GROMACS - Task #3312
Feature # 3311 (In Progress): GPU infrastructure development

Data type for coordinates, xyzq data, LJ parameters data to use for GPU buffers

01/22/2020 08:44 AM - Artem Zhmurov

**Status:** In Progress  
**Priority:** Normal  
**Assignee:** Artem Zhmurov  
**Category:**  
**Target version:** 2021-refactoring  
**Difficulty:** uncategorized

**Description**
To use opaque DeviceBuffer type in CPU parts of the code, one needs a proper data-type for, e.g. coordinates. Current solutions include passing a void* and using DeviceBuffer<float>, both of which are faulty. Proposed solutions are:

**Solution 1. Declare native GPU types for the CPU code-path.**
Both CUDA and OpenCL have native vector types, which can be declared for CPU code path.

**Pros:**
- Native types - no need for casting.

**Cons:**
- Polluted data-type space.
- Introducing new data type will require defining it across the platforms.
- Potentially, more difficult integration of OpenCL and CUDA code-paths.
- SYCL?

**Problems:**
- OpenCL float3 format has float4 layout.

**Solution 2. Define new or use existing CPU types.**

**Pros:**
- No need to define new data types for most used objects, e.g. can use RVec for coordinates.
- Casting can be done in the GPU kernel: the rest of the code can potentially be platform-agnostic.

**Cons:**
- Data will have to be casted to native types, probably inside computational kernel. Safety checks for the casts will be required.
- Some new data types will be needed (e.g. for C6-C12 LJ parameters).

**Examples:**
- [https://gerrit.gromacs.org/#/c/gromacs/+/15439/](https://gerrit.gromacs.org/#/c/gromacs/+/15439/)

**Subtasks:**
- Bug # 3372: Re-enable RVec and float3 compatibility tests  
  **Closed**

**Related issues:**
- Related to GROMACS - Task #2936: introduce check that CPU-GPU transfers/assig...  
  **New**

**Associated revisions**
- Revision c5c220a0 - 02/06/2020 08:49 AM - Artem Zhmurov
  Use RVec instead of float for x, v and f device buffers

- Using RVec instead of float for coordinates data-types allows to remove multiplications by DIM when the addresses, offsets and sizes

04/05/2020
are computed. Since the native device types are not used in CPU part of the code, the type casting remains.

Refs #3312 and #2936

Change-id: laea914a474195f214ca860f7345f6878e9a04813

History

#1 - 01/22/2020 09:41 AM - Artem Zhmurov
- Tracker changed from Feature to Task
- Assignee set to Artem Zhmurov

#2 - 01/22/2020 09:45 AM - Artem Zhmurov
- Target version set to 2021

#3 - 01/22/2020 09:47 AM - Artem Zhmurov
- Target version changed from 2021 to 2021-refactoring

#4 - 01/22/2020 04:52 PM - Mark Abraham
Solution 1 also makes any code that uses the compatibility types (even just by name) dependent on the value of GMX_GPU. Currently that would make for a nasty dependency on config.h. That nastiness can be tackled, but solution 2 naturally solves it by having types that make sense in the domain of use (e.g. also xyzq for nbnxm module, fdv0 for tables)

#5 - 01/24/2020 11:47 AM - Artem Zhmurov
If there are no strong arguments against, I suggest we go with Solution 2 (see e.g. https://gerrit.gromacs.org/#/c/gromacs/+/15439/). @Alan, do you have anything against/for this decision?

#6 - 01/24/2020 03:30 PM - Alan Gray

@Alan, do you have anything against/for this decision?

I agree with the decision: I've recently found it very tricky in https://gerrit.gromacs.org/c/gromacs/+/14223 to work with all the different types and this should make things easier.

#7 - 02/05/2020 06:06 PM - Szilárd Páll
- Related to Task #2936: introduce check that CPU-GPU transfers/assignments are made between compatible types added