

GROMACS - Task #2859

Change ArrayRef iterator type from pointer to std::iterator

02/07/2019 10:03 AM - Paul Bauer

Status:	Resolved
Priority:	Normal
Assignee:	
Category:	core library
Target version:	
Difficulty:	hard
Description	
<p>Currently a number of coding bugs are possible when comparing iterators from ArrayRef to those from other std::containers, as ArrayRef::iterator is actual a pointer to the data in the container.</p> <p>Changing the iterator type would avoid bugs where those iterators are compared to pointers or nullptr, but would add complexity to the code.</p> <p>Another option would be using std::span for ArrayRef itself but this would either need to be natively implemented or imported from an existing implementation.</p>	

Associated revisions

Revision 8b901fad - 02/11/2019 05:23 PM - Roland Schulz

Remove EmptyArrayRef

Make it more aligned with std::span.

Almost everywhere it wasn't needed anyhow.

Only exception: ternary conditional operator. But there the type was already explicit in all but one case and that one case (src/gromacs/domdec/distribute.cpp) was confusing because of multiple implicit conversions.

Related #2859

Change-Id: I0b61abdc2e60285a7ca17e3bdc7e53cb72c5cf6a

Revision c95f8de2 - 06/26/2020 06:03 PM - Roland Schulz

Change ArrayRef to use a proper iterator

Fixes #2859

History

#1 - 02/07/2019 01:22 PM - Paul Bauer

- *Tracker changed from Feature to Task*

I think this is better described as a task

#2 - 02/07/2019 05:57 PM - Mark Abraham

My suggestion is that we fix the ArrayRef iterator (I think that is easy). Then we should get a c++14 std::span like thing into compat, or similar. I did have some WIP on gerrit for a c++11 span some time ago, but by now we'd do better to import a new one.

#3 - 02/07/2019 07:12 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2859](#).

Uploader: Roland Schulz (roland.schulz@intel.com)

Change-Id: gromacs~master~I80e2b6a75bca0973f550e0ab8fa62c622c330f90

Gerrit URL: <https://gerrit.gromacs.org/9109>

#4 - 02/07/2019 07:15 PM - Roland Schulz

- *Status changed from New to Fix uploaded*

Instead of importing a new std::span we probably should explore making span a simple 1D template alias of mdspace. If we don't discover any significant disadvantage this would avoid duplication.

#5 - 02/07/2019 08:02 PM - Roland Schulz

We need to fix this for begin/end for mdspan too (added [here](#)). Independent of whether we use mdspan for span or not.

#6 - 02/08/2019 11:00 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2859](#).

Uploader: Roland Schulz (roland.schulz@intel.com)

Change-Id: gromacs~master~l0b61abdc2e60285a7ca17e3bdc7e53cb72c5cf6a

Gerrit URL: <https://gerrit.gromacs.org/9119>

#7 - 06/26/2020 06:15 PM - Roland Schulz

- *Status changed from Fix uploaded to Resolved*

Applied in changeset [c95f8de2b627bb8e26e3fc29bf99722ec909b640](#).