GROMACS - Bug #2635

Failing to detect GPUs should not write to terminals

09/04/2018 09:59 PM - Mark Abraham

Status:	Closed		
Priority:	Normal		
Assignee:	Mark Abraham		
Category:	mdrun		
Target version:	2019		
Affected version - extra info:		**Difficulty:**	uncategorized
Affected version:	2018.3		

Description

Per policy at #1505, we should not report failure to detect GPUs to the terminal when we have a GPU-enabled build running on a machine that lacks a driver.

On a CUDA-linked build on my laptop, there's no driver installed, and none found. But e.g.

```
$ gmx mdrun -rerun prod.trr -deffnm prod
                      :-) GROMACS - gmx mdrun, 2018.3 (-:

                            GROMACS is written by:
     Emile Apol      Rossen Apostolov      Paul Bauer     Herman J.C. Berendsen
    Par Bjelkmar    Aldert van Buuren   Rudi van Drunen     Anton Feenstra
  Gerrit Groenhof    Aleksei Iupinov   Christoph Junghans   Anca Hamuraru
 Vincent Hindriksen Dimitrios Karkoulis    Peter Kasson        Jiri Kraus
  Carsten Kutzner      Per Larsson      Justin A. Lemkul    Viveca Lindahl
  Magnus Lundborg   Pieter Meulenhoff    Erik Marklund      Teemu Murtola
    Szilard Pall       Sander Pronk      Roland Schulz     Alexey Shvetsov
   Michael Shirts     Alfons Sijbers     Peter Tieleman    Teemu Virolainen
 Christian Wennberg    Maarten Wolf
                           and the project leaders:
        Mark Abraham, Berk Hess, Erik Lindahl, and David van der Spoel

Copyright (c) 1991-2000, University of Groningen, The Netherlands.
Copyright (c) 2001-2017, The GROMACS development team at
Uppsala University, Stockholm University and
the Royal Institute of Technology, Sweden.
check out http://www.gromacs.org for more information.

GROMACS is free software; you can redistribute it and/or modify it
under the terms of the GNU Lesser General Public License
as published by the Free Software Foundation; either version 2.1
of the License, or (at your option) any later version.

GROMACS:      gmx mdrun, version 2018.3
Executable:   /opt/gromacs/2018.3/bin/gmx
Data prefix:  /opt/gromacs/2018.3
Working dir:  /home/mabraham/redmines/redmine-2634
Command line:
  gmx mdrun -rerun prod.trr -deffnm prod

NOTE: Detection of GPUs failed. The API reported:
            GROMACS cannot run tasks on a GPU.
Reading file prod.tpr, VERSION 2018.2 (single precision)
Changing nstlist from 10 to 100, rlist from 1 to 1
...
```

while the log file reports

```
...
```

```
CUDA driver:        0.0
CUDA runtime:       32.66

NOTE: Detection of GPUs failed. The API reported:
            GROMACS cannot run tasks on a GPU.

Running on 1 node with total 2 cores, 4 logical cores, 0 compatible GPUs
Hardware detected:
  CPU info:
    Vendor: Intel
    Brand:  Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz
    Family: 6   Model: 142   Stepping: 9
    Features: aes apic avx avx2 clfsh cmov cx8 cx16 f16c fma htt intel lahf mmx msr nonstop_tsc pc
id pclmuldq pdcm pdpe1gb popcnt pse rdrnd rdtscp sse2 sse3 sse4.1 sse4.2 ssse3 tdt x2apic
  Hardware topology: Full, with devices
    Sockets, cores, and logical processors:
      Socket  0: [   0   2] [   1   3]
    Numa nodes:
      Node  0 (8242876416 bytes mem):   0   1   2   3
      Latency:
               0
         0  1.00
    Caches:
      L1: 32768 bytes, linesize 64 bytes, assoc. 8, shared 2 ways
      L2: 262144 bytes, linesize 64 bytes, assoc. 4, shared 2 ways
      L3: 4194304 bytes, linesize 64 bytes, assoc. 16, shared 4 ways
    PCI devices:
      0000:00:02.0  Id: 8086:5916  Class: 0x0300  Numa: 0
      0000:3a:00.0  Id: 168c:003e  Class: 0x0280  Numa: 0
      0000:3c:00.0  Id: 1c5c:1283  Class: 0x0108  Numa: 0

++++ PLEASE READ AND CITE THE FOLLOWING REFERENCE ++++
...
```

We should remove that output from the terminal - it duplicates log file content, and we cannot infer that failure to find a usable GPU from a suitably configured GROMACS is critical or warning-level information, particularly while our build system can default to building for a GPU e.g. when a suitable CUDA library is found.

On login1, the situation is similar for the 2018.1 from the module, but apparently we made a small change already since then:

```
Command line:
  gmx mdrun -rerun prod.trr -s prod

NOTE: GPUs cannot be detected:
      No valid CUDA driver found
      Can not use GPU acceleration, will fall back to CPU kernels.
Reading file prod.tpr, VERSION 2018.2 (single precision)
```

Associated revisions

Revision 15b4b09f - 09/05/2018 04:58 PM - Mark Abraham

Be quieter when failing to detect GPUs

A normal mdrun user should not be informed on the
terminal that no GPUs could be detected. That information
is in the log file.

Fixes #2635

Change-Id: Idd254411c211c0fdef11d685ca1be2071aa6e281

History

#1 - 09/04/2018 11:58 PM - Szilárd Páll

```
NOTE: Detection of GPUs failed. The API reported:
            GROMACS cannot run tasks on a GPU.
```

That seems broken: if the API didn't report anything the message should be omitted; on the other hand, detection did not fail and the messages are IMO confused and confusing (a successful detection that reported no devices is not a failure).

I'm also on the opinion that removing this message even from the verbose output will be a regression (I can't imagine that the rolling counter or the note on the tpx version is more important than a note that a GPU build did not detect any GPUs).

#2 - 09/05/2018 01:04 AM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #2635.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: gromacs~master~Idd254411c211c0fdef11d685ca1be2071aa6e281
Gerrit URL: https://gerrit.gromacs.org/8307

#3 - 09/05/2018 01:44 AM - Mark Abraham

Szilárd Páll wrote:

> I'm also on the opinion that removing this message even from the verbose output will be a regression

Why? That this message is written to the terminal is contrary to the agreed policy. Are you saying the policy is wrong, or that this should be an exception to the policy, or that the proposed fix does not follow the policy?

> (I can't imagine that the rolling counter or the note on the tpx version is more important than a note that a GPU build did not detect any GPUs).

Nobody said they were more important. I reported and fixed a problem that I actually saw. If you've found others, please report them and/or fix them.

#4 - 09/05/2018 05:00 PM - Mark Abraham

- Status changed from Fix uploaded to Resolved

Applied in changeset 15b4b09fac08037a2cf0a41e49d45bafe4ff99ea.

#5 - 09/07/2018 06:09 PM - Szilárd Páll

- Status changed from Resolved to Feedback wanted

Mark Abraham wrote:

> Szilárd Páll wrote:
>
> > I'm also on the opinion that removing this message even from the verbose output will be a regression
>
> Why? That this message is written to the terminal is contrary to the agreed policy. Are you saying the policy is wrong, or that this should be an exception to the policy, or that the proposed fix does not follow the policy?

The policy is fine in general, but for many cases it is incomplete or simply not applied with enough care: the mdrun console UX degrades if the only goal in mind is to yank messages without considering the broader impact. E.g. as a result of the changes pushed in recently, certain console error and status messages have become impossible to interpret or outright confusing as these were constructed assuming a context that's now removed while these messages were left untouched.

Instead of just removing stuff, I'd expect that such proposals and related fixes come with a review/redesign of the console messages that relied on the output.

Case in point (using the current master):

```
$ gmx mdrun -v -gpu_id 1
[...]
GROMACS:      gmx mdrun, version 2019-dev-20180907-bcb0584
Executable:   /home/pszilard/projects/gromacs/gromacs-master/build_gcc73_cuda92/bin/gmx
Data prefix:  /home/pszilard/projects/gromacs/gromacs-master (source tree)
Working dir:  /home/pszilard/projects/gromacs/testing/grappas/grappa-045
Command line:
  gmx mdrun -v -gpu_id 0

-------------------------------------------------------
Program:     gmx mdrun, version 2019-dev-20180907-bcb0584
Source file: src/gromacs/mdrun/runner.cpp (line 487)

Fatal error:
You limited the set of compatible GPUs to a set that included ID #1, but that
```

```
ID is not for a compatible GPU. List only compatible GPUs.

For more information and tips for troubleshooting, please check the GROMACS
website at http://www.gromacs.org/Documentation/Errors
-------------------------------------------------------
```

This error is now unhelpful and in some cases significantly more confusing. It's not useful to give a convoluted explanation that's lacking specifics to a simple problem that boils down to either of these cases:

1. GPU#1 is not compatible/available/stable/etc.
2. there is no GPU#1
3. can't detect GPUs e.g. because of driver/runtime mismatch or because there are no devices at all.

The code that issues these messages is kept "simple" and the output less wordy by first issuing a detection output (here NOTE: Detection of GPUs failed. The API reported:) and upon encountering an error issuing the above message that refers to the former (implicitly or explicitly). As a result of the merged changed, this error is now more confusing than useful in both cases 2 and 3. (when there simply isn't a GPU#1).

The proposed change should IMO have come with extensions to provide context in the error messages that rely on the detection notes, e.g. the above would be understandable if it looked something like this:

```
$ gmx mdrun -v -gpu_id 1
[...]
-------------------------------------------------------
Program:     gmx mdrun, version 2019-dev-20180907-bcb0584
Source file: src/gromacs/mdrun/runner.cpp (line 487)

Fatal error:
You limited the set of compatible GPUs to a set that included ID #1, but
no GPUs were detected (GPU detection failed).

For more information and tips for troubleshooting, please check the GROMACS
website at http://www.gromacs.org/Documentation/Errors
-------------------------------------------------------
```

The error could also just tell the user to dig in the log file to figure out what caused the error, but that still doesn't undo the confusion caused (that did exist to some extent before, but was relatively easy to reconcile given the detection note a dozen or so lines above).

#6 - 09/07/2018 06:11 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #2635.
Uploader: Szilárd Páll (pall.szilard@gmail.com)
Change-Id: gromacs~master~I022efbff589f207d625f260ba9700a65dfa41e5d
Gerrit URL: https://gerrit.gromacs.org/8319

#7 - 09/12/2018 11:01 PM - Mark Abraham

Note in passing - I did introduce a spurious comma in this code in 8daff31c6ee88, which got found in one of Roland's printf fixing patches

#8 - 10/30/2018 08:45 AM - Erik Lindahl

Szilard: While there might be a point in reworking some other messages, the reason we have a policy is to avoid these things always turning into long discussions.

As we have all agreed, we cannot have a situation where each author independently decides what should go on the console and then reverts changes that try to implement the agreed policy unless we have a long discussion in each case about whether we should follow our policy and not follow it exactly and all agree in each single case.

The way we manage this is:

1) We gradually commit changes that implement the policy we have agreed on.

2) If that means some other messages are less clear we should try to fix them, ideally in the same commit, but the point of (1) is that we will have less information.

3) If somebody wants to write information on the console, bring up a discussion about it, and If - but only if - we all agree, it can stay there.

However, the argument that somebody things something could be useful does not mean everything is placed on indefinite hold before step 1 :-)

#9 - 10/30/2018 10:55 AM - Mark Abraham

I made https://gerrit.gromacs.org/#/c/7841/ for master five months ago, but never pushed people to review it. It moves some related user-input checking to one place, after we've done hardware checking and reported it. A good way to both implement the policy (ie quiet terminal by default) and be able to provide the desired context on the terminal for error reports is to save e.g. the results of GPU detection to a string, so that code like https://gerrit.gromacs.org/#/c/7841/4/src/gromacs/taskassignment/usergpuids.cpp doesn't need to duplicate it (and users get to see the information in one layout, which will hopefully become familiar). So perhaps a preliminary change to do the saving of GPU (or general hardware) detection strings would prepare the context better for my proposal at 7841.

#10 - 10/30/2018 03:27 PM - Szilárd Páll

Erik Lindahl wrote:

> Szilard: While there might be a point in reworking some other messages, the reason we have a policy is to avoid these things always turning into long discussions.

I did not think the policy in question trumps usability.

> 3) If somebody wants to write information on the console, bring up a discussion about it, and If - but only if - we all agree, it can stay there.

This is not about somebody wanting to write something to the console. I simply noted the inconsistency/usability issues introduced (and voiced the opinion that again, console output is completely removed with no consideration to verbosity levels either), tested many/most use-cases and provided feedback on what's broken. The proposed solution (the revert) was simply the minimal change that eliminates the regression introduced -- per policy (if an issue is introduced that's not caught in review, a revert of the said change is made to restore the original functionality).

@Mark: post-beta I'll look at 7841, I admittedly missed it. It'd be good if it had a ref to some redmine (this?).

#11 - 10/30/2018 04:45 PM - Mark Abraham

Szilárd Páll wrote:

> This is not about somebody wanting to write something to the console. I simply noted the inconsistency/usability issues introduced (and voiced the opinion that again, console output is completely removed with no consideration to verbosity levels either),

My original one line change doesn't prevent someone implementing that mdlog.info should write to the terminal in a verbose mode (if that is the policy). So there was nothing about verbosity level for me to consider there.

If you think other changes people have made in recent years were wrong, please propose policies and fixes. I'm sick of having the responsibility to get the team to agree on workable policies, and still having to debate all of history every time I fix things. Particularly when the code that is now possibly less usable was evolved from code written ages ago, without any policy or testing in place, and perhaps even pre-dates code review. If you want to be the UI maintainer, then go right ahead, but then I'll be the GPU developer and have a lot more fun :-).

> tested many/most use-cases and provided feedback on what's broken. The proposed solution (the revert) was simply the minimal change that eliminates the regression introduced -- per policy (if an issue is introduced that's not caught in review, a revert of the said change is made to restore the original functionality).

How would we have known this could have been thought to be a regression, since there are no policies (e.g. "that gmx mdrun always writes GPU detection results and diagnostics to the terminal, so that any subsequent error messages will have context") or tests in place of the old implementation? People need to be able to fix bugs without being held ransom to unknowns. Back in February, we could have negotiated that we need the above special case until we've improved the GPU log file reporting to the point where the output can be saved so it can be added to terminal only when needed for writing error messages. But nobody remembered that we needed it, so it was not in anybody's mind when I found this issue in September.

If you want us to have rigid implementation of policies about whether we auto-revert changes that break things, then we have to have rigid definition of what breaking things means e.g. a written statement of expectations that the team can agree to, or (better) a failing test. Otherwise, we're in a grey zone. In the cases you describe, restoring the GPU detection output does improve the user experience a bit. The issue I fixed does improve the user experience a bit in different cases. But the proper course of action is to actually work on our detection and reporting code, which is something that I expect you to remember when we're planning 2020 activities.

> @Mark: post-beta I'll look at 7841, I admittedly missed it. It'd be good if it had a ref to some redmine (this?).

I made it well before this Redmine and related gerrit changes, so doesn't ref it. It is relatable now, though.

#12 - 12/17/2018 02:34 PM - Paul Bauer

I think the discussion here should go on in a more general issue on how information should be logged or presented to the user. Is there a different issue that could be referenced, so we can resolve this one?

#13 - 12/21/2018 09:20 AM - Paul Bauer

- Status changed from Feedback wanted to Resolved

#14 - 12/21/2018 09:20 AM - Paul Bauer

- Status changed from Resolved to Closed