

## GROMACS - Feature #2715

### Avoid requesting the user to recompile gromacs for Intel OpenCL support

10/29/2018 08:51 AM - Erik Lindahl

<b>Status:</b>	New	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Category:</b>	core library	
<b>Target version:</b>	2020	
<b>Difficulty:</b>	uncategorized	
<b>Description</b>		
Instead of setting <code>c_nbnxnGpuClusterSize</code> to a constant based on a CMake setting, it seems like the obvious solution is to make it a template parameter and compile with by 4 and 8.		
<b>Related issues:</b>		
Related to GROMACS - Task #2518: redesign task-assignment code for OpenCL		<b>New</b>

#### History

##### #1 - 10/29/2018 10:41 AM - Mark Abraham

- Category set to core library

- Target version set to 2020

Indeed. It need not be necessary to template, either. It's not clear to me that the setup and search code benefits a lot from knowing that the cluster width is a compile-time constant, since its role is to set up data structures that also have several dynamically sized aspects. Since addressing this does require functional changes to the NBNXN module, we should target 2020.

Unfortunately the Intel OpenCL functionality sat in Gerrit and was not integrated over the May-August timeframe despite review, fixes, and voting from several people. That probably inhibited people from considering working on such usability aspects. Clearly we need to improve our workflows there. It's not acceptable that someone proposing GPU code has to wait months for us to prioritize the final stages of integrating it. We like having input from those most experienced with our GPU code, but when they are not available, we need to integrate so we can work on other aspects without large rebase chains. If there had been functional aspects of the Intel OpenCL code to address when we had time from those developers (e.g. I believe Intel OpenCL is currently broken for PME), it would have been fine to fix them in follow up work in August-September.

##### #2 - 10/30/2018 07:56 AM - Mark Abraham

- Related to Task #2518: redesign task-assignment code for OpenCL added

##### #3 - 10/30/2018 07:57 AM - Mark Abraham

I mentioned this issue as one of the larger (and also unaddressed) issues associated with OpenCL support at [#2518](#)

##### #4 - 10/31/2018 09:24 AM - Berk Hess

There are only a handful of functions in `nbnxn_search.cpp` which would need to be templated. The OpenCL kernels are always jitted, right? Then this is a rather small change.

##### #5 - 10/31/2018 12:09 PM - Mark Abraham

Berk Hess wrote:

There are only a handful of functions in `nbnxn_search.cpp` which would need to be templated.

Do we **need** templating? We would anyway need to have a cluster size chosen at run time in order to choose the instance of the template to use, and it is simpler to just pass 4 or 8. :-)

The OpenCL kernels are always jitted, right? Then this is a rather small change.

Yes, always jitted. I'd hope it was fairly small, but I didn't want to tackle it while Intel OpenCL support was still in gerrit and the scale of any changes to search code for the update groups wasn't known.

##### #6 - 10/31/2018 12:16 PM - Szilárd Páll

I'm not sure we need templating, in particular not the pairlist initialization, but when I looked into it it seemed to me that some reorganization of data

members around the search grid cell size ( $na_c$ ) and search-time cluster size would be necessary.