GROMACS - Bug #433

replica exchange+constant pressure broken with one cpu per replica

06/11/2010 02:32 PM - Floris Buelens

Status:	Closed		
Priority:	Normal		
Assignee:	Erik Lindahl		
Category:	mdrun		
Target version:	4.0		
Affected version - extra info:		**Difficulty:**	uncategorized
Affected version:			

Description

When running replica exchange at constant pressure, the exchange routine (exchange_state) exchanges box size information with the calls

```
exchange_rvecs(ms,b,state->box,DIM);
  exchange_rvecs(ms,b,state->box_rel,DIM);
  exchange_rvecs(ms,b,state->boxv,DIM);
  exchange_rvecs(ms,b,state->pres_prev,DIM);
```

In this call, 'state' in repl_ex.c corresponds to 'state_global' from do_md ('state_local' in repl_ex.c corresponds to 'state' in do_md - I find this a bit cryptic).

This means that on exchange, box information is exchanged in do_md's 'state_global', but not in do_md's 'state'. Running with more than one CPU per replica, this is resolved with the following call:

```
2557       if (bExchanged && PAR)
2558       {
2559          if (DOMAINDECOMP)
2560          {
2561             dd_partition_system(fplog,step,cr,TRUE,1,
2562                                 state_global,top_global,ir,
2563                                 state,&f,mdatoms,top,fr,
2564                                 vsite,shellfc,constr,
2565                                 nrnb,wcycle,FALSE);
2566          }
2567          else
2568          {
2569             bcast_state(cr,state,FALSE);
2570          }
2571       }
```

and everything is ok (with domain decomp. at least, haven't tested PD). But with one CPU per replica (PAR evaluates false) the new box size doesn't get copied to 'state' and box size and coordinates are mismatched, which leads to clashes and a crash on the next step.
I fixed this with:

```
else if (bExchanged) {
      copy_mat(state_global->box,state->box);
      copy_mat(state_global->box_rel,state->box_rel);
      copy_mat(state_global->boxv,state->boxv);
      copy_mat(state_global->pres_prev,state->pres_prev);
      for(i=0; i&lt;state-&gt;ngtc; i++)
      {
          state->nosehoover_xi[i]  = state_global->nosehoover_xi[i];
          state->therm_integral[i] = state_global->therm_integral[i];
      }
    }
```

straight after the line 2571 close-brace above and it seems to work.

Only tested on 4.0.5 but can't see any relevant changes in the latest code. Also most likely only relevant to people using the the (unfinished?) code for exchange between free energy replicas and different lambdas.

History

#1 - 06/16/2010 09:27 AM - Berk Hess

I fixed this for 4.5.

Berk