# GROMACS - Bug #2757

## mdrun refuses to start with .cpt if nsteps is -1 in .tpr

11/14/2018 12:56 PM - Carsten Kutzner

| | | | |
|---|---|---|---|
| **Status:** | Closed | | |
| **Priority:** | Normal | | |
| **Assignee:** | Mark Abraham | | |
| **Category:** | mdrun | | |
| **Target version:** | 2018.5 | | |
| **Affected version - extra info:** | | **Difficulty:** | uncategorized |
| **Affected version:** | 2018.4 | | |

**Description**

Should be reproducible with any .tpr that has nsteps -1 and a checkpoint file.

Error was introduced in commit 4dcb2a1aec05ca4fe50492007251ccd9e31b948a "Issue fatal error if checkpoint does not suit the .tpr". There is also a note in the added lines in checkpoint.cpp that we do not intend to support the use of mdrun -nsteps -1, but in this case the -1 comes from the .mdp file setting.

Command line:
mdrun_threads_AVX2_256 -v -pin on -nt 20 -noappend -nb gpu -pme gpu -maxh 48 -s topol.tpr -cpi state_08.cpt -cpo state_09.cpt

Error message is something like
---
Program: mdrun_threads_AVX2_256, version 2018.4-dev-20181112-3fd2d78
Source file: src/gromacs/fileio/checkpoint.cpp (line 2622)

Fatal error:
The input requested -1 steps, however the checkpoint file has already reached
step 188393400. The simulation will not proceed, because either your
simulation is already complete, or your combination of input files don't
match.

**Related issues:**

| | |
|---|---|
| Related to GROMACS - Bug #2717: mdrun runs infinitely when checkpoint file is... | **Closed** |
| Related to GROMACS - Task #1781: re-design benchmarking functionality | **Accepted** |
| Related to GROMACS - Task #2569: announce deprecations in GROMACS 2019 | **Closed** |

### Associated revisions

**Revision 61285613 - 11/15/2018 05:37 PM - Mark Abraham**

Fix checkpoint restart of tpr with infinite step count

The recent fix of #2717 did not account for the way a user's .tpr file
can require an infinite number of steps by using the special value
-1. Such special values are difficult to remember when maintaining the
code, so we should tend to avoid introducing them.

Fixes #2757

Change-Id: I6570c4f4e7d63b2375dbb595a514c9e709f18856

### History

**#1 - 11/15/2018 12:25 AM - Mark Abraham**

*- Related to Bug #2717: mdrun runs infinitely when checkpoint file is beyond the designated end point added*

**#2 - 11/15/2018 12:27 AM - Mark Abraham**

*- Related to Task #1781: re-design benchmarking functionality added*

**#3 - 11/15/2018 12:27 AM - Mark Abraham**

*- Related to Task #2569: announce deprecations in GROMACS 2019 added*

#### #4 - 11/15/2018 12:44 AM - Mark Abraham

Thanks for the report.

Clearly we don't have enough test coverage to support fancy stuff like mdrun -nsteps, or the .mdp feature that negative nsteps has a non-intuitive interpretation. Since those who want things like gmx mdrun -nsteps evidently don't work on the implementation of that (e.g. total lack of action from https://redmine.gromacs.org/issues/1781#note-50), then we will remove that immediately in master branch. If they want to reimplement that in gmx benchmark that's entirely on them. I'm sick of fixing bug reports on things I didn't design or implement, while those who insist we can't remove them just sit on the sidelines and work on fancy features and performance. Szilard, Roland, Berk, and others - either design tests, fix the code so it can be maintained by others, and review the maintenance commits, or it must go.

Similarly, all the fancy stuff like particular numerical values of .mdp options having special interpretations cannot be implemented in ways that are robust to maintenance. All of that must go. We can support the functionality if someone is prepared to implement the feature and support that with tests, but it must be a separate .mdp option.

#### #5 - 11/15/2018 04:43 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #2757.
Uploader: Paul Bauer (paul.bauer.q@gmail.com)
Change-Id: gromacs~release-2018~I7cbafb26d33827655fd94f67ea475e619498eabb
Gerrit URL: https://gerrit.gromacs.org/8706

#### #6 - 11/15/2018 04:56 PM - Erik Lindahl

While Paul's change might fix the acute problem, it adds to the already complex checking related to restarts. Not at all Paul's fault, but even the comment in the code about " we do not intend to support the use of mdrun -nsteps to circumvent this condition" makes it unclear to me whether it works or not, and what will happen when/if people try it.

It's also not at all clear to me what will happen for the various combinations of -maxh and -nsteps during checkpoint restarts, even after the change.

So, all in all, I strongly support Mark's conclusion.

#### #7 - 11/15/2018 05:08 PM - Paul Bauer

I'm also for dropping those feature(s) all together, because my change is in the end just some duct tape to keep things from falling apart.
I would prefer it more if there was one way to set the simulation length (the mdp file nsteps field) and nothing else. This would drastically simplify the issue at hand here.

#### #8 - 11/15/2018 05:09 PM - Paul Bauer

If there is no strong objection to it, I will upload a patch that removes -maxh and -nsteps in master, if Mark hasn't done so already :)

#### #9 - 11/15/2018 05:25 PM - Mark Abraham

Erik Lindahl wrote:

> While Paul's change might fix the acute problem, it adds to the already complex checking related to restarts. Not at all Paul's fault, but even the comment in the code about " we do not intend to support the use of mdrun -nsteps to circumvent this condition" makes it unclear to me whether it works or not, and what will happen when/if people try it.

I suspect the fix is much simpler than Paul's attempt, as the important part of Carsten's mdrun call is the nsteps=-1 in the .tpr file, not the use of maxh. The logic on line 2616 of checkpoint.cpp is incomplete because it doesn't check that ir->nsteps is non-negative.

That comment was part of my fix to #2717. It is indeed unclear at this point of the code that gmx mdrun -nsteps handling has not yet been invoked. An important part of any attempt to fix these things is that we stop doing such checks and handling in multiple places. The function for loading the checkpoint should just load the checkpoint :-)

> It's also not at all clear to me what will happen for the various combinations of -maxh and -nsteps during checkpoint restarts, even after the change.

The -maxh is intended more like a dynamic stopping criteria (determined implicitly by the user+simulation+environment), whereas the step counter is more static (ie. determined explicitly by the user). There too, we have historically had no developer docs describing the intent, or tests of the effects. Pascal did some very good work cleaning up aspects of the implementation (see the new StopHandler class in release-2019 branch).

The general problem is one of too much complexity. The individual things are useful in isolation, but the combination of them is hard to design, implement, understand, test, and maintain, and it is the latter that should drive whether we continue to have them.

#### #10 - 11/15/2018 05:39 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue #2757.
Uploader: Mark Abraham (mark.j.abraham@gmail.com)

Change-Id: gromacs~release-2018~I6570c4f4e7d63b2375dbb595a514c9e709f18856
Gerrit URL: https://gerrit.gromacs.org/8708

#### #11 - 11/15/2018 05:41 PM - Mark Abraham

Paul Bauer wrote:

> If there is no strong objection to it, I will upload a patch that removes -maxh and -nsteps in master, if Mark hasn't done so already :)

I think -maxh is useful to many users, well implemented, and has some tests (could be improved, no doubt). It fits nicely into the framework of "every now and then, check for some criteria and the do an operation" which we need for checkpointing and any analysis-driven dynamic stopping critera.

Removing -nsteps is something I strongly support.

We would still need to clean up how the init_step, simulation_part, nsteps, step, and steprel cluster of variables are managed. In particular, the checkpoint handling code should have nothing to do with that. I've been cleaning that up slowly for a few years, but it's difficult to understand how to fix anything while having to try to keep everything working.

#### #12 - 11/15/2018 05:41 PM - Mark Abraham

*- Status changed from New to Fix uploaded*

*- Assignee set to Mark Abraham*

#### #13 - 11/15/2018 08:00 PM - Mark Abraham

*- Status changed from Fix uploaded to Resolved*

Applied in changeset 6128561309d330490e80c17e24b67cf9551555ea.

#### #14 - 11/16/2018 09:22 AM - Paul Bauer

*- Status changed from Resolved to Closed*

I understand, maybe it is just that I haven't been using -maxh myself :)
I think the issue at hand is resolved, but the underlying problems with -nsteps still need to be taken care of.