GROMACS - Task #3077

Task # 3370 (New): Further improvements to GPU Buffer Ops and Comms
Feature # 2891 (In Progress): PME/PP GPU communications

PME/PP GPU Comms unique pointer deletion causes seg fault when CUDA calls exist in destructor

09/05/2019 03:48 PM - Alan Gray

Status: Feedback wanted
Priority: Normal
Assignee:
Category:
Target version:
Difficulty: uncategorized

Description
When the unique pointer used for the PME-PP GPU communications objects are automatically deleted, the code sometimes seg-faults. I originally thought this was only the case when CUDA calls exist in the destructor, but have now also seen it happen even with default destructors. I have reverted to regular pointers for now. This should be investigated further, with unique pointers reinstated.

Related issues:
Related to GROMACS - Feature #3115: Device stream manager

Associated revisions
Revision c2e5f578 - 10/29/2019 10:55 AM - Alan Gray
Explicitly destroy PME-PP GPU communication object
Add code to destroy object when it is no longer required. Even although object is managed by a unique pointer, this needs to be done while the GPU context still exists, otherwise a seg fault can occur when it is automatically destroyed later.
Addresses #3077
Change-Id: I9d6f798d79a73e2ce366c9fb85a0ff939fc9f88

History
#1 - 09/11/2019 06:10 PM - Alan Gray
- Description updated

#2 - 09/19/2019 12:15 PM - Szilárd Páll
Is this still an issue?

#3 - 09/19/2019 01:03 PM - Alan Gray
Yes, it's still an issue - I've not had time to properly investigate/fix it yet.

#4 - 09/20/2019 01:04 PM - Mark Abraham
I've not seen any issues with such patches

#5 - 10/29/2019 03:57 PM - Alan Gray
- Status changed from New to Closed

#6 - 11/01/2019 06:56 PM - Szilárd Páll
- Status changed from Closed to Feedback wanted

We not have the same issue with gpuHaloExchange, I assume, only because we are not doing cudaStreamCreate?

Also, while looking into this I realized that:
- c2e5f578 added the freeing quite early; I suggest moving it closer to the place where related freeing happens.
in runner.cpp, around where gmx_pme_destroy() is called.
- we do not have a cudaStreamDestroy for pmepCommStream_; I suggest adding the missing call to the destructor.

As noted on #3021, we need docs on this lifetime management concerns.
Side-note: we could side-step such issues if we had the code for #3115 as that would make the lifetime dependencies more clear.

#7 - 11/01/2019 06:56 PM - Szilárd Pál
- Related to Feature #3115: Device stream manager added