The nbnxn-ljpme-geometric test fails with GMX_SIMD=SSE2 in single precision. It looks like some NaN-s show up in the energies after step 0, see log file attached.

Tested on AMD K10 and Intel SB with gcc 4.7, 4.8, and clang 3.3.

Associated revisions

Revision 73ae4791 - 07/28/2014 10:26 PM - Christian Wennberg

Fix overflow in LJ-PME nbnxn kernels

The SIMD exp function in the LJ-PME nbnxn kernels could overflow for pair distances far beyond the cut-off. Added a mask to avoid this.

Fixes #1552

Change-Id: Id87710f3815b341f53a69df0a2990d0bb4edfa74

History

#1 - 07/03/2014 08:07 PM - Christian Wennberg

Have tracked the issue a bit now, and it seems like the half step energies (if that is what tcstat->ekinh[j][m] contains) in sum_ekin becomes NaN when running on 5,6 or 7 MPI-threads.

#2 - 07/04/2014 07:16 PM - Szilárd Páll

Christian Wennberg wrote:

Have tracked the issue a bit now, and it seems like the half step energies (if that is what tcstat->ekinh[j][m] contains) in sum_ekin becomes NaN when running on 5,6 or 7 MPI-threads.

Funky. Only with thread-MPI, not with MPI?
I get the same behaviour for both MPI and Thread-MPI

Tracked the issue a bit further, the strange kinetic-energy values seems to come from the forces obtained in the nbnxn-kernels.
I get NaN for some of the forces that come out of nbnxn_atomdata_add_nbat_f_to_f_part (called from nbnxn_atomdata_add_nbat_f_to_f in sim_util.c).

nbnxn_atomdata_add_nbat_f_to_f is just doing reduction and format conversion, so unless the bug is there (which is quite unlikely, I think), source of NaNs is the non-bonded kernel - which is also what the SSE2-only nature of the bug suggests.

I don't think it is that clear. nbnxn_atomdata_add_nbat_f_to_f does do some explicit SIMD in nbnxn_atomdata_reduce_reals_simd. Plus there's lots of lovely pointer indirection and tasty assumptions there will be 2 or 4 of this and that :-P

Ideas for detecting which code paths might be wrong

- define env var GMX_USE_TREEREDEUCE - uses different implementation of force reduction intended for use on MIC, but if it just blows up then we should ask Roland about it
- configure with GMX_SIMD=Reference - the defaults match SSE2 (IIRC, but diff the log files to be sure), so the result ought to be binary reproducible, and has the benefit that it does not call nbnxn_atomdata_reduce_reals_simd, and does not call a SIMD kernel. If this works, then we can hack the SIMD version to call nbnxn_atomdata_reduce_reals instead.
- run double on AVX, because that is 4-wide SIMD also and might be illuminating
Christian - any update on this?

Using GMX_SIMD=reference or AVX_256 as suggested by Szilard will not result in an error.

I've only been able to recreate the problem when using SSE2, but the results is not consistent over the number of openmp-threads that is running, so I guess it might be related to some decomposition-scheme over the tasks that are running.

Me and Szilard had a quick look at the coefficients (charges and/or LJ-values) that was passed into the kernels some time ago, and found that some of them was NaN which I guess should never happen.

That is worrying indeed, since it might be an indication of something overwriting memory or doing other bad things, it is just that we only notice it on SSE2 for some reason, so this could affect default LJPM runs.

Is anybody working actively on it?

Not that I'm aware of. I can dig into it a bit more in detail the upcoming week.

Gerrit received a related patchset '1' for Issue #1552.
Uploader: Roland Schulz (roland@rschulz.eu)
Change-Id: Ib1b3afc525706f4b171564fcaf06ebf3b2be3122
Gerrit URL: https://gerrit.gromacs.org/3829

With ACX, 6 tmpi ranks, and the patch it shows that gmx_simd_exp_r overflows because cr2_S0 is too large (nbnxn_kernel_simd_2xnn_inner.h:787), because jx_S, jy_S, and jz_S contain wrong numbers (-107, -214, -14338 - exactly not rounded). The overflow only happens with nbnxn-ljpme-geometric and only with tmpi-ranks>4 and thus this should be the same problem. Valgrind/Msan/Tsan doesn't detect any errors.

These odd values come from NBAT_FAR_AWAY in nbnxn_atomdata.c. But I don't understand how the filler particles are suppose to work and why they seem to cause a problem with nbnxn-ljpme-geometric.

Gerrit received a related patchset '1' for Issue #1552.
Uploader: Christian Wennberg (christian.wennberg@sciifelab.se)
Change-Id: Id87710f3815b341f53a69df0a2990d0bb4edfa74
Gerrit URL: https://gerrit.gromacs.org/3830

- Status changed from New to Fix uploaded
There's a lesson to learn: more flexible Jenkins tests are needed (verification or nightly). This bug slipped in because AFAIK we have hard-coded test setups with only 1-3 ranks (mostly with GPUs) and 1-2 threads.

[ I tried to create a 5.0 nightly testing project and start setting up longer/more extensive tests, but bumped into some issues and had to put it on ice. Can anybody help/continue where I left off? ]

#16 - 08/10/2014 08:20 PM - Roland Schulz

- Status changed from Fix uploaded to Closed

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>md.log</td>
<td>16.9 KB</td>
<td>07/03/2014</td>
<td>Szilárd Páll</td>
</tr>
</tbody>
</table>