GROMACS - Task #3040

Refactor Restraint module

07/18/2019 03:59 PM - Eric Irrgang

Status:	New
Priority:	Normal
Assignee:	
Category:	
Target version:	
Difficulty:	uncategorized

Description

The Restraint module introduced in GROMACS 2019 is more of a special case than it was intended to be. It should be normalized with respect to the other MD modules. However, some design consensus is needed in several respects.

In GROMACS 2019, an object implementing a restraint is added to the primary MdRunner in each process with a public member function of MdRunner, before mdrunner() is called. This member function adds an object (via a IRestraintPotential handle) to the list of restraints managed on each CPU process. Each process has one RestraintManger implementation object, though each MdRunner instance (in the case of tMPI) has a unique handle to it. The RestraintManager then takes care of registering an IForceProvider for each restraint object it is managing.

Some of the reasons for this architecture are alleviated by work under #2945 and related issues.

It would seem to make sense that, with the Restraint module in the GROMACS source, all restraints should have their force calculations combined under a single IForceProvider interaction with a single Restraint module object.

However, the Restraint module and RestraintManager were intended as temporary shims until MD modules could be acquired abstractly by the MdRunner through the SimulationContext or MdRunnerBuilder. We could move the RestraintManager to a public interface of the Restraint module that it is configured by the client and then passed to the MdRunnerBuilder, which then makes the manager available to the RestraintModule in an initialization process like that of the other MD modules. Or we could do away with the Restraint module as a non-conformant internal MD module and just have the MdRunner request IMDModule handles from the Context.

We might assume the Restraint module will remain as a built-in module, though, to support optimization of special cases of IForceProviders, in which case it can be restructured to

1. Provide a single IForceProvider for all restraints operating under the Restraint API.
2. Remove the need for MdRunner::addPotential with either a public interface to the module or with a protocol for the module to initialize itself with function objects provided by the SimulationContext.

Note that externally implemented restraints depend on the installed headers gromacs/restraintpotential.h and gmxapi/md/mdmodule.h. Clients use the gmxapi::addSessionRestraint() function to attach a restraint when launching a simulation. gmx::MdRunner::addPotential() is used internally to implement gmxapi::addSessionRestraint().

Restraint's internal facilities also need to be modernized with respect to LocalAtomSets and internal library data structures.

Related issues:

Related to GROMACS - Feature #3038: Improvements to MD plugin development env...	**New**
Related to GROMACS - Task #2945: Give MdModules access to simulation resource...	**Closed**
Related to GROMACS - Task #2375: Clarify execution phases for MD simulation	**New**
Related to GROMACS - Task #2590: Essential Dynamics as module providing forces	**New**
Related to GROMACS - Task #2492: implement force calculation via ForceProvide...	**New**
Related to GROMACS - Task #2623: Allow extensible MDModules and forceProviders.	**Closed**

History

#1 - 07/18/2019 03:59 PM - Eric Irrgang

- Related to Feature #3038: Improvements to MD plugin development environment added

#2 - 07/18/2019 04:29 PM - Eric Irrgang

- Related to Task #2945: Give MdModules access to simulation resources (e.g. atom selection manager or communication infrastructure) added

#3 - 07/18/2019 04:30 PM - Eric Irrgang

- Related to Task #2375: Clarify execution phases for MD simulation added

#4 - 07/18/2019 04:32 PM - Eric Irrgang

- Related to Task #2590: Essential Dynamics as module providing forces added

#5 - 07/18/2019 04:32 PM - Eric Irrgang

- Related to Task #2492: implement force calculation via ForceProviders containing collections of IForceProvider added

#6 - 07/18/2019 04:33 PM - Eric Irrgang

- Related to Task #2623: Allow extensible MDModules and forceProviders. added