 5.1	**Difficulty:**	uncategorized
Affected version:	5.1.1		

Description

With 2 energy groups, mdrun -nb cpu and mdrun -nb gpu writes .edr files such that gmxcheck -e cpu-run -e2 gpu-run gives

There are 39 terms to compare in the energy files

```
Coulomb (SR)      step   0:      -11637.2,  step   0:     -15106.6
Potential         step   0:      -6509.07,  step   0:      -9978.4
Total Energy      step   0:      -6501.01,  step   0:     -9970.33
Coul-SR:URE-URE   step   0:       9361.08,  step   0:     -15106.6
LJ-SR:URE-URE     step   0:        336.99,  step   0:      3585.22
Coul-SR:URE-SOL   step   0:      -3929.27,  step   0:            0
LJ-SR:URE-SOL     step   0:      -168.613,  step   0:            0
Coul-SR:SOL-SOL   step   0:        -17069,  step   0:            0
LJ-SR:SOL-SOL     step   0:       3416.84,  step   0:            0
Coulomb (SR)      step   1:      -11657.9,  step   1:     -15127.2
```

Even with a single energy group, I get

```
Coulomb (SR)      step   0:      -11637.2,  step   0:     -15106.6
Potential         step   0:      -6509.07,  step   0:     -9978.41
Total Energy      step   0:      -6501.01,  step   0:     -9970.34
Coul-SR:URE-URE   step   0:       9361.08,  step   0:     -15106.6
LJ-SR:URE-URE     step   0:        336.99,  step   0:      3585.22
Coul-SR:URE-rest  step   0:      -3929.27,  step   0:            0
LJ-SR:URE-rest    step   0:      -168.613,  step   0:            0
Coul-SR:rest-rest  step   0:        -17069,  step   0:            0
LJ-SR:rest-rest   step   0:       3416.84,  step   0:            0
Coulomb (SR)      step   1:      -11657.9,  step   1:     -15127.2
```

Tarball with repro materials and output attached.

The GPU .log file does say "NOTE: With GPUs, reporting energy group contributions is not supported". (In #1293 it was suggested we move/add such a comment near the end of the .log file. #1727 also misunderstood how to use the code)

Since energy groups are not supported on GPUs, we should not write an .edr file with energy groups, so that users cannot erroneously use the incorrect data they contain. If we're unwilling to do that, then perhaps energy-analysis tools should have a check for "all the fields zero except the first".

Frankly, there's something to be said for only writing group-wise contributions during a rerun. (Our code is likely not agile enough to be able to call the energy-group kernels only on energy-output steps, so even on the CPU the small overhead of energy groups is being paid every MD step.) This would be slightly easier to do once we've removed the group scheme.

Related issues:

Related to GROMACS - Bug #1293: energygrps issue with cuda runs	**Closed**	**06/28/2013**
Related to GROMACS - Bug #1727: g_energy computes zero energy values with GPU	**Closed**	**05/06/2015**

Associated revisions

Revision 4a4dc78e - 01/03/2018 01:26 PM - Erik Lindahl

Don't allow multiple energy groups for GPU runs

Exit with a fatal error instead of only warning, since the
latter leads to writing data for energy groups that
is incorrect to the energy file.

Fixes #1822.

Change-Id: I34ccb10bba6d6e1350283e34ebc908c6f830baab

History

#1 - 09/15/2015 11:42 AM - Mark Abraham

- Related to Bug #1293: energygrps issue with cuda runs added

#2 - 09/15/2015 11:42 AM - Mark Abraham

- Related to Bug #1727: g_energy computes zero energy values with GPU added

#3 - 09/18/2015 11:58 AM - Berk Hess

The nbnxn CPU kernels only occurs overhead from energy groups at nstcalcenergy steps, and even then it's small.
Removing the energy group contributions in a GPU run makes the edr file non-compatible with a cpu run, which can be inconvenient for analysis.
The main issue here is that many people used energy groups protein and water. With the group scheme this was free, since we had water-water and water-other kernels anyhow. So many people might still have energy groups selected. In addition, many people think that these energies are meaningful, whereas for 99% of the applications they are not. But if you actually have a meaningful use for them, you need to sample them very frequently, since they are extremely noisy, so you'd rather do it during the run.

But something looks very wrong in the comparison you pasted. Is that from the same tpr? The total energies should match then.

#4 - 09/18/2015 01:45 PM - Berk Hess

I can not reproduce the incorrect total Coulomb energy with 5.0, 5.1 or master, but I can with 4.6.8. I assume this is due to an incorrect/missing shift (coulomb-modifier=potential-shift) of the plain-Coulomb potential.
Note that one should never use a plain cut-off for Coulomb interactions, unless all pairs fall within the cut-off.

#5 - 05/08/2016 04:17 PM - Erik Lindahl

4.6 is no longer supported, so let's check if we can reproduce this, and otherwise abandon it.

#6 - 06/01/2016 04:47 PM - Mark Abraham

It does look like maybe something was wrong in my original comparison, but aspects of this issue remain.

When we run on GPUs, we write a set of energy-group-pair fields that are actually the total, and a bunch of zeroes, e.g. re-using the above inputs

```
$ source ../r2016/build-cmake-gcc-release/install/bin/GMXRC
$ gmx mdrun -s two-energy-groups.tpr -nsteps 1 -nb cpu -deffnm cpu-two-energy-groups
$ source ../r2016/build-cmake-gcc-gpu-release/install/bin/GMXRC
$ gmx mdrun -s two-energy-groups.tpr -nsteps 1 -nb gpu -deffnm gpu-two-energy-groups
$ gmx check -e cpu-two-energy-groups -e2 gpu-two-energy-groups.edr
GROMACS:      gmx check, version 2016-beta2-dev-20160531-13e0dc7-unknown
Executable:   /home/marklocal/redmines/redmine-1822/r2016/build-cmake-gcc-gpu-release/install/bin/gmx
Data prefix:  /home/marklocal/redmines/redmine-1822/r2016/build-cmake-gcc-gpu-release/install
Command line:
  gmx check -e cpu-two-energy-groups -e2 gpu-two-energy-groups.edr

Opened cpu-two-energy-groups.edr as single precision energy file
Opened gpu-two-energy-groups.edr as single precision energy file
^MReading energy frame      0 time    0.000         ^MReading energy frame      0 time    0.000         ^MRead
ing energy frame      1 time    0.001         ^MReading energy frame      1 time    0.001         ^MLast energ
y frame read 1 time    0.001         ^MLast energy frame read 1 time    0.001
GROMACS reminds you: "It takes money to make money, they say" (Lou Reed)

comparing energy file cpu-two-energy-groups.edr and gpu-two-energy-groups.edr

There are 39 terms in the energy files

There are 39 terms to compare in the energy files

Coul-SR:URE-URE  step   0:       9361.08,  step   0:     -11637.2
LJ-SR:URE-URE    step   0:        336.99,  step   0:      3585.22
Coul-SR:URE-SOL  step   0:      -3929.27,  step   0:            0
LJ-SR:URE-SOL    step   0:      -168.613,  step   0:            0
Coul-SR:SOL-SOL  step   0:      -17069.1,  step   0:            0
```

```
LJ-SR:SOL-SOL     step   0:        3416.84,  step   0:             0
Coul-SR:URE-URE   step   1:        9352.03,  step   1:      -11657.9
LJ-SR:URE-URE     step   1:        335.338,  step   1:       3582.94
Coul-SR:URE-SOL   step   1:       -3933.18,  step   1:             0
LJ-SR:URE-SOL     step   1:       -168.767,  step   1:             0
Coul-SR:SOL-SOL   step   1:       -17076.7,  step   1:             0
LJ-SR:SOL-SOL     step   1:        3416.37,  step   1:             0
```

All the code is doing is dumping Coulomb (SR) into the first energy-group-pair field, which is simply wrong. Among other possibilities is that someone runs a Verlet-scheme .tpr on a GPU and gets a garbage result from analysing the non-zero values in the first energy-group-pair fields that are not actually the energy-group-pair values they expect to get from last time they ran this .mdp/.tpr on their old CPU-only cluster.

I don't mind if we don't implement energy groups for running on GPUs (because performance would suck and probably many users would not choose well if the choice was available to them), but we need to
a) refuse to run such .tpr files with GPUs available, unless the user chooses -nb cpu, or
b) run energy steps on the CPU only (similar to what we do for rerun right now but probably harder to code), or
c) run energy-only kernels on the CPU in parallel with the GPU force kernels, or
d) continue to issue a note, and not write any energy-group fields (so at least analyses always fail hard).

#7 - 12/31/2017 05:03 PM - Erik Lindahl

No action on this for a year, so it's time to at least get rid of the bad behavior of silently dumping energies in the wrong place. We should not deliberately do incorrect stuff for convenience.

If the user asks for energy groups but that is not possible to compute, the obvious solution is to refuse to run unless "-nb cpu" was selected.

#8 - 12/31/2017 05:30 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #1822.
Uploader: Erik Lindahl (erik.lindahl@gmail.com)
Change-Id: gromacs~release-2018~I34ccb10bba6d6e1350283e34ebc908c6f830baab
Gerrit URL: https://gerrit.gromacs.org/7407

#9 - 12/31/2017 05:30 PM - Erik Lindahl

- Status changed from New to Fix uploaded

#10 - 01/01/2018 11:10 AM - Mark Abraham

- Assignee set to Erik Lindahl

- Target version set to 2018

#11 - 01/03/2018 02:02 PM - Erik Lindahl

- Status changed from Fix uploaded to Resolved

Applied in changeset 4a4dc78e0c059ed662ae29331fd4a6c2ad6278a2.

#12 - 01/03/2018 03:47 PM - Erik Lindahl

- Status changed from Resolved to Closed

Files

energy-groups-issue.tgz	423 KB	09/15/2015	Mark Abraham