

GROMACS - Task #1925

remove concept of unilateral global communication

03/18/2016 06:57 PM - Mark Abraham

Status:	In Progress
Priority:	Normal
Assignee:	
Category:	mdrun
Target version:	2020
Difficulty:	uncategorized
Description	
<p>In GROMACS 5.1, on CPUs the Verlet scheme chooses <code>nstlist=25</code>, which combines poorly with reasonable-seeming <code>nsttcouple=20</code> into triggering <code>compute_globals</code> every <code>lcd(nstlist,nsttcouple)=5</code> steps. On GPUs, that's probably <code>lcd(50,20)=10</code>. Obviously this scales over multiple nodes much worse than people expect.</p> <p>The cross dependency of <code>nstlist</code> and <code>nstglobalcomm</code> is an historical artefact of the group scheme. Perhaps for GROMACS 2016, when using the Verlet scheme, we can remove <code>nstlist</code> from the <code>lcd()</code> in <code>check_nstglobalcomm()</code>?</p> <p>Various points of the the MD loop should be refactored to avoid expressing themselves in terms of <code>nstglobalcomm</code>, e.g. we should have code express that it needs the half-step KE this step so that it can compute the on-step KE later on, rather than use <code>do_per_step(step-1, nstglobalcomm)</code> etc. The semantics of <code>bGStat</code> also must be cleared up, but probably these happen together.</p> <p><code>gmX mdrun -gcom</code> should just disappear.</p>	
Related issues:	
Related to GROMACS - Task #2569: announce deprecations in GROMACS 2019	Closed

Associated revisions

Revision cf2d8336 - 10/13/2018 09:53 PM - Mark Abraham

Deprecate various functionality in GROMACS 2019

Published a deprecation policy.

Updated the release notes to refer also to previously deprecated features.

Announced intent to change some functionality:

- `gmX mdrun -membed` options (but not feature)
- `gmX mdrun -rerun` option (but not feature)
- integrator `.mdp` field will contain only integrators
- `gmX do_dssp` to be replaced by `gmX dssp`
- `gmX trjconv` and friends to be split and rewritten

List of newly deprecated functionality:

- conversion of aromatic rings to virtual sites
- `gmX mdrun -table` options (but not feature)
- `gmX mdrun -gcom` option and feature
- `gmX mdrun -nsteps` option and feature
- `gmX mdrun -nsteps -resetstep -resetway` moved to a `gmX benchmark` tool
- `gmX mdrun -confout` removed

Also updated release notes for functionality removed in GROMACS 2019.

Refs #2495, #1781

Fixes #2569, #1925

Change-Id: I1d00859d0f15409a472984f5a65347a50c71ad17

Revision 786e0e87 - 04/16/2019 01:27 PM - Mark Abraham

Removed `mdrun -gcom`

This was previously deprecated, and is now removed to make the behaviour of mdrun simpler to understand and implement.

Renamed a function whose job was previously not to check a thing, and is now clearly to compute something

Noted several TODOs to clean up behaviour related to nstcomm.

Refs #1925

Change-Id: I0b3a803fb209148e865957f796c871caef2f1fea

History

#1 - 03/20/2016 08:38 AM - Berk Hess

I agree that mdrun -gcom is very inconvenient from a coding point of view. But without it fixing 30% time in global communication is very tedious. Then we should have statistics output of which things have triggered global communication with what frequency.

#2 - 05/27/2016 12:04 AM - Mark Abraham

Berk Hess wrote:

I agree that mdrun -gcom is very inconvenient from a coding point of view. But without it fixing 30% time in global communication is very tedious.

But -gcom was around for years, (I think) never got a mention in the reference manual, nor background information in mdrun -h, until I added some words to the user guide. So either the .mdp-based solution isn't tedious, or -gcom isn't being used to solve a problem people realize that they have, or people don't realize there's a problem they might solve. Either way, I think the "it's useful" argument needs some evidence from somewhere.

Then we should have statistics output of which things have triggered global communication with what frequency.

If our code is too complex for us to analyze, then more stuff has to go. :-) But once <https://gerrit.gromacs.org/#/q/topic:md-loop-cleanup> are finalized, rerun can be implemented outside of do_md() (my patch was ready six months ago, but is still off gerrit waiting for the cleanup to finalize). With the group scheme also gone, then we can probably start to analyze these things properly and hopefully implement things in a way that can be tested, easily documented and easily used.

#3 - 09/13/2018 01:47 AM - Mark Abraham

- Related to Task #2569: announce deprecations in GROMACS 2019 added

#4 - 10/05/2018 11:49 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#1925](#).
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~master~I1d00859d0f15409a472984f5a65347a50c71ad17
Gerrit URL: <https://gerrit.gromacs.org/8488>

#5 - 10/13/2018 10:15 PM - Mark Abraham

- Status changed from New to Resolved

Applied in changeset [cf2d8336856fc925be2fa82127706ad0ca43520b](#).

#6 - 12/17/2018 01:43 PM - Paul Bauer

- Status changed from Resolved to Closed

#7 - 04/16/2019 10:42 AM - Mark Abraham

- Status changed from Closed to In Progress

#8 - 04/16/2019 10:42 AM - Mark Abraham

- Target version changed from future to 2020

#9 - 07/15/2019 06:42 PM - Szilárd Páll

Mark Abraham wrote:

Berk Hess wrote:

I agree that mdrun -gcom is very inconvenient from a coding point of view. But without it fixing 30% time in global communication is very tedious.

But -gcom was around for years, (I think) never got a mention in the reference manual, nor background information in mdrun -h, until I added some words to the user guide.

I think that's not the case:

```
$ ~/projects/gromacs/gromacs-4.6/build_gcc4.8_cuda6.5/src/kernel/mdrun -h 2>&1 | grep -C1 gcom
```

```
The option -gcom can be used to only do global communication every n steps.
This can improve performance for highly parallel simulations where this
```

```
--
and/or barostat the temperature and/or pressure will also only be updated
every -gcom steps. By default it is set to the minimum of nstcalcenergy and
nstlist.
```

```
--
-dds      real    0.8      Minimum allowed dlb scaling of the DD cell size
-gcom     int     -1       Global communication frequency
-nb      enum    auto     Calculate non-bonded interactions on: auto, cpu,
```

So either the .mdp-based solution isn't tedious, or -gcom isn't being used to solve a problem people realize that they have, or people don't realize there's a problem they might solve. Either way, I think the "it's useful" argument needs some evidence from somewhere.

It was in the mdrun -h output and some have used it.

I've just encountered the second user who ran into this issue when testing master. This user (a Cray application perf engineer) asked for documentation/advise on this removal. AFAICT, we don't have any, do we?

I agree with Berk, users would be better off with some advise how to achieve the same as -gcom gave previously. Sure, we can tell people to just pick the "right" nst* mdp options, but I have doubts that will be sufficient for many.