

## GROMACS - Task #2395

### break up commrec

01/31/2018 03:34 PM - Mark Abraham

<b>Status:</b>	In Progress	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	Mark Abraham	
<b>Category:</b>	mdrun	
<b>Target version:</b>	2021-infrastructure-stable	
<b>Difficulty:</b>	uncategorized	
<b>Description</b>		
t_commrec currently handles lots of things and goes lots of places. We should break it into pieces as suggested in the checklist.		
This will make it much easier to deploy modules that might be exposed by an API, or implement standard interfaces for command-line options, etc.		
<b>Subtasks:</b>		
Feature # 3307: General interface for communication between simulation ranks		<b>New</b>
<b>Related issues:</b>		
Related to GROMACS - Bug #3241: Bonded GPU kernel launched in the wrong strea...		<b>Closed</b>

### Associated revisions

#### Revision f8937dc1 - 02/12/2018 05:42 PM - Mark Abraham

Remove commrec from hardware detection

This is preparatory refactoring for aspects of #2395

The OpenCL logic was ineffective, because duty is not yet decided, and anyway we might soon want the detection on PME-only ranks.

Replaced the thread-MPI single-rank assertion with a more direct implementation.

Minimized contents of detecthardware.h

Refs #2395

Change-Id: I03af65805bd14515a0213d511ae8cdb627c2f05c

#### Revision 65aaa064 - 12/11/2019 08:06 PM - Mark Abraham

Document DOMAINDECOMP correctly

It is likely there are numerous cases where this is used mistakenly when havePPDomainDecomposition expresses the real intent. If so, runs with 1 PP and 1 PME rank may have buggy behaviour.

Refs #2395

Change-Id: I07be73a6c690887b3043140a2a78ae6fe6bb17f1

#### Revision 59e622e1 - 04/07/2020 09:10 PM - Pascal Merz

Require explicit MPI\_COMM for gmx\_bcast and gmx\_barrier

This changes gmx\_bcast and gmx\_barrier to take the MPI communicator explicitly instead of taking a pointer to t\_commrec and using mpi\_comm\_mygroup. This also allows to remove gmx\_bcast\_sim and leave the responsibility of passing the right communicator to the caller.

This is a first step in breaking up t\_commrec. These functions are the subset of low-level networking functions which are used before domain decomposition (and hence PP/PME ranks) is set up.

Refs #2395

#### Revision 0a48bcd0 - 04/22/2020 07:21 PM - Pascal Merz

Make `init_dires` independent of `t_commrec`

`init_disres` was requesting a full pointer to the `commrec`, but only uses a single communicator and checks for master rank and whether the run is parallel. This information is now passed in explicitly, simplifying the planned splitting of `t_commrec`.

Note that passing a `nullptr` for `commrec` was (mis)used by `gmx_disre` only - effectively signalling that `init_disres` was called from an analysis tool and not from `mdrun`. This has been made explicit.

Refs #2395

#### Revision ff03803b - 04/23/2020 07:10 PM - Pascal Merz

Make `boxdeformation` independent of `t_commrec`

`boxdeformation` requested a pointer to the full `t_commrec`, but only used on communicator and information on whether the current rank is master and whether the simulation is run in parallel. This has been made explicit, simplifying subsequent changes splitting up `t_commrec`.

Refs #2395

#### Revision 5734027b - 07/09/2020 02:08 PM - Pascal Merz

Divide default communicator from DD communicators

The communicators `mpi_comm_mysim` and `mpi_comm_mygroup` inside `t_commrec` got initialized in `init_commrec` (to `MPI_COMM_WORLD` if no `multisim`, to a subset otherwise). These communicators were then used in subsequent setup work, before they got reassigned during the construction of the `DDBuilder` object and the construction of the actual domain decomposition object. Effectively, this means that the same communicators (and, hence, identical function calls) do very different things depending on whether they get used before or after the setup of domain decomposition. It also means that before DD set up, `mpi_comm_mysim` and `mpi_comm_mygroup` are **identical**.

This change introduces an additional communicator within `t_commrec`, `mpiDefaultCommunicator`, which helps to make these implicit assumptions explicit. Consequently, this also redefines `PAR`, `MASTER`, and `SIMMASTER`.

This change will allow to move the `sim` and `group` communicators, which are now only created at DD time, into the DD object, logically separating the DD object from `t_commrec`.

Refs #2395

#### Revision e0f481ae - 07/15/2020 07:09 PM - Mark Abraham

Divide default communicator from DD communicators

The communicators `mpi_comm_mysim` and `mpi_comm_mygroup` inside `t_commrec` got initialized in `init_commrec` (to `MPI_COMM_WORLD` if no `multisim`, to a subset otherwise). These communicators were then used in subsequent setup work, before they got reassigned during the construction of the `DDBuilder` object and the construction of the actual domain decomposition object. Effectively, this means that the same communicators (and, hence, identical function calls) do very different things depending on whether they get used before or after the setup of domain decomposition. It also means that before DD set up, `mpi_comm_mysim` and `mpi_comm_mygroup` are **identical**.

This change introduces an additional communicator within `t_commrec`, `mpiDefaultCommunicator`, which helps to make these implicit assumptions explicit. Consequently, this also redefines `PAR`, `MASTER`, and `SIMMASTER`.

This change will allow to move the sim and group communicators, which are now only created at DD time, into the DD object, logically separating the DD object from t\_commrec.

Refs #2395

## History

---

### #1 - 02/01/2018 01:40 PM - Mark Abraham

[f746a4a4aedb76995](#) already started on this effort

### #2 - 02/01/2018 02:22 PM - Gerrit Code Review Bot

Gerrit received a related patchset '1' for Issue [#2395](#).  
Uploader: Mark Abraham ([mark.j.abraham@gmail.com](mailto:mark.j.abraham@gmail.com))  
Change-Id: gromacs~master~l03af65805bd14515a0213d511ae8cdb627c2f05c  
Gerrit URL: <https://gerrit.gromacs.org/7531>

### #3 - 03/29/2018 01:09 PM - Mark Abraham

[4868388f24ecee03d75d](#) and [9a2e38a91c0621d2ecbf1](#) also made progress here

### #4 - 03/29/2018 01:12 PM - Mark Abraham

- Description updated  
- Status changed from New to In Progress

### #5 - 09/19/2018 03:13 PM - Mark Abraham

- Target version changed from 2019 to 2020

### #6 - 12/02/2019 01:24 PM - Paul Bauer

- Target version changed from 2020 to 2021-infrastructure-stable

### #7 - 12/11/2019 03:37 PM - Mark Abraham

I suspect most devs don't realise that DOMAINDECOMP(cr) is true when there is 1 PP and 1 PME rank, so we should go through the uses and replace them by what is actually intended, which is mostly havePPDomainDecomposition.

And document DOMAINDECOMP(cr) correctly.

### #8 - 12/15/2019 11:51 AM - Mark Abraham

Mark Abraham wrote:

I suspect most devs don't realise that DOMAINDECOMP(cr) is true when there is 1 PP and 1 PME rank, so we should go through the uses and replace them by what is actually intended, which is mostly havePPDomainDecomposition.

And document DOMAINDECOMP(cr) correctly.

Eg [#3241](#)

### #9 - 12/15/2019 11:51 AM - Mark Abraham

- Related to Bug [#3241](#): Bonded GPU kernel launched in the wrong stream with 1 PP + 1 PME rank added