GROMACS - Bug #1157

float/real mismatches in cuda code

02/24/2013 02:38 AM - Peter Kasson

Status: Closed
Priority: Normal
Assignee: Peter Kasson
Category: mdrun
Target version: 4.6.4
Affected version - extra info: release-4-6
Difficulty: uncategorized
Affected version: 4.6.1

Description
I came across this when I accidentally tried to compile double. But this looks iffy to me. Am I missing something?

nbxn_cuda.cu:
void nbxn_cuda_wait_gpu(nbnxn_cuda_ptr_t cu_nb,
const nbnxn_atomdata_t *nbatom,
int flags, int aloc,
float *e_lj, float *e_el, rvec *fshift)

nbxn_cuda.cu:
void nbxn_cuda_wait_gpu(nbnxn_cuda_ptr_t cu_nb,
const nbnxn_atomdata_t * nbatom,
int flags, int aloc,
real *e_lj, real *e_el,
rvec *fshift) FUNC_TERM

Associated revisions

Revision 37985183 - 02/28/2013 05:02 PM - Peter Kasson
Declaration-definition consistency nbxn_cuda_wait_gpu.
Changed float->real in nbxn_cuda_wait_gpu() for consistency between
function declaration and function definition. Added note that this is
only implemented for single-precision at the moment. CMake will
complain if GMX_GPU and GMX_DOUBLE are both set.
Addresses Bug #1157.
Change-Id: Ide495cbaba6d120d91106c6a87ca04e46a2f5a8

Revision 68bd2ee6 - 06/13/2013 09:45 AM - Peter Kasson
Declaration-definition consistency nbxn_cuda_wait_gpu.
Changed float->real in nbxn_cuda_wait_gpu() for consistency between
function declaration and function definition. Added note that this is
only implemented for single-precision at the moment. CMake will
complain if GMX_GPU and GMX_DOUBLE are both set.
Addresses Bug #1157.
Change-Id: Ide495cbaba6d120d91106c6a87ca04e46a2f5a8

History

#1 - 02/24/2013 01:00 PM - Szilárd Páll
This is indeed not very nice, but in single-precision builds real is typedef-ed to float anyway, so I don't think it is a major concern - as long as we don't
have double or some mixed precision modes supported on GPUs.

#2 - 02/24/2013 11:25 PM - Peter Kasson
Part it comes down to how we want the double-precision compile failure mode to look. As is, this causes compile errors.
If I read correctly, it looks like we detect GMX_GPU and GMX_DOUBLE in cmake/gmxManageGPU.cmake and throw an error.
This sounds like the right behavior.
So it isn't a real issue, but for code cleanliness I might suggest consistent usage (either real or float consistently) in the code. I'm happy to implement it and send you a CL for review. Do you have a preference for real vs. float? (real would help future-proof the code in case we want to support double in the future and also be more consistent with the rest of our codebase. float would make the float-only nature of the current CUDA code more explicit.)

#3 - 02/28/2013 01:17 AM - Szilárd Pál

- Category set to mdrun

Indeed, for code cleanliness it's nest if this issues is solved. We have been discussing that it might be minor amount of effort to enable double-precision support in the CUDA kernels, but it has to be seen ho much will the performance at least on GPUs which are not crippled (i.e most GeForce and K10) suffer from the need to emulate double atomic ops (which is not supported natively).

#4 - 04/29/2013 07:46 PM - Mark Abraham

- Target version deleted (4.6.1)
- Affected version set to 4.6.1

#5 - 07/17/2013 02:04 PM - Teemu Murtola

- Status changed from New to Resolved
- Assignee set to Peter Kasson
- Target version set to 4.6.x

Had been resolved by the linked commit, probably for 4.6.2.

#6 - 10/16/2013 03:39 PM - Mark Abraham

- Status changed from Resolved to Closed
- Target version changed from 4.6.x to 4.6.4