GROMACS - Task #2453

Feature # 2054 (Accepted): PME on GPU

PME OpenCL porting effort

03/14/2018 01:55 PM - Aleksei Iupinov

Status:	Resolved	
Priority:	High	
Assignee:	Aleksei Iupinov	
Category:		
Target version:	2021	
Difficulty:	uncategorized	

Description

With porting PME from CUDA to OpenCL I'm first going with a dirty code with lots of duplication to see how to strike a balance between neatness and extensibility. Most of the host-side logic is quite easy to wrap to look the same in CUDA/OpenCL since there is no C++ limitations.

Functionality achieved (https://github.com/yupinov/gromacs/tree/pme_opencl_gerrit):
- PME OpenCL kernels passing unit tests on NVIDIA and AMD GPUs;
- clFFT ~~still broken~~ fixed upstream in with RoCM 2.0.

TODO:
- ~~check correctness on Intel;~~
- document and cleanup FIXMEs;
- subtasks.

Subtasks:

Task # 2498: OpenCL memory pinning/mapping	**New**
Task # 2500: detect and allow linking external clFFT, or no clFFT	**Closed**
Task # 2514: PME OpenCL reductions with intrinsics	**New**
Task # 2515: clFFT RocM compatibility problem	**Closed**
Task # 2516: Support PME OpenCL execution width < 16	**New**
Task # 2519: Improve/remove PME OpenCL kernel barriers	**New**
Task # 2520: Treat OpenCL kernel width more diligently	**New**
Task # 2521: Implement alternating PME/NB wait for OpenCL	**New**
Task # 2522: OpenCL context duplication	**Resolved**
Task # 2527: Rename GpuEventSynchronizer to something more fitting (after mergin PME Op...	**New**
Task # 2529: Improve test timeouts handling	**Closed**
Task # 2531: Consider optimizing tabulated data access on GPU	**New**
Task # 2532: enable queue priorities in OpenCL	**New**
Task # 2535: consider compiling opencl fft kernels once	**New**
Bug # 2536: clFFT execution not timed in PME	**Closed**
Task # 2537: Simplify PME solve reduction	**New**

Related issues:

Related to GROMACS - Task #2402: PME kernels general performance improvements	**New**
Related to GROMACS - Task #2524: struct alignment/packing for OpenCL host & d...	**New**

Associated revisions

Revision b2a95c76 - 05/28/2018 03:11 PM - Aleksei Iupinov

Support persistent device context-derived data in PME tests

PME OpenCL will need to not recompile kernels for running each
unit test. With this in mind, a persistent PmeGpuProgram class
is tasked with GPU kernel setup, and passed around, using a typedef.
The purpose of the class is to hold the PME program data that should
only be set up once and live forever for the given device context.
PmeGpuProgramImpl structure is now tasked with managing
function pointers to the CUDA kernels' instances, and will later

be tasked with compiling OpenCL kernels.

Refs #2453, #2522

Change-Id: I85a01bfc92ec3a108825414b14e2be3731433c9a

Revision 442845fd - 05/31/2018 09:10 AM - Aleksei Iupinov

Gather the PME GPU constants/macros in a single header

Those will be translated to defines by the OpenCL kernel compiler.

Refs #2453, #2528

Change-Id: I03062c908db6dad9bbc8c62accdc4707b03ff527

Revision a19dd7d5 - 08/15/2019 09:46 PM - Szilárd Páll

Fix OpenCL gather reduction

On >=16-wide execution it is correct (narrower is checked and excluded
during compilation).

TODO: Consider changing the default on NVIDIA & Intel where offloading
PME is generally not advantageous to performance.

Addresses part of #2519
Refs #2453 #2516

Change-Id: I24beaaeea096954ba32b3a80251945a9d82a3c05

History

#1 - 03/14/2018 02:14 PM - Aleksei Iupinov

- Description updated

#2 - 03/14/2018 02:44 PM - Aleksei Iupinov

- Description updated

#3 - 03/14/2018 03:09 PM - Aleksei Iupinov

- Subject changed from OpenCL PME porting effort to PME OpenCL porting effort

- Description updated

#4 - 03/14/2018 03:16 PM - Aleksei Iupinov

- Parent task set to #2054

#5 - 03/14/2018 03:16 PM - Aleksei Iupinov

- Private changed from Yes to No

#6 - 03/14/2018 03:17 PM - Aleksei Iupinov

- Assignee set to Aleksei Iupinov

- Target version set to 2019

#7 - 03/27/2018 01:56 PM - Aleksei Iupinov

- Description updated

Update: gather kernel working as well as spread on AMD/Intel/NVidia

https://github.com/yupinov/gromacs/tree/pme_opencl_dirty (the code is a mess and likely won't even compile without OpenCL)

#8 - 04/30/2018 12:26 PM - Aleksei Iupinov

Haven't updated this in a while. Some preliminary changes have already made their way into master branch, more to come.

Conclusion to whether we want most common GPU (host) code in a shared implementation header, or a source file that is made special (compiled by
nvcc with CUDA) by cmake rules:
Whatever works. I have a working draft of such a cmake rule. If I finish it and it makes through code review - good. Otherwise we will have to resolve

to 2 small CUDA/OpenCL source files including large shared implmenetation header. Note that NB still has to face this de-duplication problem as well.

The answer to the naming rules we have worked out with Szilard - given that OpenCL 1.2 doesn't allow for true constexprs, we agreed to allow using same c_usefulConstant syntax for defines (which will have to defined when compiling the OpenCL kernels). For run-time constants, which actually live in a constant GPU memory (together with the kernel arguments), we agreed to use cm_ prefix. I have to see whether I follow that everywhere in the PME GPU device-side code.

The answer to the last 3 checkboxes ticked, about possibilities of CUDA/OpenCL device duplication - I will duplicate all the kernel code. All the macro decoration is not worth it for now with OpenCL 1.2.
The exceptions still possibly to consider would be wrapping a few one-liners with 100% same meaning (e.g. threadIdx.y and get_local_id(1) to get index of the thread, __syncthreads() and barrier(CLK_LOCAL_MEM_FENCE) to synchronize block-local memory), as the checkbox above says. That still won't do much for reducing number of different lines though.

#9 - 04/30/2018 12:31 PM - Aleksei Iupinov

- Description updated

#10 - 05/07/2018 11:43 AM - Aleksei Iupinov

DeviceBuffer functions are in master.
As noted in #2498, changingPinningPolicy will have to become more permissive for OpenCL allocations, one way or another, as there are no way to do one-sided host pinning in CUDA sense.
Still figuring out the way to handle a persistent PME GPU context data struct.

#11 - 05/24/2018 06:13 PM - Aleksei Iupinov

- Description updated

#12 - 05/25/2018 11:25 AM - Aleksei Iupinov

- Related to Task #2402: PME kernels general performance improvements added

#13 - 05/26/2018 03:17 PM - Szilárd Páll

- Related to Task #2524: struct alignment/packing for OpenCL host & device code added

#14 - 05/28/2018 03:13 PM - Gerrit Code Review Bot

Gerrit received a related patchset '19' for Issue #2453.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~master~I85a01bfc92ec3a108825414b14e2be3731433c9a
Gerrit URL: https://gerrit.gromacs.org/7818

#15 - 05/29/2018 03:08 PM - Gerrit Code Review Bot

Gerrit received a related patchset '6' for Issue #2453.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~master~I03062c908db6dad9bbc8c62accdc4707b03ff527
Gerrit URL: https://gerrit.gromacs.org/7861

#16 - 07/30/2018 05:52 PM - Mark Abraham

There were some open questions about naming some constants at https://gerrit.gromacs.org/#/c/7924/5 before we submitted it. We should reconsider these once some more stuff is in.

#17 - 10/10/2018 12:12 PM - Mark Abraham

Currently our task assignment will run PME tasks on an Intel OpenCL device, which segfaults on my laptop. Not sure where the issue is (it has once worked), but we need to at least revisit our defaults for task assignment.

#18 - 12/31/2018 11:17 AM - Paul Bauer

- Status changed from New to Resolved

I resolved this for now because all the checkboxes have been ticked off, but please tell me if this should be retargeted instead.

#19 - 12/31/2018 01:47 PM - Paul Bauer

- Target version changed from 2019 to 2020

retargeted because of the remaining subtasks

#20 - 04/16/2019 05:42 PM - Szilárd Páll

- Description updated

#21 - 04/23/2019 03:52 PM - Szilárd Páll

- Description updated

#22 - 12/20/2019 11:58 AM - Paul Bauer

- Target version changed from 2020 to 2021