

## GROMACS - Bug #1985

### CUDA build system refactoring awaiting review

06/09/2016 01:13 PM - Mark Abraham

<b>Status:</b>	Fix uploaded	
<b>Priority:</b>	Low	
<b>Assignee:</b>		
<b>Category:</b>	build system	
<b>Target version:</b>	2021	
<b>Affected version - extra info:</b>	all CUDA versions	<b>Difficulty:</b> uncategorized
<b>Affected version:</b>	2016	
<b>Description</b>		
While working on an unrelated issue on a non-GPU machine,		
<pre>[micl r2016 ((f9ecd80...))] \$ (cd build-cmake-icc-debug-micl; cmake .. --debug-trycompile -UGMX_DETECTSIMD_RUN -UGMX_DETECTSIMD_COMPILED) debug trycompile on CUDA_TOOLKIT_ROOT_DIR not found or specified -- Could NOT find CUDA (missing: CUDA_TOOLKIT_ROOT_DIR CUDA_NVCC_EXECUTABLE CUDA_INCLUDE_DIRS CUDA_CUDART_LIBRARY) (Required is at least version "5.0") -- No compatible CUDA toolkit found (v5.0+), disabling native GPU acceleration -- Detecting best SIMD instructions for this CPU</pre>		
Clearly FindCUDA is simply noisy, but cmake .. -DGMX_GPU=off did not fix the issue and cmake .. -DGMX_GPU_AUTO=off did, so we have something to fix.		
<b>Related issues:</b>		
Related to GROMACS - Bug #2357: GMX_GPU=no doesn't work if initially set to auto		<b>Closed</b>

#### Associated revisions

##### Revision 6f7d2e9a - 12/20/2017 02:17 PM - Aleksei lupinov

Set GMX\_GPU\_AUTO to FALSE with GMX\_GPU defined

Refs #1985, #2357

Change-Id: I5cada97015ee94717ea6eb988b3a84a351f11293

#### History

##### #1 - 06/10/2016 02:32 AM - Szilárd Páll

The idea is that detection is in "auto" mode by default; in this case "CUDA\_FIND\_QUIETLY" is not set and FindCUDA in this case emits the first line -- which is the annoying message you're referring to, I assume. I don't think we can do much without either making FindCUDA silent or modifying the FindCUDA module.

##### #2 - 06/10/2016 08:59 AM - Mark Abraham

We can run the detection the first cmake call, and not keeping running it every time (or at least run it in quiet mode thereafter).

And GMX\_GPU=off must override GMX\_GPU\_AUTO=on, when the latter was set in a previous run of cmake.

##### #3 - 06/29/2016 03:57 AM - Szilárd Páll

Something is indeed broken because this should be handled correctly as the quiet detection is intended to kick in:  
<http://redmine.gromacs.org/projects/gromacs/repository/revisions/master/entry/cmake/gmxManageGPU.cmake#L75>

Looks like GMX\_GPU\_AUTO is stateful and if set once it will remain set, turning it off if detection fails solves the issue, but that's just a hack. Not sure what the proper solution is, but will try to come up with something.

##### #4 - 06/29/2016 06:33 AM - Teemu Murtola

Respecting GMX\_GPU=off like Mark is asking is impossible in the current design in all cases, since there is no way to tell whether the user set it if it was already set earlier automatically to the same value...

<https://gerrit.gromacs.org/#/c/5586/> can solve the issue (if it already doesn't), but it has other issues as identified by code review comments nearly half a year ago...

#### #5 - 06/29/2016 12:17 PM - Szilárd Páll

Thanks for the note Teemu, I did not notice the change as it does not ref this issue.

#### #6 - 06/29/2016 06:18 PM - Mark Abraham

@Teemu Indeed, I'd like to get back to that  
@Szilard Sure, that patch pre-dates this issue by about 5 months ;-)

#### #7 - 06/29/2016 11:36 PM - Gerrit Code Review Bot

Gerrit received a related patchset '2' for Issue [#1985](#).  
Uploader: Mark Abraham ([mark.j.abraham@gmail.com](mailto:mark.j.abraham@gmail.com))  
Change-Id: I7df81dea738da3ec9cd3971ad3507298a9f97dff  
Gerrit URL: <https://gerrit.gromacs.org/5586>

#### #8 - 12/09/2016 04:02 AM - Gerrit Code Review Bot

Gerrit received a related patchset '6' for Issue [#1985](#).  
Uploader: Mark Abraham ([mark.j.abraham@gmail.com](mailto:mark.j.abraham@gmail.com))  
Change-Id: gromacs~master~I7df81dea738da3ec9cd3971ad3507298a9f97dff  
Gerrit URL: <https://gerrit.gromacs.org/5586>

#### #9 - 12/13/2017 12:11 AM - Erik Lindahl

This too seems to work fine in the present code. If anybody wants to keep it open, please specify exactly why, and change the subject to something more concrete.

#### #10 - 12/13/2017 02:42 AM - Mark Abraham

- Subject changed from *CUDA build system is annoying to CUDA build system refactoring awaiting review*
- Status changed from *New* to *Fix uploaded*
- Target version set to *2019*

#### #11 - 12/13/2017 02:47 AM - Mark Abraham

There are issues remaining, e.g. run cmake on a machine that finds gcc and cuda, then run it again and see

```
-- Found CUDA: /usr/local/cuda (found suitable version "9.0", minimum required is "6.5")
-- Found OpenMP_C: -fopenmp
-- Found OpenMP_CXX: -fopenmp
-- Configuring done
-- Generating done
-- Build files have been written to: /home/mabraham/git/r2018/build-cmake-gcc-gpu-debug
```

But if you start from a fresh build tree and do the first cmake with `-DGMX_GPU=on`, then the second cmake only does

```
-- Found OpenMP_C: -fopenmp
-- Found OpenMP_CXX: -fopenmp
-- Configuring done
-- Generating done
-- Build files have been written to: /home/mabraham/git/r2018/build-cmake-gcc-gpu-debug
```

That's minor, of course, but both of those cases don't follow our general expectation that repeat invocations of cmake are quiet. They indicate that the way we use the cache in the current implementation should be improved, which my <https://gerrit.gromacs.org/5586> proposes. I strongly suggest we review that (after releasing 2018) before attempting any further work on the GPU support in our build system.

#### #12 - 12/14/2017 04:52 AM - Mark Abraham

More examples, this time using clang, where distros versions are often still without openmp support (e.g. clang-4.0.1 in ubuntu 17.10):

```
$ cmake .. -DCMAKE_C_COMPILER=clang -DCMAKE_CXX_COMPILER=clang++ -DGMX_GPU=on -DGMX_USE_OPENCL=on
```

...

```
-- Looking for CL_VERSION_2_0
-- Looking for CL_VERSION_2_0 - found
-- Found OPENCL: /usr/lib/x86_64-linux-gnu/libOpenCL.so (found version "2.0")
-- Could NOT find OpenMP_C (missing: OpenMP_C_FLAGS OpenMP_C_LIB_NAMES) (found ve
rsion "1.0")
```

```
-- Could NOT find OpenMP_CXX (missing: OpenMP_CXX_FLAGS OpenMP_CXX_LIB_NAMES) (found version "1.0")
CMake Warning at cmake/gmxManageOpenMP.cmake:65 (message):
  The compiler you are using does not support OpenMP parallelism. This might hurt your performance a lot, in particular with GPUs. Try using a more recent version, or a different compiler. For now, we are proceeding by turning off OpenMP.
Call Stack (most recent call first):
  CMakeLists.txt:328 (include)
```

...

```
-- Performing Test PTHREAD_SETAFFINITY
-- Performing Test PTHREAD_SETAFFINITY - Success
CMake Warning at cmake/gmxManageOpenCL.cmake:72 (message):
  To use GPU acceleration efficiently, mdrun requires OpenMP multi-threading. Without OpenMP a single CPU core can be used with a GPU which is not optimal. Note that with MPI multiple processes can be forced to use a single GPU, but this is typically inefficient. You need to set both C and C++ compilers that support OpenMP (CC and CXX environment variables, respectively) when using GPUs.
Call Stack (most recent call first):
  CMakeLists.txt:582 (gmx_gpu_setup)
```

...

```
-- Configuring done
-- Generating done
```

then when you immediately do

```
$ cmake ..
CMake Warning at cmake/gmxManageOpenCL.cmake:72 (message):
  To use GPU acceleration efficiently, mdrun requires OpenMP multi-threading. Without OpenMP a single CPU core can be used with a GPU which is not optimal. Note that with MPI multiple processes can be forced to use a single GPU, but this is typically inefficient. You need to set both C and C++ compilers that support OpenMP (CC and CXX environment variables, respectively) when using GPUs.
Call Stack (most recent call first):
  CMakeLists.txt:582 (gmx_gpu_setup)

-- Configuring done
-- Generating done
```

which you cannot even suppress with `-DGMX_OPENMP=off`. Such warnings made more sense when people might have had ancient gcc that didn't have openmp, but that's no longer a relevant consideration. IMO the right time to make observations about performance is during the run (to a log file). We could consider writing some CMake code to make suggestions at the end of the first run of cmake, but that needs buy in from multiple people prepared to write, review, and test the cmake code.

#### #13 - 12/19/2017 08:49 PM - Teemu Murtola

- Related to Bug #2357: `GMX_GPU=no` doesn't work if initially set to `auto` added

#### #14 - 12/20/2017 04:37 AM - Mark Abraham

Another example: Run cmake, detecting CUDA. Run ccmake to turn `GMX_GPU=off`. Run ccmake to turn `GMX_DOUBLE=on`, and get warned that GPU support is not available in double precision.

#### #15 - 12/20/2017 02:18 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#1985](#).  
Uploader: Aleksei lupinov ([a.yupinov@gmail.com](mailto:a.yupinov@gmail.com))  
Change-Id: gromacs~release-2018~l5cada97015ee94717ea6eb988b3a84a351f11293  
Gerrit URL: <https://gerrit.gromacs.org/7377>

#### #16 - 12/20/2017 06:49 PM - Szilárd Páll

Mark Abraham wrote:

More examples, this time using clang, where distros versions are often still without openmp support (e.g. clang-4.0.1 in ubuntu 17.10(:

[...]

then when you immediately do

[...]

which you cannot even suppress with `-DGMX_OPENMP=off`. Such warnings made more sense when people might have had ancient gcc that didn't have openmp, but that's no longer a relevant consideration. IMO the right time to make observations about performance is during the run (to a log file). We could consider writing some CMake code to make suggestions at the end of the first run of cmake, but that needs buy in from multiple people prepared to write, review, and test the cmake code.

I think this latter is mostly a dev-concern, so it would be nice to have, but for users not that important: most users configure and build once, so the message won't keep reappearing (and it's still relevant especially as these days it's rare that a compiler does not support OpenMP at all).

**#17 - 10/15/2018 05:21 PM - Mark Abraham**

- *Target version changed from 2019 to 2020*

We will probably need to rework the GPU build system some time, but strategy is unclear and resources to review my proposed changes are not available.

**#18 - 12/20/2019 12:07 PM - Paul Bauer**

- *Target version changed from 2020 to 2021*

any chance that this can be resolved at some point?