

GROMACS - Feature #2967

GPU reallocateDeviceBuffer improvements

06/03/2019 05:25 PM - Szilárd Páll

Status:	New
Priority:	Normal
Assignee:	
Category:	mdrun
Target version:	2020-beta3
Difficulty:	uncategorized
Description	
<p>When the "buffered" GPU allocation was reimplemented/modernized, the buffering got dropped (i.e. the <code>size_alloc = size</code> at first allocation effectively disabling the buffering).</p> <p>We should extend the current implementation providing an overloaded version (to accommodate for current legacy code) of <code>reallocateDeviceBuffer()</code> that: takes a standard vector / <code>ArrayRef</code> and also re-introduces the buffering.</p> <p>The open question is how to implement buffering? Options:</p> <ul style="list-style-type: none">- always keep the device array size/allocation size in sync with <code>h_vector.size()</code> and <code>h_vector.capacity()</code>, resp;- using <code>h_vector.capacity()</code> everywhere (in particular PME grids) could risk running out of device memory so consider a different capacity heuristic.	

History

#1 - 06/13/2019 02:14 PM - Berk Hess

Buffering is actually done in `reallocateDeviceBuffer()`, but it is not coupled to the CPU side allocation. We might or might not want that.

#2 - 06/13/2019 02:41 PM - Szilárd Páll

Berk Hess wrote:

Buffering is actually done in `reallocateDeviceBuffer()`, but it is not coupled to the CPU side allocation. We might or might not want that.

As we discussed offline, if we want to keep the standard vector allocation behavior and not attempt a always/mostly reserve and avoid `post_back` approach (for host buffers of data that's copied to the GPU), than for the GPU-side buffers we likely want a custom buffered allocation that does not simply use the `h_vector.capacity()` as this could easily lead to running out of memory on low-end GPUs with little global memory.

#3 - 09/24/2019 06:52 PM - Mark Abraham

- Target version changed from *2020-beta1* to *2020-beta2*

#4 - 10/31/2019 03:35 PM - Szilárd Páll

- Target version changed from *2020-beta2* to *2020-beta3*