

## GROMACS - Bug #2405

### improve gpu\_utils-test

02/08/2018 06:41 PM - Mark Abraham

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b> Mark Abraham	
<b>Category:</b> testing	
<b>Target version:</b> 2018.1	
<b>Affected version - extra info:</b>	<b>Difficulty:</b> uncategorized
<b>Affected version:</b> 2018	
<b>Description</b>	
<p>I added some actual device transfers in some of the "unit" tests here, but that seems to be a bad idea. At least one user had a setup where the whole test binary took 75 seconds to run, and it's even 7 s on my desktop. Szilard noted at the time that the transfers were not needed, but we ran with it because the tests were set up to not use a device if one wasn't found.</p> <p>Plus gentoo, debichem and bioconda have found issues with these tests for non-GPU builds, even though I built them to work, and our Jenkins non-GPU builds have been OK.</p> <p>So I propose</p> <ul style="list-style-type: none"><li>• we eliminate device transfers from tests for builds that support GPU execution</li><li>• we eliminate most/all of the tests for builds that do not support GPU execution</li></ul>	
<b>Related issues:</b>	
Related to GROMACS - Bug #2420: OpenCL implementation not doing device sanity...	<b>Closed</b>

### Associated revisions

#### Revision ea874de9 - 02/13/2018 05:12 PM - Mark Abraham

Use death test only when supported

In build farms for bioconda and DebiChem, some environments don't support death tests, so hopefully if we use the GoogleTest machinery to avoid using them when not supported we can avoid issues in such environments.

Refs #2405

Change-Id: I17efe3b620b162d615b91a5f5cd4c8bd458143a5

#### Revision 8223d564 - 02/27/2018 10:14 AM - Mark Abraham

Fix GPU utils test

We should only attempt device-side work when we have compatible GPUs.

Fixes #2405 for CUDA

Change-Id: Idc28c7e89a5f08ee1d19943e3663385f2b23ff44

### History

#### #1 - 02/12/2018 12:29 PM - Aleksei lupinov

I have no objections. At the time I probably didn't voice agreement with Szilard to not push hard on minor issues just before the release. I do agree that transfers are not the thing being tested here.

By the way, I assume that most of those 7 seconds (or 10 times as much without driver persistence) come from the test calling `cudaDeviceReset()` after each instance.

I have to admit that this is something that I deliberately and sneakily did not do in the PME unit tests exactly because it takes so much time. PME GPU structures, CUDA stream, etc is created/destroyed for each test instance, but device is not reset, and contexts are reused. With that, some real code issues have been found by those tests anyway, so I assume it is safe to not reset the device for different test instances (unless we're doing something extraordinary like mixing different compute capabilities and GPU code paths, which, as we have seen before on Jenkins, is likely to be broken anyway). Maybe we can just have a couple separate tests like `CanSwitchContextsAndLaunchWork`, `CanResetDeviceAndLaunchWorkAgain`,

etc.

## #2 - 02/13/2018 02:52 PM - Szilárd Páll

Aleksei Lupinov wrote:

By the way, I assume that most of those 7 seconds (or 10 times as much without driver persistence) come from the test calling `cudaDeviceReset()` after each instance.

How long does a reset take and any idea what influences its cost?

With that, some real code issues have been found by those tests anyway,

Interesting, namely what?

so I assume it is safe to not reset the device for different test instances

Not sure how is the former later to this latter point of mixing ctxes being safe?

(unless we're doing something extraordinary like mixing different compute capabilities and GPU code paths, which, as we have seen before on Jenkins, is likely to be broken anyway).

I've not thought of this much, but I think we should *clearly* document such behavior if that's a feature or at least motivate the reuse of the context in code comments.

## #3 - 02/13/2018 03:20 PM - Szilárd Páll

Mark Abraham wrote:

Plus gentoo, debichem and bioconda have found issues with these tests for non-GPU builds, even though I built them to work, and our Jenkins non-GPU builds have been OK.

Any idea what was the underlying reason? Can you ref those failures?

So I propose

- we eliminate device transfers from tests for builds that support GPU execution

I'm not sure we *have* to do that; in general, in the presence of a compatible device, just as testing on-device kernels is reasonable, some utility function that wraps a host-<->device transfer is reasonable too. Admittedly, the usefulness of such a test is limited without actually testing the functionality e.g. memory content transferred (e.g. transfer an array size N values, initialized with values 1..N, and test the content on-device).

- we eliminate most/all of the tests for builds that do not support GPU execution

I agree with that point. Testing some transfer-related logic may be useful with no device present, but in the current use-cases there is no need for that.

## #4 - 02/13/2018 04:10 PM - Mark Abraham

Szilárd Páll wrote:

Mark Abraham wrote:

Plus gentoo, debichem and bioconda have found issues with these tests for non-GPU builds, even though I built them to work, and our Jenkins non-GPU builds have been OK.

Any idea what was the underlying reason? Can you ref those failures?

From email with Stian Soiland-Reyes (part of BioExcel) regarding bioconda:

```
There was also an MPI GPU test that then fell over, which I found odd given that I had -DGMX_GPU=off -- it timed out trying to run ChangingPinningPolicyRequiresCuda which GTest gave a multi-thread-fork() warning about.
```

I can look more into this and report properly if you think there might be something there, but perhaps for users installing from bioconda we

can assume they won't use multi-machine MPI?

Also <http://lists.aliases.debian.org/pipermail/debichem-devel/2018-January/008785.html> - but the issue may have been with GoogleTest death tests on that platform.

I can't find anything to back up my memory of a Gentoo report.

So I propose

- we eliminate device transfers from tests for builds that support GPU execution

I'm not sure we *have* to do that; in general, in the presence of a compatible device, just as testing on-device kernels is reasonable, some utility function that wraps a host->device transfer is reasonable too. Admittedly, the usefulness of such a test is limited without actually testing the functionality e.g. memory content transferred (e.g. transfer an array size N values, initialized with values 1..N, and test the content on-device).

One doesn't have to test the content on device, you can transfer it back into a new buffer and observe that the results are what you expect. That's how we test simd load and store, and is good enough unless one would aim to catch some kind of maliciousness...

But we can leave this point up in the air for 2018.1 and see if people continue to have problems.

- we eliminate most/all of the tests for builds that do not support GPU execution

I agree with that point. Testing some transfer-related logic may be useful with no device present, but in the current use-cases there is no need for that.

#### #5 - 02/13/2018 05:11 PM - Mark Abraham

Mark Abraham wrote:

- we eliminate most/all of the tests for builds that do not support GPU execution

Looking more closely, the only issue at hand seems to be that the death test isn't supported everywhere, and we don't use those anywhere else, so perhaps we can use the portable option GoogleTest supplies and we've solved the problem well enough

#### #6 - 02/13/2018 05:12 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2405](#).

Uploader: Mark Abraham ([mark.j.abraham@gmail.com](mailto:mark.j.abraham@gmail.com))

Change-Id: gromacs~release-2018~117efe3b620b162d615b91a5f5cd4c8bd458143a5

Gerrit URL: <https://gerrit.gromacs.org/7570>

#### #7 - 02/16/2018 03:29 PM - Mark Abraham

- Status changed from New to Resolved

I'm still open to removing more things that prove problematic, but we can find out how these tests work in the external CI environments first.

#### #8 - 02/17/2018 07:48 PM - Szilárd Páll

Related: while testing [#2415](#) I switched a GPU in my machine to "prohibited" mode (nvidia-smi -c 2) and it seems that this upsets these unit tests which fail due to encountering cudaErrorDevicesUnavailable (needs the fix to [#2415](#) to not trip first an assertion that checks for previous errors).

#### #9 - 02/23/2018 01:25 PM - Mark Abraham

Szilárd Páll wrote:

Related: while testing [#2415](#) I switched a GPU in my machine to "prohibited" mode (nvidia-smi -c 2) and it seems that this upsets these unit tests which fail due to encountering cudaErrorDevicesUnavailable (needs the fix to [#2415](#) to not trip first an assertion that checks for previous errors).

That fix is now merged, and I'll see if I can reproduce this report.

#### #10 - 02/23/2018 03:17 PM - Gerrit Code Review Bot

Gerrit received a related patchset '2' for Issue [#2405](#).

Uploader: Mark Abraham ([mark.j.abraham@gmail.com](mailto:mark.j.abraham@gmail.com))

Change-Id: gromacs~release-2018~ldc28c7e89a5f08ee1d19943e3663385f2b23ff44

Gerrit URL: <https://gerrit.gromacs.org/7621>

**#11 - 02/23/2018 03:18 PM - Mark Abraham**

Szilárd Páll wrote:

Related: while testing [#2415](#) I switched a GPU in my machine to "prohibited" mode (nvidia-smi -c 2) and it seems that this upsets these unit tests which fail due to encountering cudaErrorDevicesUnavailable (needs the fix to [#2415](#) to not trip first an assertion that checks for previous errors).

The test logic was wrong - it checked for detected devices, but needs to check for useful devices, which is "compatible" ones for now.

**#12 - 02/23/2018 06:29 PM - Aleksei lupinov**

- Related to Bug #2420: OpenCL implementation not doing device sanity checks added

**#13 - 02/27/2018 10:15 AM - Mark Abraham**

Applied in changeset [8223d564e68f1ff32985d51c43f2764f18207485](#).

**#14 - 03/02/2018 03:52 PM - Mark Abraham**

- Status changed from Resolved to Closed