GMX_MAX_MPI_THREADS has no implementation

07/02/2014 10:46 PM - Mark Abraham

Status: Closed
Priority: Low
Assignee: mdrun
Category: mdrun
Target version: 
Affected version: 4.6.x
Affected version: 5.0
Difficulty: uncategorized

Description
GMX_MAX_MPI_THREADS is supposed to be available to the user to limit the scope of the default behaviour of mdrun, so that the number of threads launched does not completely fill the machine, which might be nice on a user's desktop, etc.

There's no implementation, so we should remove its documentation and impact on error messages, or implement it.

Related issues:
Related to GROMACS - Bug #1543: complex/nbnxn_vsite regressiontests fails wit... Closed 06/30/2014

Associated revisions
Revision 054a2bc8 - 06/20/2015 08:08 PM - Erik Lindahl
Remove GMX_MAX_MPI_THREADS

This variable has been orpaned for quite a while, and it's a good example how we introduce bugs by introducing lots of ways to control the same functionality, and then we forget to update some of them later. Let's focus on the command line options as the single way to control the number of threads in Gromacs.

Fixes #1550.

Change-Id: I37a87125b9dcac36362b561418d787167bb64cec

History
#1 - 07/02/2014 10:51 PM - Roland Schulz
I don't see a point in this environment variable. The same can be achieved with -ntmpi or -nt.

#2 - 07/02/2014 11:14 PM - Mark Abraham
- Related to Bug #1543: complex/nbnxn_vsite regressiontests fails with GPUs added

#3 - 07/02/2014 11:16 PM - Mark Abraham
I'm not a fan of the environment variable either - the behaviour for choosing run-time settings is too complex and untested already. But it might have some use in the related issue #1543

#4 - 07/02/2014 11:43 PM - Szilárd Páll
Actually, I just remembered how did the GMX_MAX_MPI_THREADS get orphaned, it used to be called GMX_MAX_THREADS in 4.5 and it was supposed to be renamed, but instead it got removed.

Roland Schulz wrote:
I don't see a point in this environment variable. The same can be achieved with -ntmpi or -nt.

No, -nt and -ntmpi can only tell mdrun to run with exactly N threads which is not the same as at most N threads.

Let me repeat again the role is this variable (from the discussion at #1543):
The idea was to allow users to set this variable e.g. in their .bashrc in order to limit the number of tMPI threads spawned automatically without having to always type in the command line option. This is useful for people who use their laptop/workstation e.g. for minimization or equilibration, but want to
leave 1-2 cores available for other tasks.

Mark Abraham wrote:

    But it might have some use in the related issue #1543

I don't see any other way to correctly address the thread limit issue than allowing to impose through passing something to mdrun, be it a gmxrc file or an environment variable, or alternatively by checking first what hardware is there and then setting -nt N if the number of hardware threads detected is larger than N.

#5 - 07/02/2014 11:59 PM - Roland Schulz

If we change the meaning of "-nt/-ntmpi/-ntomp" as use maximum that many, than we the user could simply use an alias. And this change usually wouldn't change the behavior. Only in those cases where we currently print a fatal error (e.g. that OpenMP isn't supported) we would instead print a warning and instead just use less.

#6 - 07/03/2014 12:07 AM - Szilárd Páll

Roland Schulz wrote:

    If we change the meaning of "-nt/-ntmpi/-ntomp" as use maximum that many, than we the user could simply use an alias. And this change usually wouldn't change the behavior. Only in those cases where we currently print a fatal error (e.g. that OpenMP isn't supported) we would instead print a warning and instead just use less.

I don't think that's a good idea, off the top of my head for the following reasons:

- if the user wants to use N tMPI ranks, he/she should be able to without mdrun outsmarting her;
- the meaning of -ntomp was by design equivalent with OMP_NUM_THREADS;
- on some platforms sysconf calls don't work (correctly) and in those cases we would always force the user to use a single tMPI rank (and a single OpenMP thread if such an implementation would disregard OMP_NUM_THREADS).

#7 - 06/19/2015 09:33 AM - Gerrit Code Review Bot

Gerrit received a related patchset '2' for Issue #1550.
Uploader: Erik Lindahl (erik.lindahl@gmail.com)
Change-Id: I37a87125b9dcac36362b561418d787167bb64cec
Gerrit URL: https://gerrit.gromacs.org/4745

#8 - 06/19/2015 09:39 AM - Erik Lindahl

- Status changed from New to Fix uploaded

Removed the feature in the commit above. Looking through all open Redmine issues, there is a nasty pattern of a bunch of bugs being related to convenience features that somebody added, but a year later they cause bugs because we have too many execution paths or ways to set things. It's reasonably easy to write a shell script to accomplish this feature with the command line flags instead.

#9 - 06/22/2015 12:43 PM - Rossen Apostolov

- Status changed from Fix uploaded to Closed