

## GROMACS - Task #2971

### Rework TPR reading to allow reading of raw bytes from disk and communication of complete information at setup time

06/05/2019 11:19 AM - Paul Bauer

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	Paul Bauer	
<b>Category:</b>	core library	
<b>Target version:</b>	2020-beta1	
<b>Difficulty:</b>	hard	
<b>Description</b>		
<p>While reading the TPR file, the fields are currently populated one by one while the raw bytes are read from disk. It would make sense to instead read the whole size of the TPR as a raw byte stream into a buffer and then populate the datastructures by reading from this buffer. This would also allow the communication of the whole TPR as one message during simulation setup, with the individual nodes populating the fields from the buffer instead of communicating the fields individually.</p> <p>To reach this point, the current TPR reading and writing code first needs to be adapted to use the serializer interface. This can be followed by changes that introduces a new version of the TPR format that contains information about the total number of bytes to read in the TPR file header. Old versions will still need to be read directly from disk to know the total number of bytes first before reading the information into the buffer. A final change can then communicate the byte buffer instead of the individual fields.</p>		
<b>Related issues:</b>		
Related to GROMACS - Task #1729: Resolve whether and how to resolve "state" v...	<b>New</b>	<b>05/12/2015</b>
Related to GROMACS - Bug #3084: gmx report-methods test unstable	<b>Closed</b>	

#### Associated revisions

##### Revision e887207f - 06/18/2019 07:45 PM - Paul Bauer

Extend ISerializer functionality

Add datatypes to ISerializer that are needed to read TPR files but were missing.

Refs #2971

Change-Id: I5d5e7f1f91c533a079cb287b018fa1d579c4f3f9

##### Revision 21d45dd2 - 06/19/2019 08:07 PM - Paul Bauer

Split tpr header reading reading from tpr body

Split the low level functions for TPR file header and file body reading into fully separate parts to allow reading the main part of the file without having to read the header again.

Also gave the header datastructure a new name in line with naming conventions and default initialized all fields.

Refs #2971

Change-Id: I110fb80cf19d9d2e59df1576e50c64806f532e00

##### Revision 1e6316f4 - 06/20/2019 01:07 PM - Paul Bauer

Use ISerializer for TPR file IO

Change all the function calls in do\_tpx and friends to use the ISerializer instead of the previous t\_fileio pointer. This is intended to prepare for the change where the datastructures get populated from a byte buffer instead of reading them from disk one by one.

Refs #2971

Change-Id: I9c2d51c4af0cad5a14da7026d58ecbe053e8efb7

**Revision 3836f527 - 09/18/2019 07:48 PM - Paul Bauer**

Read TPR file body in buffer

For now only reads the buffer and uses it to write new files.

Refs #2971

Change-Id: I77a18ca50e96486d688db8b0d7acdbedf29d613d

**Revision caf88a3a - 09/19/2019 11:04 AM - Paul Bauer**

Split up do\_tpx\_body functions

Will facilitate only communicating the parts of the TPR file needed on ranks other than master.

Refs #2971

Change-Id: Ia5a5fe4f1c9bda1340e0776a0a2d9e96a90d4d07

**Revision 6983f2be - 09/23/2019 12:17 PM - Paul Bauer**

Change MPI setup to communicate TPR as buffer

Changed the initial setup of nodes to communicate the full tpr file buffer instead of using the individual calls for the fields.

Now non-master nodes receive the inputrec and mtop and populate them themselves.

Refs #2971

Change-Id: Id4f3739a978ca507dacc45c78a8a75368cfe86fd

**Revision 2b209eb5 - 09/24/2019 02:10 PM - Paul Bauer**

Read in TPR char buffer as vector

Perform the I/O of the TPR char buffer as xdr\_vector operation instead of using single bytes.

Also use the xdr vector specialization for unsigned char and rvecs.

Refs #2971

Change-Id: I20534985fbdee8108792f676b3cb4264ab74c456

## History

---

**#1 - 06/05/2019 11:19 AM - Paul Bauer**

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

**#2 - 06/05/2019 03:24 PM - Paul Bauer**

- Private changed from Yes to No

**#3 - 06/07/2019 02:27 AM - Mark Abraham**

This will also remove the need to serialize the string symbol table. And make aspects of the code easier testable and faster to test.

**#4 - 06/25/2019 01:27 PM - Paul Bauer**

It was decided to not focus on this for GROMACS 2020.

The reason is that it might happen that the allocations needed for communicating the full TPR and populating a dummy state might exceed limits of available memory on hardware that has a low memory per possible hardware thread. The default setting of launching one (thread-) MPI thread per available hardware thread could then lead to cases where not enough memory available to allocate all datastructures.

Testing with a system of 648000 water molecules shows that when using 4 threads, about 25 MB are needed for each thread during maximum memory load, before the transitional datastructures are deallocated again.

**#5 - 09/18/2019 06:14 PM - Szilárd Páll**

Paul Bauer wrote:

It was decided to not focus on this for GROMACS 2020.

Has this decision been changed?

**#6 - 09/20/2019 01:06 PM - Mark Abraham**

- *Status changed from New to In Progress*

- *Target version set to 2020-beta2*

Szilárd Páll wrote:

Paul Bauer wrote:

It was decided to not focus on this for GROMACS 2020.

Has this decision been changed?

ToolsTest sometimes fails because tpr operations take forever and/or too much memory on the TSAN build. Berk observed that fixing some far-too-heavy operations on vectors stored as xdr would help with this, so Paul resurrected some past work. Not essential for the first beta however.

**#7 - 09/27/2019 05:27 PM - Paul Bauer**

- *Status changed from In Progress to Resolved*

- *Target version changed from 2020-beta2 to 2020-beta1*

**#8 - 09/27/2019 05:27 PM - Paul Bauer**

- *Status changed from Resolved to Closed*

**#9 - 10/11/2019 03:36 PM - Mark Abraham**

However ToolsTest continues to fail from time to time

**#10 - 10/12/2019 08:01 PM - Mark Abraham**

- *Related to Bug #3084: gmx report-methods test unstable added*

**#11 - 10/12/2019 08:02 PM - Mark Abraham**

Mark Abraham wrote:

However ToolsTest continues to fail from time to time

Perhaps resolved by [cef36d09e64b3f5e3a0248722d8da8d7f1cc584d](https://github.com/gromacs/gromacs/pull/3640)

**#12 - 10/14/2019 01:31 PM - Paul Bauer**

This wasn't supposed to resolve the issue with TPR generation, this is always limited by the way grompp assigns parameters and not (I think) by file access.