

Amber18-bf12

Webpage

<http://ambermd.org/>

Version

Amber 18-bf12 + AmberTools 18-bf13

Build Environment

- Intel Compiler 17.0.8
- Intel MKL 2017 update 4
- Intel MPI 2017.0.4
- CUDA 9.1.85
- Python 2.7

Files Required

- Amber18.tar.bz2
- AmberTools18.tar.bz2
- (Amber18 update.1-12; get in the script below)
- (AmberTools18 update.1-13; get in the script below)

Build Procedure

```
#!/bin/sh

VERSION=18
TOOLSVERSION=18

INSTALL_DIR="/local/apl/lx/amber18-bf12"
TARBALL_DIR="/home/users/${USER}"

PARALLEL=12

#-----
module purge
module load intel_parallelstudio/2017update8
module load cuda/9.1

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

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}/* .
```

```
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
```

```
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
```

Notes

- GPU version is available both for P100 and V100. Please note that [bf11-volta](#) version might show better performance on V100.
- module name: `amber/18/bugfix12`
- Files are installed in `/local/apl/lx/amber18-bf12` directory.
- Samples are available under `samples/` directory.
- Environment setting script (`amber.sh`, `amber.csh`) can be found in `/local/apl/lx/amber18-bf12` directory. Note that those scripts do not set `CUDA-9.1` paths.
- `CUDA-9.1` setting is necessary if you want to use `CUDA` version binaries. GPU version samples in `samples/` directory may be useful.
- You can find test logs in `logs/` directory. All the tests except for `CUDA` versions are performed at one of the front-end nodes. `CUDA` version is tested on a V100 node.