GROMACS - Bug #1730

gmx compare does not compare all fields of a .tpr

05/12/2015 02:45 PM - Mark Abraham

Status:	New		
Priority:	Normal		
Assignee:			
Category:	core library		
Target version:	future		
Affected version - extra info:	all versions since 4, probably	**Difficulty:**	uncategorized
Affected version:	5.0.5		

Description

At least state->fep_state is not compared between .tpr files, because currently it is too easy to add fields to data structures without bumping all of the different kinds of init, cleanup, I/O and comparison routines. This has led to bug fix https://gerrit.gromacs.org/#q,Ibcee7bff1e090fb1991969c4562f44f056868a03,n,z

This is awkward to fix because (at least) fep_state has written uninitialized data to .tpr files for years, so the regressiontests probably can't compare new .tpr files with old ones (or old tpr files with a future fixed version of gmx compare). So we'd at least have to bump the .tpx version, so that we can compare fep_state only when that is a valid operation, and probably re-generate all regressiontests .tpr files. Needing to do such useless work is its own problem (#1587) - our primary check should be that given .mdp fragments produce given .tpr fragments rather than comparing old and new .tpr fragments.

Such a regressiontests re-generation is work for someone to do, and I think that only makes sense in the lead-up to a non-patch release. The downside is that someone doing gmx compare on a non-FE .tpr from before the fep_state initialization fix will compare the uninitialized vs the initialized-to-zero value in the fixed code. I think we can live with that.

It may not be sensible to have fep_state in the .tpr at all (see #1729), in which case this issue goes away.

Related issues:

Related to GROMACS - Task #1729: Resolve whether and how to resolve "state" v...	**New**	**05/12/2015**
Related to GROMACS - Task #1587: improve the configurability of regression tests	**New**	

Associated revisions

Revision 9fbe7e5b - 05/12/2015 03:31 PM - Mark Abraham

Fix uninitialized fields in grompp t_state

Allocating t_state on the stack and using an incomplete
pseudo-constructor means it is possible to write an uninitialized
value in fep_state field to the .tpr file, which we've been
doing. Found with Memory Sanitizer. Perhaps this behaviour lies behind
some of the strange behaviour that is periodically seen on Jenkins.

Fixed by allocating t_state on the heap, which might resolve other
issues, since snew() zeroes the memory as a side effect. Also
initialized fep_state field in init_state().

Refs #1729, #1730

Change-Id: Ibcee7bff1e090fb1991969c4562f44f056868a03

History

#1 - 05/12/2015 02:46 PM - Mark Abraham

- Related to Task #1729: Resolve whether and how to resolve "state" variables stored in .tpr added

#2 - 05/12/2015 02:46 PM - Mark Abraham

- Related to Task #1587: improve the configurability of regression tests added

#3 - 05/12/2015 02:50 PM - Gerrit Code Review Bot

Gerrit received a related patchset '2' for Issue #1730.

Uploader: Mark Abraham (mark.j.abraham@gmail.com)
Change-Id: Ibcee7bff1e090fb1991969c4562f44f056868a03
Gerrit URL: https://gerrit.gromacs.org/4575

#4 - 06/12/2015 09:33 PM - Erik Lindahl

- Target version changed from 5.1 to future

The immediate bug has been fixed in the branch leading to 5.1. Whether or when somebody looks into the general problem and fixes things once and for all remains to be seen in the future, so removing target 5.1.