

GROMACS - Feature #2587

Feature # 2585 (Resolved): Infrastructure supporting external API

Provide Context (e.g. to runner code) to manage client and runtime environment

07/24/2018 03:55 PM - Eric Irrgang

Status: In Progress	
Priority: Normal	
Assignee:	
Category:	
Target version:	
Difficulty: uncategorized	
Description	
gmxapi milestone 5 as described in #2585	
Proposal:	
Instead of relying on a global variable for ProgramContext, explicitly require client code to pass a context object to library code. This allows a context to be configured differently depending on the use case of the client code, as well as to capture details of the computing environment that could be better abstracted.	
This issue covers restructuring of GROMACS to support an expanded role of client-managed "context". Specific enhancements are deferred to downstream issues to be linked from here.	
Near term goals of the expanded role for the Context include	
<ul style="list-style-type: none">• Support run-time extensibility of MD code, such as by providing a source of factory functions from which MD module implementation code can be retrieved.• ownership of highest level communicator(s) and source of simulation-level communicators• Translation of ANSI C signals to GROMACS signalling facilities• Handling of logging and status messaging	
Longer term goals include	
<ul style="list-style-type: none">• Negotiating allocation of computing resources• I/O abstractions and filesystem-decoupling• Interfaces for workflow-level check-pointing	
Completion:	
This issue requires that code like gmx::Mdrunner (and the call stack below) receive the context from the client code. A context implementation should not <i>require</i> global variables, and as much as possible globals should be removed from the current ProgramContext.	
Provisions for runtime extensibility:	
<ul style="list-style-type: none">• Library-facing Context interface provides factory functions for, e.g.<ul style="list-style-type: none">- MDModules- Communications- Logger- Messenger- Reference data- I/O• Simple C API allows registering resource providers named in simple hierarchical schema.	
Related issues:	
Blocked by GROMACS - Feature #2605: Library access to MD runner	Closed

Associated revisions

Revision f3ff9147 - 10/03/2018 01:33 PM - Eric Irrgang

Move mdrun mainFunction to client code.

supports gmxapi milestone 4, described at #2605

API clients need to be able to initialize and run MD simulations with calls from separate code blocks or even translation units. Also, we need clear distinctions between a pre-launch master Mdrunner and the non-master Mdrunner threads.

In this change:

- Insert a Builder on which to develop the distinction between user interface and implementation code. User interface is handled in Director code, while the Builder and code further down the mdrun call stack should be user interface agnostic (i.e. not CLI-centric).
- Introduces a Context object to hold some resources. (see milestone 5 and issue #2587)
- Prepare for more stateful Mdrunner objects, documenting resources in need of clearer management, describing future use cases and possible implementation details.
- Make Mdrunner non-copyable and not constructable by clients. Worker threads can initialize a new Mdrunner from the master instance with `cloneOnSpawnedThread`.
- Clarify parameter setting for spawned Mdrunner threads.
- Begin to encapsulate filename options available to client code from the actual client code.
- Remove extraneous variables ``nfile`` and ``fnm``.

Later changes will:

- Separate Mdrunner into Launcher and Worker at `mdrunner()` call so that tMPI runner does not look reentrant.
- Hide data and clarify ownership / modernize memory management / clarify the subjects of the Builder ``add`` operations.
- Clarify separation of user interface from API parameters.
- Flesh out `SimulationContext` and related classes.

Refs #2605

Change-Id: I1db1d34b07ec0f8ba5f246ab763c74ad9eafe8f3

Revision 7a90bcdb - 10/03/2018 05:27 PM - Eric Irrgang

Add `gmxapi::Context`.

The Context holds the details necessary to launch work in a given environment. A runtime session is launched using the configured details of a Context implementation and the object bindings configured by the user (managed by the Context).

Part of a sequence of changes introducing gmxapi classes System, Context, Session, and Workflow.

Supports gmxapi milestone 5

Refs: #2587

Change-Id: I75baed9b5c856f284bc2c2370ef284319e95f13e

Revision e39949c7 - 10/01/2019 07:24 AM - Mark Abraham

Removed dependency on `commrec` of `mdrun` setup

Changes no functionality.

Setup is now parameterized directly on `MPI_COMM_WORLD`, which we will want later for letting library-based callers pass in an `MPI_Communicator`. This permits `commrec` to be initialized later, once the threads have been spawned for the thread-MPI ranks.

The initialization of multi-simulations moves from `LegacyMdrunOptions` to `SimulationContext`, which is more appropriate for ensemble-parallelism established directly by the user.

Before the decision about the duty of a rank, there is no difference between `MASTER` and `SIMMASTER`, so several calls to macros taking a `t_commrec` pointer are replaced by booleans. Introduced

findsSimulationMasterRank to compute that value. This eliminates early use of t_commrec that has necessitated other hacks and workarounds.

Removed redundant check for replica exchange when the number of multi simulations is less than two, because gmx_multisim_t constructor already prohibits that.

Resolves several TODO items and improves modularity, too.

Refs #2587, #2605, #3081

Change-Id: I48bd3b713bc181b5c1e4cbcd648706a9f00eab96

History

#1 - 08/09/2018 01:20 PM - Eric Irrgang

- Description updated

#2 - 08/09/2018 02:34 PM - Eric Irrgang

- Blocked by Feature #2605: Library access to MD runner added

#3 - 08/21/2018 08:01 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2587](#).

Uploader: M. Eric Irrgang (ericirrgang@gmail.com)

Change-Id: gromacs~master~I1db1d34b07ec0f8ba5f246ab763c74ad9eafe8f3

Gerrit URL: <https://gerrit.gromacs.org/8213>

#4 - 08/21/2018 08:14 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2587](#).

Uploader: M. Eric Irrgang (ericirrgang@gmail.com)

Change-Id: gromacs~master~I337df3ce0b16d0ba07e1178cc9ee36dcf78948da

Gerrit URL: <https://gerrit.gromacs.org/8214>

#5 - 08/27/2018 02:18 PM - Gerrit Code Review Bot

Gerrit received a related patchset '3' for Issue [#2587](#).

Uploader: M. Eric Irrgang (ericirrgang@gmail.com)

Change-Id: gromacs~master~I75baed9b5c856f284bc2c2370ef284319e95f13e

Gerrit URL: <https://gerrit.gromacs.org/8240>

#6 - 10/15/2018 03:34 PM - Eric Irrgang

- Status changed from New to In Progress

For GROMACS 2019, we have a gmxapi::Context to manage the environment of client code, and a gmx::SimulationContext to manage the resources provided by the client to the gmx::Mdrunner. In GROMACS 2020, we will need to refine the client and library facets of these facilities, improve resource ownership semantics / encapsulation, add functionality, and more fully migrate to the new resource management scheme.