GROMACS - Bug #1576
lincs_update_atoms_ind operates on coordinates which are NaN
08/11/2014 11:44 PM - Roland Schulz

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Low</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Berk Hess</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>Affected version - extra info:</td>
<td></td>
</tr>
<tr>
<td>Affected version:</td>
<td>git master</td>
</tr>
</tbody>
</table>

**Description**

With fp-exceptions (https://gerrit.gromacs.org/#/c/3829/8) enabled nbnxn_pme_order5 with GPU, OpenMP and MPI crashes in lincs_update_atoms_ind on bs_nix1204

Run as:
mpirun -np 2 gmx mdrun -ntomp 2 -reprod yes

Gdb output:

```plaintext
#0 0x000007f0afaf7d65f in lincs_update_atoms_ind (ncons=197, ind=0x7f0adc2189c0, bla=0x2240340, prefac=499.999969, fac=0x22472e0,
    r=0x222431c0, invmass=0x22322b0, x=0x219d910) at /mnt/workspace/Gromacs_Gerrit_master/072f4ab0/gromacs/src/gromacs/mdlib/clincs.c:297
297             x[j][0] += tmp0*im2;
(gdb) print j
$1 = 1926
(gdb) bt
#0 0x000007f0afaf7d65f in lincs_update_atoms_ind (ncons=197, ind=0x7f0adc2189c0, bla=0x2240340, prefac=499.999969, fac=0x22472e0,
    r=0x222431c0, invmass=0x22322b0, x=0x219d910) at /mnt/workspace/Gromacs_Gerrit_master/072f4ab0/gromacs/src/gromacs/mdlib/clincs.c:297
#1 0x000007f0afaf7da52 in lincs_update_atoms (li=0x216a0b0, th=1, prefac=499.999969, fac=0x22472e0,
    r=0x222431c0, invmass=0x22322b0, x=0x219d910) at /mnt/workspace/Gromacs_Gerrit_master/072f4ab0/gromacs/src/gromacs/mdlib/clincs.c:340
#2 0x000007f0afaf7f1c7 in do_lincs (x=0x2197a00, xp=0x226b410, box=0x20905b0, pbc=0x7fff2c2732c0, lincsd=0x216a0b0, th=1,
    invmass=0x22322b0, cr=0x1405c10, bCalcLambda=0, wangle=30, warn=0x7fff2c27113c, invdt=499.999969, v=0x219d910, bCalcVir=0,
    vir_r_m_dr=0x216a23c) at /mnt/workspace/Gromacs_Gerrit_master/072f4ab0/gromacs/src/gromacs/mdlib/clincs.c:684
#3 0x000007f0afaf830ab in constrain_lincs_omp_fn.0 (.omp_data_i=0x7fff2c271070)
    at /mnt/workspace/Gromacs_Gerrit_master/072f4ab0/gromacs/src/gromacs/mdlib/clincs.c:1565
#4 0x000007f0af66ee3aa in ?? () from /usr/lib/x86_64-linux-gnu/libgomp.so.1
#5 0x000007f0af66ee9a in start_thread (arg=0x7f0aea848700) at pthread_create.c:308
#6 0x000007f0af8a373d in clone () at ./.sysdeps/unix/sysv/linux/x86_64/clone.S:112
#7 0x0000000000000000 in ?? ()
(gdb) print x[j][0]
$2 = -nan(0x2f1d06)
```

It looks to me like it operates on coordinates which are non-local and not correctly communicated. But I don't really understand the indexing. Might very well have no effect on the result but even then we probably don't want to compute on NaNs.

**Associated revisions**

Revision 839b4f6c - 08/24/2014 07:20 PM - Berk Hess

Initialize unused velocity constraint components
With domain decomposition, velocity components for communicated atoms could be uninitialized. These components were never used, but this could lead to valgrind warnings and floating point exceptions.

Fixes #1576.

Change-Id: Ic673680699578ddf62c2d240342138cbc16490c8

History

#1 - 08/12/2014 12:27 AM - Roland Schulz
Actually the fact that disabling gpu (-nb cpu) fixes the problem, suggests that it isn't simply always operating on some non-local atoms.

#2 - 08/12/2014 09:04 AM - Berk Hess
I can't reproduce this on my machine with GPU and either AVX or SSE4. Are there NaN's in the log file as well? Is the output incorrect with exceptions turned on, but ok without or with git master?

#3 - 08/14/2014 09:52 AM - Roland Schulz
The compiler version (I tested GCC 4.5 and 4.4) and SIMD (SSE4.1 or AVX) doesn't seem to matter. What does matter is the CUDA version. I can only reproduce it with Cuda 4.1. Both with 4.0 and 4.2 I don't get an exception. No numbers are printed to the log file before the exception occurs. If I disable the exception (but keep the rest of the commit) then I don't see any NaN in the log file and it passes the regression test. I did this on the Jenkins bs-nix1204 node.

#4 - 08/15/2014 12:01 AM - Roland Schulz
I just ran it again with 4.0 and got an exception in nbnxn_vsite. I cannot reproduce it (rerunning it 50 times). That isn't that unsurprising given that the CUDA kernels aren't reproducible. I assume the rounding in 4.1 makes it likely to occur but it is something which can also (but unlikely) happen with other CUDA versions (or even maybe without GPU).

#5 - 08/18/2014 06:17 PM - Roland Schulz
Berk, have you been able to reproduce it with my added information? Anything else I can help with this?

#6 - 08/19/2014 03:45 PM - Berk Hess
The only not deterministic part in the CUDA kernels is the force reduction. I don't see how a difference in the final few bits of the forces could lead to a difference of (not) crashing.
Could you run mdrun -pforce, with a value of e.g. 5000? Then we can see if there are large or NaN forces.

#7 - 08/20/2014 01:34 AM - Roland Schulz
If I add

```c
if (i==1872) fprintf(stderr, "%f ", x[i][2]);
```

to clincs.c at line 296 than I get as output e.g.:

```
0.969000 0.969191 0.969255 0.969253 255.657227
```

This is both with and without my exception commit. The 5th value varies widely (including 0). It seems the 5th value printed is uninitialized and thus causes under some condition the NaN exception when I use my exception commit.

How can I check whether the value of i is valid here? Is any i valid which is either a local atom or an atom copied by dd_move_x_constraints? I noticed i is larger than nat_tot. But nat_tot doesn't include the constraints, right? How do I know which atoms are copied. dd_move_x_constraints is incomprehensible to me. Or are you intentionally also modifying x[i] in lincs_update_atoms_ind for some i where x[i] is not initialized (e.g. because you don't care what happens to the non-local atom)?

My suspicion is that the x[1872] is set by dd_move_x_constraints to some uninitialized value.

I don't get any large forces (larger than 3500) wither either Cuda 4.1 or 4.2.

On bs_nix1204 /home/jenkins/testing/gromacs.ref/gcc45_cuda41 has a build of 2403e51acb7 but with the exception disabled. 
/home/jenkins/testing/gromacs/gcc45_cuda41 is the build of 2403e51acb7 without any modifications (exception enabled).

#8 - 08/21/2014 09:45 AM - Berk Hess
All indices should be valid, they always correspond to home atoms from other nodes.

dd_get_constraint_range gets you the range of the communicated atoms for constraints. But I am pretty sure that will not be the issue. That's very well tested code and if the issue would be there, there should be no correlation at all with the CUDA version or any non-bonded interaction issue. My suspicion is that too large forces are returned and the first overflow occurs in LINCS.

Did you run with -pforce?
I tried to log into bs-nix1, but my password doesn't get accepted. How should I log into bs_nix1204?

I now got access to this machine.

valgrind indeed says that this element in lincs is uninitialized.

But I see now that these are velocities, not coordinates! Unfortunately you hadn't noticed this yet, that would have saved all of us a lot of time!

The lincs atom update only needs to work on local coordinates, but doing so would require a conditional, which is probably why I didn't do this. Operating on the non-local coordinates is fine, since they are never used after this. But the non-local velocities might be unset, but the array is allocated for enough elements. So this is harmless, but it would be better to avoid it.

Conditionals is probably not a good idea. We could zero the non-local velocities when the coordinate communication is done.

Maybe we should also rename x to xp. Because it is really confusing to use x for velocities. Odd that you got a valgrind error. I ran it also with valgrind I didn't see any. That would have helped too.

Yes zero the non-local velocities sounds good. If you think it might have a performance impact we could do that inside #ifdef NDEBUG.

Gerrit received a related patchset ‘1’ for Issue #1576.

Uploader: Berk Hess (hess@kth.se)
Change-Id: Ic673680699578ddf62c2d2d20432138cbc16490c8
Gerrit URL: https://gerrit.gromacs.org/3928

Yes zero the non-local velocities sounds good. If you think it might have a performance impact we could do that inside #ifdef NDEBUG.

#10 - 08/21/2014 04:14 PM - Roland Schulz

Maybe we should also rename x to xp. Because it is really confusing to use x for velocities. Odd that you got a valgrind error. I ran it also with valgrind I didn't see any. That would have helped too.

Yes zero the non-local velocities sounds good. If you think it might have a performance impact we could do that inside #ifdef NDEBUG.

#11 - 08/22/2014 12:40 PM - Gerrit Code Review Bot

Gerrit received a related patchset ‘1’ for Issue #1576.

Uploader: Berk Hess (hess@kth.se)
Change-Id: Ic673680699578ddf62c2d2d20432138cbc16490c8
Gerrit URL: https://gerrit.gromacs.org/3928

#12 - 08/25/2014 06:16 AM - Roland Schulz

- Status changed from In Progress to Closed