Replica-exchange simulations of NPT ensemble with a single CPU core per replica have unreasonable box-size behavior (see Box-X_of_t-single-original.png). This seems sometimes to cause the simulations to crash. This does not happen with multi-CPU assigned to each replica (see Box-X_of_t-multi-original.png). I attached the input tpr files. I use the version 4.6.1, but the version 4.6.3 also have the same problem (as far as I check).

I have found a (potential) solution as shown below to escape this behavior (see Box-X_of_t-modified.png). However, I can not evaluate its validity...

I guess the unreasonable behavior is due to improper box-size exchanges, though my coding ability does not allow my full understanding of GROMACS codes.

Here is my guess:
The box-sizes and their velocity updated by pcoupling is first stored in the t_state structure named state in do_md. The t_state structures of state and state_global (in do_md) are passed into replica_exchange as state_local and state, respectively. At this point state and state_global are not the same; state_global seems to have older box-sizes, probably as suggested in Bug #1191. The routine replica_exchange exchanges the information in state, which corresponds to state_global in do_md. Thus, the routine seems to exchange old box-sizes, not updated ones. Perhaps, the synchronizing process between the state and state_global does not work well in NPT ensemble with a single CPU core per replica...

My tentative solution is do the opposite-way copy of line 1380 of repl_ex.c:
Override state in replica_exchange by state_local.

```cpp
username$ diff -u  (original)repl_ex.c  (modified)replex.c
--- (original)repl_ex.c
+++ (modified)replex.c
@ -877,7 +877,8 @
     dpV = (beta[ap]*re->pres[ap]-beta[bp]*re->pres[bp])*(Vol[b]-Vol[a])/PRESFAC;
     if (bPrint)
         {
-            fprintf(fplog, "  dpV = %10.3e  d = %10.3e\n", dpV, delta + dpV);
+            //fprintf(fplog, "  dpV = %10.3e  d = %10.3e\n", dpV, delta + dpV);
+            fprintf(fplog, "  dpV = %10.3e  d = %10.3e\n", dpV, delta + dpV);
         }
     delta += dpV;
 }
@ -1343,6 +1344,10 @
     pd_collect_state(cr, state);
 }
+    else{
+/* This part is added by TNAGAI as tentative solution */
+    copy_state_nonatomdata(state_local,state);
+    }

if (MASTER(cr))
{...
```
P.S. There is a trivial typo in repl_ex.c, which confuses demux.pl.
P.S. I wish this would also help fix Bug #1245.

Related issues:
Related to Gromacs - Bug #1245: Bug in REMD with particle decomposition

Associated revisions
Revision 93a3bc69 - 10/29/2013 07:09 PM - Berk Hess
corrected volume with serial NPT replica exchange

Replica exchange with replicas run in serial would only update
x and v, not the other state data. This gave incorrect volumes
with NPT replica exchange.
Fixes #1362
Change-Id: lb726fbb75e800c624ef6131e76a5d4e408b9c

History
#1 - 10/28/2013 03:15 PM - Berk Hess
- Status changed from New to Fix uploaded
- Priority changed from Normal to High
- Target version changed from 4.6.x to 4.6.4

I think your fix is correct, I uploaded it to gerrit.

#2 - 10/29/2013 07:09 PM - Mark Abraham
- Status changed from Fix uploaded to Resolved

#3 - 10/29/2013 07:10 PM - Berk Hess
- % Done changed from 0 to 100

Applied in changeset 93a3bc696acab4caeb4ca8e04f15ac091b465ca7f588.

#4 - 11/04/2013 03:32 AM - Tetsuro NAGAI
Does anyone has idea whether the bug affects NVT-ensemble REMD simulations?

#5 - 11/12/2013 12:02 PM - Berk Hess
Since with NVT the volume is that same for all replicas and there is no pressure coupling, those issues are irrelevant for NVT.
The only issue is that for serial NVT replica exchange runs the nose-hoover state was not transferred. In practice I don't think this is an issue, unless
you have a very small system with very frequent exchanges.

#6 - 11/12/2013 09:20 AM - Tetsuro NAGAI
Thank you very much for your comments.
I hope so; the effect of an incorrect nose-hoover state has little effects on results.

However, from (perhaps very) pessimistic viewpoint, I may have to mention two facts.
There is an article suggesting that replica-exchange-attempt frequency is much better[*],
and thus one may employ very frequent exchanges.
I guess those who use serial version should simulate relatively small systems.
(Otherwise, one will use parallel version for large systems...)

But anyway, it will be alright!! Thanks.