Amber20 update 9

Webpage

http://ambermd.org/

Version

Amber20 update 9, AmberTools 20 update 15

Build Environment

- Intel Parallel Studio 2017 Update8 (MPI only)
- GCC 7.3.1 (devtoolset-7)
- CUDA 11.1 Update 1

Files Required

- Amber20.tar.bz2
- AmberTools20.tar.bz2
- (Amber20 update.1-9 & AmberTools20 update.1-15; obtained during the build procedure below)

Build Procedure

```
#/bin/sh

VERSION=20
TOOLSVERSION=20

INSTALL_DIR="/local/apl/lx/amber20-up9"
TARBALL_DIR="/home/users/${USER}/Software/AMBER/20"

PARALLEL=12

#------------------------------------------------------------------------
module purge
module load intel_parallelstudio/2017update8
module load scl/devtoolset-7
module load cuda/11.1

export AMBERHOME=${INSTALL_DIR}
export CUDA_HOME="/local/apl/lx/cuda-11.1"

export LANG=C
export LC_ALL=C

# install directory has to be prepared before running this script
if [ ! -d ${AMBERHOME} ]; then
  echo "Create ${AMBERHOME} before running this script."
  exit 1
fi

# the install directory must be empty
if [ "$(ls -A ${AMBERHOME})" ]; then
  echo "Target directory ${AMBERHOME} not empty"
  exit 2
fi

ulimit -s unlimited

# prep files
cd ${AMBERHOME}
bunzip2 -c ${TARBALL_DIR}/Amber${VERSION}.tar.bz2 | tar xf -
bunzip2 -c ${TARBALL_DIR}/AmberTools${TOOLSVERSION}.tar.bz2 | tar xf -
```

```
mv amber${VERSION}_src/* .
rmdir amber${VERSION}_src

# install python first. otherwise, update_amber failed to connect ambermd.org
./AmberTools/src/configure_python
AMBER_PYTHON=$AMBERHOME/bin/amber.python

# apply patches and update AmberTools
echo y | $AMBER_PYTHON ./update_amber --upgrade
$AMBER_PYTHON ./update_amber --update

echo "[GPU serial edition (two versions)]"
LANG=C ./configure --no-updates -cuda gnu
make -j${PARALLEL} install && make clean

echo "[GPU parallel edition (two versions)]"
LANG=C ./configure --no-updates -mpi -cuda gnu
make -j${PARALLEL} install && make clean

echo "[CPU serial edition]"
LANG=C ./configure --no-updates gnu
make -j${PARALLEL} install && make clean

echo "[CPU openmp edition]"
LANG=C ./configure --no-updates -openmp gnu
make -j${PARALLEL} install && make clean

echo "[CPU parallel edition]"
LANG=C ./configure --no-updates -mpi gnu
make -j${PARALLEL} install && make clean

# run tests
. ${AMBERHOME}/amber.sh
cd ${AMBERHOME}

# parallel tests first
export DO_PARALLEL="mpirun -np 2"

make test.parallel && make clean.test
make test.cuda_parallel && make clean.test # DPFP
cd test; ./test_amber_cuda_parallel.sh SPFP; make clean; cd ../

export DO_PARALLEL="mpirun -np 4"
cd test; make test.parallel.4proc; make clean; cd ../

unset DO_PARALLEL

# openmp tests
make test.openmp && make clean.test

# serial tests
make test.serial && make clean.test
make test.cuda_serial && make clean.test # DPFP
cd test; ./test_amber_cuda_serial.sh SPFP; make clean; cd ../

cd ${AMBERHOME}
chmod 700 src
```

Tests

- Only minor numerical errors were found.
- (Error on CUDA version of GAMD found in former build (update0) no longer exists. Thank you for the fix!)

Notes

- Built and tested on ccgpup (P100 frontend node).
  - (Recently, access to external sites from ccgpup via http/https is allowed.)
- No significant changes in performance of pmemd.cuda from the previous build (update0). (Performance on V100 might be very slightly (< 1%) worse than previous build...)
- On P100, pmemd.cuda of amber18-bf12 shows better performance than this one.
- No verified python3.

(Notes about cmake version)

- (NOTE: we decided not to use cmake this time due to the error on pbsa_cuda_cg test; this error does not happen for "configure" version.)
- If python3 is used for miniconda (configure_python -v 3), we met an error during the build process (python2 was invoked somehow).
- Build procedure and related flags seem to be differencent from "configure" version.
  - In case of "configure", sander API is disabled for cuda version. However, in case of "cmake", sander API is enabled for cuda version.
  - There might be some other differences?
- "pbsa_cuda_cg" test of "test_at_cuda" failed for cmake version. This error does not happen for "configure" version.
  - Non-negligible errors happen. (e.g. large deviation of EPB, "Iterations required" number is zero, value next to "Convergence achieved" seems curious as seen in below. (Similar errors can be found in bc5 and bc10 diff.)
  - There are no problems for all the other tests.
  - We also tested with gcc7+cuda-11.1, gcc7-cuda10.1, and gcc6+cuda-9.1, but was in vain. (Not tested with gcc-8 or intel compilers)
  - (So far we couln't understand what is wrong.)
    - The sander API issue described above is not related this. (-DBUILD_SANDER_API=FALSE upon cmake didn't change anything.)
    - -DBLA_VENDOR=Generic doesn't change anything, either. (openblas will be used if this flag omitted)
    - caused by mischoice of some cuda related library?

```
possible FAILURE: (ignored)  check mdout.1a93_B.p22.min_bc2.dif
/local/apl/lx/amber20-up9/AmberTools/test/pbsa_cuda_cg
929c929
<    Iterations required       :   220
>    Iterations required       :     0
931c931
<    Norm of the residual vector:      0.1317175924778
>    Norm of the residual vector:      0.
932c932
<    Convergence achieved      : 9.6113021224978365E-5
>    Convergence achieved      : 1.1278324955777715E-44
933c933
< Total surface charge      -1.9750
> Total surface charge      -2.0000
934c934
< Reaction field energy   -982.8577
> Reaction field energy   -805.5799
936c936
< Etot   =    -2949.8108  EKtot   =       0.  EPtot     =       0.
> Etot   =    -2772.5330  EKtot   =       0.  EPtot     =       0.
936c936
< Etot   =    -2949.8108  EKtot   =       0.  EPtot     =       0.
> Etot   =    -2772.5330  EKtot   =       0.  EPtot     =       0.
939c939
< EELEC  =    -1786.2464  EPB    =    -982.8577  RESTRAINT  =       0.
> EELEC  =    -1786.2464  EPB    =    -805.5799  RESTRAINT  =       0.
945c945
< Etot   =    -2949.8108  EKtot   =       0.  EPtot     =       0.
> Etot   =    -2772.5330  EKtot   =       0.  EPtot     =       0.
948c948
< EELEC  =    -1786.2464  EPB    =    -982.8577  RESTRAINT  =       0.
> EELEC  =    -1786.2464  EPB    =    -805.5799  RESTRAINT  =       0.
### Maximum absolute error in matching lines = 2.20e+02 at line 929 field 4
### Maximum relative error in matching lines = 8.52e+39 at line 932 field 4
```

cmake flags for GPU serial version:

(AMBER_PYTHON points miniconda python (prepared beforehand), CUDA_HOME is /local/apl/lx/cuda-(version).)

```
cmake .. \
    -DCOMPILER=GNU \
    -DMPI=FALSE \
    -DCUDA=TRUE \
    -DINSTALL_TESTS=FALSE \
    -DDOWNLOAD_MINICONDA=FALSE \
    -DPYTHON_EXECUTABLE=${AMBER_PYTHON} \
    -DCUDA_TOOLKIT_ROOT_DIR=${CUDA_HOME} \
    -DCHECK_UPDATES=FALSE
```