Let mdrun dump coordinates with non-finite energy

My simulation keeps hitting total potential energy is -nan, error which is a new feature from gromacs 2016 branches:

Program: mdrun_mpi, version 2016.3
Source file: src/gromacs/mdlib/sim_util.cpp (line 777)
MPI rank: 25 (out of 112)

Fatal error:
The total potential energy is -nan, which is not finite. The LJ and electrostatic contributions to the energy are -104484 and -1258.37, respectively. A non-finite potential energy can be caused by overlapping interactions in bonded interactions or very large or NaN coordinate values. Usually this is caused by a badly or non-equilibrated initial configuration or incorrect interactions or parameters in the topology.

I can't find anything obviously wrong with my simulation trajectory. i.e if things are blowing up i have no way to assess what the problem is, it would be great if the error could dump the co-ordinates/ or in which instance this is happening.

As far as this system is concerned it is well equilibrated, I am not sure if this is reproducible in the 2016.4 branch but i am going to try.

Suggest attaching input files/command line

Suggest attaching input files/command line

The simulation is a replica exchange between 28 replicas in NVT ensemble.
gromacs version 2016.3 compiled with GCC 5.3.0 and MVAPICH2

The command line is:
mpirun -n 28 -ppn 7 -env MV2_CPU_BINDING_POLICY bunch mdrun_mpi -pin on -rdd 1.5 -multi 28 -plumed -replex 10000 -s tpr/topol.tpr -x traj.xtc -db no

-nan occurs usually after one of the swaps somewhere along the trajectory. There is no pattern to when and between which window it occurs.

The weird thing is i started two duplicates and only one of them crashed. So it is very hard to pin point what is causing the issue. Hence my request that including more information about the error would be helpful.
It looks like you are not running a stock version of GROMACS, but one modified with Plumed. It is not unlikely that the issue is caused by Plumed modifications. What is Plumed doing in this case?
Note that we do not want to debug Plumed issues.

#4 - 11/30/2017 10:34 AM - Berk Hess
- Affected version changed from 2016.4 to 2016.3

#5 - 11/30/2017 10:40 AM - Berk Hess
- Tracker changed from Bug to Feature
- Affected version deleted (2016.3)

I realized now that you seem to have intended this as a feature request: dumping the coordinates on a non-finite energy error. That indeed seems useful. Note that such a feature would be implemented in the master branch. I don't know when Plumed will support the 2018 release.

#6 - 11/30/2017 03:42 PM - Flaviyan Jerome Irudayanathan
Hi Berk,
Yes this would be useful feature.
For my current case I think i am fine as one of the duplicates did not encounter this error.

#7 - 12/18/2017 10:53 AM - Berk Hess
- Subject changed from total potential energy is -nan with REMD and Plumed to Let mdrun dump coordinates with non-finite energy
- Status changed from New to Accepted
- Target version set to 2019

#8 - 12/25/2017 12:36 AM - Erik Lindahl
Although it might be helpful when debugging, we also need to learn to be much more conservative with introducing new features, so I'm not convinced this is worth the extra maintenance cost. Dumping things in a single run might not be complex, but then we also need to handle how replicas have been swapped, and how to signal dumps in parallel runs, or if we are doing non-REMD multi-runs.

We should also gradually move to make Gromacs a well-behaved library, and then it is not acceptable for a library to simply call exit(), but the error should be propagated back through exceptions. Then it would also be my preference to handle coordinate writing as part of catching that exception, rather than adding code in the lowest-level routine that writes raw data in the present directory (which would mean even more code to clean up later :-(.

This also seems related to #2224.

#9 - 12/26/2017 06:12 AM - Mark Abraham
We certainly need to do a better job with our error handling (including coordinating that in parallel). Catching the existing SimulationInstabilityError exception could lead to a suitable diagnostic dump, which is worth having because it probably replaces several special cases we currently have. But there's plenty of infrastructure to deal with first.

#10 - 10/03/2018 09:47 PM - Mark Abraham
- Priority changed from Normal to Low
- Target version changed from 2019 to future

This is unlikely to be implemented any time soon.