GROMACS - Bug #1121
slow MD with sd integrator and GPU / verlet
01/16/2013 07:59 AM - Floris Buelens

| Status:    | Closed          |
| Priority:  | High            |
| Assignee:  | Berk Hess       |
| Category:  | mdrun           |
| Target version: | 4.6          |
| Affected version - extra info: | 4.6          |
| Affected version: |               |
| Difficulty: | uncategorized  |

Description
I'm seeing MD simulation running a lot slower with the sd integrator than with md - ca. 10 vs. 30 ns/day for my 47000 atom system. I found no documented indication that this should be the case. Log files are attached. Wall time seems to be accumulating in Update and Rest, adding up to >60% of total. Using the group cutoff scheme, there seems to be a non-negligible slowdown with sd, also associated with extra wall time in Update and Rest, but this is modest compared to the impact with the verlet scheme.

System: Xeon E5-1620, 1x GTX 680

Timings:
cpu (ns/day)
sd / verlet: 6
sd / group: 10
md / verlet: 9.2
md / group: 11.4
gpu (ns/day)
sd / verlet: 11
md / verlet: 29.8

Associated revisions
Revision d120c370 - 01/17/2013 06:42 PM - Berk Hess
fixed SD+BD integration slowing down with OpenMP threads

The SD and BD integrators would integrate on all OpenMP threads, making the integration much slower instead of faster.
It is not clear if the results could be affected by this bug.
Fixes #1121
Change-Id: lea4283e0470b72f6f927cb49503ac91d65025647

Revision 98b6873d - 06/20/2013 05:03 PM - Berk Hess
fixed SD and BD integrator OpenMP performance

SD and BD integrator always integrated single threaded.
Really fixes #1121
Change-Id: l2217c40e9c188c7cd57801e413750035c6488f56

History
#1 - 01/16/2013 01:11 PM - Berk Hess
- Status changed from New to Closed

There is no bug here.
(Unfortunately we made the non-bondeds on the GPU so fast, that with SD we spend more time in the integration and constraints. The absolute time differences between md and sd should be similar with group and verlet, although settle in verlet is slightly slower because no charge groups are used, You can switch to the sd1 integrator, if that's accurate enough for you. That's about as fast as md.)
PS note that mdp_opt.html says at SD:
An accurate leap-frog stochastic dynamics integrator.
Four Gaussian random number are required
per integration step per degree of freedom. With constraints,
coordinates needs to be constrained twice per integration step.
Depending on the computational cost of the force calculation,
this can take a significant part of the simulation time.

And gives the sd1 suggestion as well.

thanks for the reply - I'm not convinced that explains the numbers though. Ignoring the GPU for now, here are some more timings using '-nb cpu' only:

<table>
<thead>
<tr>
<th>ns/day</th>
<th>Update walltime (s)</th>
<th>Rest walltime (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group cutoff scheme:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>md: 11.6</td>
<td>2.03</td>
<td>0.34</td>
</tr>
<tr>
<td>sd: 10.9</td>
<td>3.95</td>
<td>3.42</td>
</tr>
<tr>
<td>sd1: 11.7</td>
<td>2.25</td>
<td>0.33</td>
</tr>
<tr>
<td>Verlet cutoff scheme:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>md: 9.4</td>
<td>2.30</td>
<td>1.31</td>
</tr>
<tr>
<td>sd: 6.2</td>
<td>27.53</td>
<td>22.12</td>
</tr>
<tr>
<td>sd1: 7.9</td>
<td>19.12</td>
<td>1.33</td>
</tr>
</tbody>
</table>

... so with the group scheme, Update takes ca. 2x longer when you switch from md to sd
while with the verlet scheme, Update takes ca. 12x longer when you switch from md to sd

Again without GPU, switching from md to sd costs me 6% with the group scheme, and 34% with verlet.

sd1 is indeed about as fast as md with the group scheme, with verlet the hit is still significant (16%). On GPU, my sd1 timing is 18.3 ns/day, against 11 ns/day for sd and 29.8 ns/day for md, so still 38% slower than md.

You're right, I overlooked the rest time.
I see now that the SD (and BD) update is not OpenMP threaded.
A (my) comment in the code says we need to find a way to generate random seeds for different threads.
I'll think of one.

OK. Erik will be around in the morning, so I expect we can brainstorm something for 4.6. In a private branch, I once fixed seed generation for some parallel context, but I suspect my solution will be either not applicable or over-weight.

You can checkout of pull the fix from the download link on:
https://gerrit.gromacs.org/#/c/2073/

I'm not 100% sure that this bug can't affect results, so use the patched version for production.

Arrgh, clicked the wrong link while trying to fetch the tprs. Can somebody re-upload the sd-vv.tpr please?
I didn't download it. Taking any system with Verlet scheme and changing the integrator to sd will show it.

- Target version set to 4.6

- Status changed from Feedback wanted to Closed