

GROMACS - Bug #2337

Intel compiler ewald test is failing in double

12/07/2017 04:27 AM - Roland Schulz

Status: Closed	
Priority: Normal	
Assignee:	
Category:	
Target version:	
Affected version - extra info:	Difficulty: uncategorized
Affected version: 2018-beta1	
Description	
Both 17.4 and 18.1	
FAILED TESTS:	
- SaneInput1/PmeBSplineModuliCorrectnessTest.ReproducesValues/21, where GetParam() = (12-byte object <39-00 00-00 54-00 00-00 1D-00 00-00>, 7, 4-byte object <01-00 00-00>)	
../src/testutils/refdata.cpp:918: Failure	
In item: /Y/[41]	
Actual: 0.0057077485220020426	
Reference: 0.0057077485219992341	
Difference: 2.80852e-15 (3238 double-prec. ULPs, rel. 4.92e-13)	
Tolerance: abs. 2.66454e-15, 12 ULPs	
Google Test trace:	
../src/gromacs/ewald/tests/pmebsplinetest.cpp:95: Testing B-spline moduli creation (P3M) for PME order 7, grid size 57 84 29	
../src/testutils/refdata.cpp:918: Failure	
In item: /Y/[43]	
Actual: 0.0057077485220020426	
Reference: 0.0057077485219992341	
Difference: 2.80852e-15 (3238 double-prec. ULPs, rel. 4.92e-13)	
Tolerance: abs. 2.66454e-15, 12 ULPs	
Google Test trace:	
../src/gromacs/ewald/tests/pmebsplinetest.cpp:95: Testing B-spline moduli creation (P3M) for PME order 7, grid size 57 84 29	
- DifferentEwaldCoeffLJ/PmeSolveTest.ReproducesOutputs/7, where GetParam() = ({ 7, 0, 0, 0, 4.1, 0, 3.5, 2, 12.2 }, 12-byte object <09-00 00-00 07-00 00-00 17-00 00-00>, { (12-byte object <00-00 00-00 04-00 00-00 00-00 00-00>, 16-byte object <00-00 00-00 00-00 00-00 00-00 00-40 33-33 D3-3F>), (12-byte object <04-00 00-00 02-00 00-00 07-00 00-00>, 16-byte object <00-00 00-C0 1E-85 2B-40 00-00 00-00 00-00 44-C0>), (12-byte object <00-00 00-00 06-00 00-00 07-00 00-00>, 16-byte object <00-00 00-C0 CC-CC 0C-40 00-00 00-00 00-00 00-00>), (12-byte object <02-00 00-00 05-00 00-00 0A-00 00-00>, 16-byte object <00-00 00-C0 CC-CC 0C-40 00-00 00-C0 CC-4C 25-40>) }, 1.2, 2, 2.35, 4-byte object <01-00 00-00>)	
../src/testutils/refdata.cpp:918: Failure	
In item: /Virial/Cell 0 0	
Actual: -381.37902032167176	
Reference: -381.37902032167398	
Difference: 2.21689e-12 (39 double-prec. ULPs, rel. 5.81e-15)	
Tolerance: abs. 1.77636e-12, 8 ULPs	
Google Test trace:	
../src/gromacs/ewald/tests/pmesolvetest.cpp:146: Testing solving (Lennard-Jones, YZX, with energy/virial) with CPU for PME grid size 9 7 23, Ewald coefficients 2 2.35	

Associated revisions

Revision 03654275 - 12/09/2017 12:44 AM - Mark Abraham

Refine ulp tolerances for some more ewald tests

Fixes #2337

Change-Id: I86bb615fa147988c8e54f0bbd7e17c2f61b312d0

Revision 80dd3f5b - 12/21/2017 10:54 AM - Mark Abraham

Update double-precision test configurations

These changes improve coverage of double precision, using more release mode, particularly with latest gcc and icc, and using 128-bit SIMD, which have been cases that were buggy recently. The other aspects of the configurations that have been modified have been non-critical. Where appropriate, brief rationales are recorded. This resolves an old TODO item in the post-submit matrix.

Fixed a sign mismatch in initializing an OpenCL variable that didn't need to be initialized.

Noted relevant new TODOs.

Refs #2300, #2325, #2326, #2334, #2335, #2336, #2337, #2338

Change-Id: I131fa1a6776d1e7809799c3f931a1fc8100fcdc9

History

#1 - 12/07/2017 05:45 AM - Mark Abraham

Related, gcc 7.2.0 double-precision doesn't like another of the test tolerances:

```
[ RUN      ] DifferentEwaldCoeffLJ/PmeSolveTest.ReproducesOutputs/7
../src/testutils/refdata.cpp:918: Failure
  In item: /Virial/Cell 0 0
    Actual: -381.37902032167199
    Reference: -381.37902032167398
Difference: 1.98952e-12 (35 double-prec. ULPs, rel. 5.22e-15)
Tolerance: abs. 1.77636e-12, 8 ULPs
Google Test trace:
../src/gromacs/ewald/tests/pmesolvetest.cpp:146: Testing solving (Lennard-Jones, YZX, with energy/virial) with
CPU for PME grid size 9 7 23, Ewald coefficients 2 2.35
[ FAILED ] DifferentEwaldCoeffLJ/PmeSolveTest.ReproducesOutputs/7, where GetParam() = ({ 7, 0, 0, 0, 4.1, 0,
3.5, 2, 12.2 }, 12-byte object <09-00 00-00 07-00 00-00 17-00 00-00>, { (12-byte object <00-00 00-00 04-00 00
-00 00-00 00-00>, 16-byte object <00-00 00-00 00-00 00-00 00-00 00-40 33-33 D3-3F>), (12-byte object <04-00 00
-00 02-00 00-00 07-00 00-00>, 16-byte object <00-00 00-C0 1E-85 2B-40 00-00 00-00 00-00 00-00 44-C0>), (12-byte obje
ct <00-00 00-00 06-00 00-00 07-00 00-00>, 16-byte object <00-00 00-C0 CC-CC 0C-40 00-00 00-00 00-00 00-00 00-00>), (
12-byte object <02-00 00-00 05-00 00-00 0A-00 00-00>, 16-byte object <00-00 00-C0 CC-CC 0C-40 00-00 00-C0 CC-4
C 25-40> ) , 1.2, 2, 2.35, 4-byte object <01-00 00-00>) (0 ms)
[-----] 8 tests from DifferentEwaldCoeffLJ/PmeSolveTest (6 ms total)
```

icc 18.0.1.163 double-precision

```
[ RUN      ] DifferentEwaldCoeffLJ/PmeSolveTest.ReproducesOutputs/7
../src/testutils/refdata.cpp:918: Failure
  In item: /Virial/Cell 0 0
    Actual: -381.37902032167176
    Reference: -381.37902032167398
Difference: 2.21689e-12 (39 double-prec. ULPs, rel. 5.81e-15)
Tolerance: abs. 1.77636e-12, 8 ULPs
Google Test trace:
../src/gromacs/ewald/tests/pmesolvetest.cpp:146: Testing solving (Lennard-Jones,
YZX, with energy/virial) with CPU for PME grid size 9 7 23, Ewald coefficients
2 2.35
[ FAILED ] DifferentEwaldCoeffLJ/PmeSolveTest.ReproducesOutputs/7, where GetPa
ram() = ({ 7, 0, 0, 0, 4.1, 0, 3.5, 2, 12.2 }, 12-byte object <09-00 00-00 07-00
00-00 17-00 00-00>, { (12-byte object <00-00 00-00 04-00 00-00 00-00 00-00>, 16
-byte object <00-00 00-00 00-00 00-00 00-00 00-40 33-33 D3-3F>), (12-byte object
<04-00 00-00 02-00 00-00 07-00 00-00>, 16-byte object <00-00 00-C0 1E-85 2B-40
00-00 00-00 00-00 00-00 44-C0>), (12-byte object <00-00 00-00 06-00 00-00 07-00 00-00>
, 16-byte object <00-00 00-C0 CC-CC 0C-40 00-00 00-00 00-00 00-00 00-00>), (12-byte ob
ject <02-00 00-00 05-00 00-00 0A-00 00-00>, 16-byte object <00-00 00-C0 CC-CC 0C
-40 00-00 00-C0 CC-4C 25-40> ) , 1.2, 2, 2.35, 4-byte object <01-00 00-00>) (1 m
s)
[-----] 8 tests from DifferentEwaldCoeffLJ/PmeSolveTest (7 ms total)
```

I'll update some of the matrices to do a better job of covering double precision

#2 - 12/07/2017 06:04 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2337](#).
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~release-2018~1131fa1a6776d1e7809799c3f931a1fc8100fcdc9
Gerrit URL: <https://gerrit.gromacs.org/7303>

#3 - 12/07/2017 08:54 AM - Aleksei lupinov

Those 2 were never caught neither by our matrices nor by me, so they are technically not blocking the beta. Increase of testing coverage would definitely make my life easier....

#4 - 12/08/2017 02:06 AM - Mark Abraham

Aleksei lupinov wrote:

Those 2 were never caught neither by our matrices nor by me, so they are technically not blocking the beta. Increase of testing coverage would definitely make my life easier....

Indeed, working on the latter.

It does illustrate that picking arbitrary tolerances mean that tests pass only at the mercy of the implementation of the compiler optimization, which we didn't anticipate well, and which will change as the compilers develop. We should do a proper propagation-of-errors analysis, e.g. assuming suitable increase in ulp tolerance per flop, perhaps acknowledging cancellation possibilities where appropriate, so that the tests are designed to be robust.

#5 - 12/08/2017 03:04 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2337](#).
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~release-2018~186bb615fa147988c8e54f0bbd7e17c2f61b312d0
Gerrit URL: <https://gerrit.gromacs.org/7309>

#6 - 12/09/2017 12:58 AM - Roland Schulz

- Status changed from New to Closed