GROMACS - Bug #1299

Strange results on GPU

07/06/2013 06:25 PM - Cara Kreck

Status:	Closed		
Priority:	Normal		
Assignee:	Szilárd Páll		
Category:	mdrun		
Target version:	4.6.x		
Affected version - extra info:		**Difficulty:**	uncategorized
Affected version:	4.6		

Description

From http://gromacs.5086.x6.nabble.com/Inconsistent-results-between-3-3-3-and-4-6-with-various-set-up-options-tp5009584p5009585.html

Mark said I should open an issue to see if the bad performance, atypical RSMD, and zero potentials (at step 0) I get when running 4.6 on GPU are reproducible.

		Temperature		Potential E.		Pressure		
System name	Details	Average	RMSD	Average	RMSD	Average	RMSD	ns/day
3.3.3 c_md	RF nst5 group	306.0	1.4	-439029	466	0.998	125	9.7
4.6 c_md	RF nst5 group	303.9	1.4	-440455	461	0.0570	126	36.9
4.6 c_vv	RF nst5 verlet	303.0	1.2	-438718	478	1.96	134	16.5
4.6 g_md	RF nst20 verlet	303.0	1.4	-439359	3193	566	1139	19.8
4.6 g_vv	RF nst20 verlet	303.0	1.2	-438635	3048	34.3	405	18.6
4.6 c_pme	md nst5 group	303.0	1.4	-436138	461	0.135	125	25.5
4.6 g_pme	md nst40 verlet	303.0	1.4	-431621	463	416	1016	33.6

Where c_md indicates CPU only and md integrator, g_vv indicates GPU and md-vv integrator, etc. Verlet & group refer to cut-off scheme and nst# refers to nstlist frequency which was automatically changed by gromacs. I found very similar results (and run times) for the GPU runs when -nb was set to gpu or gpu_cpu.

Step 0:

System	LJ (SR)	Coulomb (SR)	Potential	Kinetic En.	Total Energy	Temperature	Pressure (bar)
3.3.3 c_md	1.80072E+04	-4.30514E+05	-4.38922E+05	6.14932E+04	-3.77429E+05	3.06083E+02	1.53992E+02
4.6 c_md	1.80072E+04	-4.30515E+05	-4.38922E+05	6.20484E+04	-3.76874E+05	3.08847E+02	1.56245E+02
4.6 c_vv	1.15784E+04	-4.83639E+05	-4.37388E+05	6.14748E+04	-3.75913E+05	3.05992E+02	-1.40193E+03
4.6 g_md	0.00000E+00	0.00000E+00	3.46728E+04	6.14991E+04	9.61719E+04	3.06113E+02	-1.70102E+04
4.6 g_vv	0.00000E+00	0.00000E+00	3.46728E+04	6.14748E+04	9.61476E+04	3.05992E+02	-1.85758E+04
4.6 c_pme	1.30512E+04	-3.37973E+05	-4.35821E+05	6.14989E+04	-3.74322E+05	3.06112E+02	4.50028E+02
4.6 g_pme	1.76523E+04	-4.89006E+05	-4.31207E+05	6.14990E+04	-3.69708E+05	3.06112E+02	4.37951E+02

History

#1 - 07/06/2013 09:28 PM - Mark Abraham

- Description updated

#2 - 07/06/2013 10:10 PM - Mark Abraham

- File dopc_tip4p2005_g_vv.log.gz added

- Status changed from New to Accepted

- Assignee changed from Berk Hess to Szilárd Páll

- Target version set to 4.6.x

- Affected version - extra info set to post-4.6.3 git also affected

Thanks.

I can see in *g_vv.log that you're using 4.6, and 82% of the run time is spent ostensibly waiting for the GPU to be available. I was able to reproduce the waiting with your .tpr on my machine (CUDA 5, GTX660Ti) with the latest git version, so there is something we should look at. (log file attached) But with the *g_pme.log there was no excess wait for you or me.

There's also a note that with GPUs group-wise energy decomposition is not supported. If you need such a thing, doing mdrun -rerun (without the GPU) is encouraged. Total throughput could well be higher, because mdrun can now run optimally on all the steps, and you probably can't get meaningful statistics from most of the steps, anyway.

I got sensible step-0 energies with your .tpr (with both git and 4.6 versions), which suggests there's a code issue that only shows up in some configurations. I'd guess the garbage energy data causes the apparently large fluctuations you saw.

All up, I'd guess that the code that manages GPU waiting has at least one bug with energy groups, causing either the excess waiting with non-PME, or inaccurate reporting of the energies on your machine (but not mine). I suspect the forces are unaffected, else things would have simply exploded, but Szilard is the best judge of that.

#3 - 07/06/2013 10:11 PM - Mark Abraham

If Cara can set up a g_vv run without using energy groups, that would be useful information, please. And might be pleasantly suprising ;-)

#4 - 07/07/2013 02:22 PM - Cara Kreck

- File dopc_tip4p2005_g_vv_neg.tpr.gz added

I've attached the .tpr without energy groups.

I was running on the single node of a GPU cluster containing two 6-core Intel Xeon X5650 CPUs and a NVIDIA Tesla C2075 GPU. Could there be a problem with the way gromacs has been installed?

Thanks.

#5 - 07/08/2013 09:20 AM - Cara Kreck

- File dopc_tip4p2005_g_vv_nce1.tpr.gz added

OK, I've run the no energy groups simulation along with a nstcalcenergy=1 (with energy groups) simulation. No energy groups hasn't made any difference for me. With nstcalcenergy=1 the step 0 LJ and Coulomb are back to normal and therefore the PE RMSD is fixed. Just ignoring the dodgy step 0 values also brings the large PE RMSD down to match the rest. However the initial pressure is still negative (now -1.40188e+03) and its average value and RMSD have gotten even worse!

#6 - 07/08/2013 03:07 PM - Cara Kreck

- File neg_nce1_logs.tar.gz added

I'm uploading the log files of the finished no energy groups (neg) and nstcalcenergy=1 (nce1) simulations.

g_vv_nce1 - runtime 16.9 ns/day

```
Energy                      Average   Err.Est.       RMSD  Tot-Drift
-------------------------------------------------------------------------------
Potential                   -438522         14    486.041   -82.6535  (kJ/mol)
Temperature                 302.999    8.9e-05    1.31495 -0.000332746  (K)
Pressure                   -97.3348        1.4    801.616   -8.25374  (bar)
```

g_vv_neg - runtime 18.6 ns/day

```
Energy                      Average   Err.Est.       RMSD  Tot-Drift
-------------------------------------------------------------------------------
Potential                   -438578         66     3047.6   -387.254  (kJ/mol)
Temperature                 303.017     0.0078    1.21502  0.0224628  (K)
Pressure                    37.9142        1.5     416.31    7.94282  (bar)
```

#7 - 07/09/2013 12:23 AM - Mark Abraham

OK thanks. People are away and holidaying for the next week or more, but I agree this should work!

#8 - 07/10/2013 11:16 AM - Szilárd Páll

Ack, this indeed looks fishy. I am not available these days, I will look at the problem, but probably won't have time before mid next week or so.

#9 - 07/10/2013 11:24 AM - Szilárd Páll

Regarding the performance, note, that these are reaction field runs where there is very little CPU force computation workload for the GPU computation to overlap with (see the "Wait GPU" row time in the log) and currently there is only one possibility to split non-bonded workload between CPU and GPU: using the -nb gpu_cpu option *with domain-decomposition* which will result in the execution of the non-local workload on the CPU. However, to achieve this, you typically want to "overload" the GPU with multiple ranks which is only possible ATM with MPI, e.g.:

```
mpirun -np 2 mdrun_mpi -gpu_id 00 -nb gpu_cpu
```

#10 - 08/01/2013 12:01 AM - Szilárd Páll

Mark Abraham wrote:

> Thanks.
>
> I can see in *g_vv.log that you're using 4.6, and 82% of the run time is spent ostensibly waiting for the GPU to be available. I was able to reproduce the waiting with your .tpr on my machine (CUDA 5, GTX660Ti) with the latest git version, so there is something we should look at. (log file attached) But with the *g_pme.log there was no excess wait for you or me.

All those are perfectly normal. The waiting happens because Cara is using 2x 6-core Westmere CPUs with a single Fermi GPU. This is a very imbalanced setup especially considering the rather large cut-offs used. Additionally, the mdrun's auto-nstlist selection which switches to nstlist=20 just makes things worse.

> I got sensible step-0 energies with your .tpr (with both git and 4.6 versions), which suggests there's a code issue that only shows up in some configurations. I'd guess the garbage energy data causes the apparently large fluctuations you saw.

Me too running on the same hardware as Cara - although much newer compilers.

> All up, I'd guess that the code that manages GPU waiting has at least one bug with energy groups,

I don't think there is such a bug in the current code, the only thing I can think of is the polling wait which was used in this case with the 4.6.0 tests (as ECC was on). As the CUDA driver is 5.0, the current code will not switch to polling.

Cara, could you please re-run some short tests with either 4.6.3 or current git. It would be great if you could try using a newer gcc.

#11 - 08/01/2013 12:02 AM - Szilárd Páll

@Mark: "post-4.6.3 git also affected" was referring to the performance, step 0 broken energies, or something else?

#12 - 08/02/2013 08:31 AM - Berk Hess

For me 4.6.3 gives non-zero energies at step 0 and a pressure of 1.4e-3 at step 0, which seems reasonable to me.
As Szilard asked, please rerun with 4.6.3.

PS You should upgrade to 4.6.3 anyhow, first release versions (4.6 in this case) tend to have a few bugs which are quickly fixed in subsequent bug-fix releases.

#13 - 08/02/2013 04:25 PM - Mark Abraham

- Status changed from Accepted to Feedback wanted

- Affected version - extra info deleted (post-4.6.3 git also affected)

Szilárd Páll wrote:

> @Mark: "post-4.6.3 git also affected" was referring to the performance, step 0 broken energies, or something else?

Sorry, I was misinterpreting "Wait GPU local" as "waiting for the GPU to be ready to accept work" rather than "waiting for GPU to finish work while there is nothing for the CPU to do." I now see there's a separate entry for "Launch GPU." So the 63% reported in my log file attached above is waste, but not waste of CPU **and** GPU :-). So, my run of Cara's g_vv with git post-4.6.3 seems fine, given how it was run.

So, I think there is no issue to address here, unless Cara continues to observe problems with 4.6.3.

#14 - 09/18/2013 09:53 PM - Szilárd Páll

Last call for an update from Cara, otherwise I'll close the issue.

#15 - 10/16/2013 03:36 PM - Mark Abraham

- Status changed from Feedback wanted to Closed

Files

redmine.tar.gz	6.07 MB	07/06/2013	Cara Kreck
dopc_tip4p2005_g_vv.log.gz	8.56 KB	07/06/2013	Mark Abraham
dopc_tip4p2005_g_vv_neg.tpr.gz	629 KB	07/07/2013	Cara Kreck
dopc_tip4p2005_g_vv_nce1.tpr.gz	629 KB	07/08/2013	Cara Kreck
neg_nce1_logs.tar.gz	289 KB	07/08/2013	Cara Kreck