Amber18-bf1 for LX

Webpage

http://ambermd.org/

Version

Amber 18-bf1 + AmberTools 18-bf2

Build Environment

- Intel Compiler 17.0.4.196
- Intel MKL 2017 update 3
- Intel MPI 2017.0.3
- CUDA 9.1.85
- Python 2.7

Files Required

- Amber18.tar.bz2
- AmberTools18.tar.bz2
- (Amber18 update.1; downloaded in the following script)
- (AmberTools18 update.1-2; downloaded in the following script)

Build Procedure

```
#!/bin/sh

VERSION=18
TOOLSVERSION=18

INSTALL_DIR="/local/apl/lx/amber18-bf1"
TARBALL_DIR="/home/users/${USER}"

export AMBERHOME=${INSTALL_DIR}
# cuda 9.1
export CUDA_HOME="/local/apl/lx/cuda-9.1"

PARALLEL=12

#------------------------------------------------------------------------

# 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${VERSION}/* .
rmdir amber${VERSION}
```

```
# apply patches if exists
./update_amber --update
# configure python separately (miniconda)
AmberTools/src/configure_python
$AMBERHOME/bin/amber.conda install mkl-rt --yes

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

echo "[GPU parallel edition (three versions)]"
./configure --no-updates -mpi -cuda gnu
make -j${PARALLEL} install && make clean
# tests of GPU versions will be done elsewhere

# load intel17; intel18 is not officially supported
. /local/apl/lx/intel2017update4/bin/compilervars.sh intel64

echo "[CPU serial edition]"
LANG=C ./configure --no-updates -mkl intel
make -j${PARALLEL} install
. ${AMBERHOME}/amber.sh
make test.serial
make clean

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

echo "[CPU parallel edition]"
LANG=C ./configure --no-updates -intelmpi -mkl intel
make -j${PARALLEL} install
export DO_PARALLEL="mpirun -np 2"
make test.parallel
export DO_PARALLEL="mpirun -np 4"
cd test && make test.parallel.4proc

cd $AMBERHOME
make clean && chmod 700 src
```

Notices

- Files are installed in /local/apl/lx/amber18-bf1.
- Sample files can be found in samples/.
- Environment setting scripts (amber.sh, amber.csh) are in /local/apl/lx/amber18 directory. Please note that these scripts do not include cuda-9.1 settings.
- You have to apply CUDA-9.1 settings when you use cuda-enabled binaries (such as pmemd.cuda_*). Please take a look at the gpu job samples in samples/ directory.
- Test logs are available in logs/ directory. All the tests excluding cuda ones were performed at a frontend node.
    - Some of python tests have failed. Please be careful if you use this functionality.