

GROMACS - Bug #1004

Memory leaks in trajectory I/O proportional to size of the trajectory

09/13/2012 05:25 AM - Teemu Murtola

Status:	Closed	
Priority:	Normal	
Assignee:	Teemu Murtola	
Category:	core library	
Target version:	4.5.6	
Affected version - extra info:	4.5-4.5.5	Difficulty: uncategorized
Affected version:	4.5.5	

Description

In several text-based trajectory I/O routines, there are memory leaks that are proportional to the number of frames read/written (as well as to the number of unique strings in the file). At least some of these have been introduced when static variables have been changed to non-static (previously, the memory was allocated only once and referenced from the static variable until the program ended). For reading, the common problem is that `syntab.h` is used to store the strings read, and it is typically allocated anew and initialized for each frame. Need to still analyze whether this is possible to fix without reintroducing global variables or large changes in the trajectory I/O calls.

For PDB writing, there is a separate problem: `residuetypes.dat` is read and parsed each time a PDB frame is written to output, and the memory used for this is leaked. The memory leak should be straightforward to fix, but we really should not do the read/parse operation more than once, which requires more changes...

Found in <https:// Gerrit.gromacs.org/#/c/1427/>, where reading of `.gro` files and writing of `.pdb` files cause these leaks; other text-based formats are probably also affected.

Associated revisions

Revision 3b8e1cd9 - 09/13/2012 06:52 AM - Teemu Murtola

Fix some memory leaks.

Fixes part of #1004.

Change-Id: I22280999c5c3c40e8efae45461bd53bae8e2ff40

Revision 3b8e1cd9 - 09/13/2012 06:52 AM - Teemu Murtola

Fix some memory leaks.

Fixes part of #1004.

Change-Id: I22280999c5c3c40e8efae45461bd53bae8e2ff40

Revision ad406a3a - 09/16/2012 08:43 PM - Teemu Murtola

Fix insolidangle selections near poles.

- Correctly treat cases where a point is so close the one of the poles that it completely covers one or more bins in the zenith angle direction.
- Adjust `update_surface_bin()` to be more robust to rounding errors and easier to understand by simplifying the code used for wrapping angles to `[-pi, pi]` interval. Problems were triggered here by the first change.
- Update comments.

Fixes #1004.

Change-Id: I2707f775793fabb64ea197bdae5fbfe68d6a8933

Revision ad406a3a - 09/16/2012 08:43 PM - Teemu Murtola

Fix insolidangle selections near poles.

- Correctly treat cases where a point is so close the one of the poles that it completely covers one or more bins in the zenith angle direction.
- Adjust `update_surface_bin()` to be more robust to rounding errors and easier to understand by simplifying the code used for wrapping angles to $[-\pi, \pi]$ interval. Problems were triggered here by the first change.
- Update comments.

Fixes #1004.

Change-Id: I2707f775793fabb64ea197bdae5fbfe68d6a8933

History

#1 - 09/16/2012 12:27 PM - Teemu Murtola

The linked [3b8e1cd9](#) fixes the mentioned memory leaks in `.gro` reading and `.pdb` writing for 4.5.6. Other formats should also be checked.

#2 - 09/18/2012 12:46 PM - Teemu Murtola

Went through the occurrences of `t_syntab` in the source. Based on reviewing the code, both `.g96` and `.pdb` formats suffer from the same problem: when reading a trajectory in either of these formats, `t_syntab` is initialized anew on each frame in `read_g96_conf()` and `read_pdbfile()`, and is never freed. There are only a few calls to these functions, all from `confio.c`, `pdbio.c` or `trxio.c`, so it should be possible to fix this without affecting a lot of code. It is not possible to fix this without changing the interface to these functions, but previous fixes to `read_g96_conf()` have already changed the signature from 4.5.5, so user code will anyways be affected.

There are also smaller memory leaks for other formats in `read_stx_conf()` (also related `t_syntab` not being freed), but that should typically occur only once in a program, so it's not worth fixing before 5.0. In particular because it would require quite large changes, since `read_stx_conf()` is used quite a lot.

#3 - 09/18/2012 01:46 PM - Teemu Murtola

- *Status changed from New to Closed*
- *Assignee set to Teemu Murtola*
- *Target version set to 4.5.6*
- *Affected version - extra info changed from 4.5 and later to 4.5-4.5.5*

Actually, the `.g96` and `.pdb` trajectory reading does work, but the code is so convoluted that it was hard to see that from the outset... The `syntab` is still allocated for each frame, but it is not used since a specially prepared `t_atoms` structure is passed in. I think the issues worth fixing for 4.5.6 are now fixed. This code would benefit from a lot of clean-up in 5.0. In particular, there are so many overlapping interfaces to the system (and a lot of them are in installed headers) that is difficult to see interactions between the parts.

#4 - 07/17/2013 07:20 AM - Teemu Murtola

- *Category set to core library*
- *Affected version set to 4.5.5*