

Amber18-bf12

ウェブページ

<http://ambermd.org/>

バージョン

Amber 18-bf12 + AmberTools 18-bf13

ビルド環境

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

ビルドに必要なファイル

- Amber18.tar.bz2
- AmberTools18.tar.bz2
- (Amber18 update.1-12; スクリプト内で取得)
- (AmberTools18 update.1-13; スクリプト内で取得)

ビルド手順

```
#!/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
```

注意

- GPU版は P100, V100 のどちらでも利用可能です。V100 での速度については [bf11-volta](#) の方が少し速い可能性があります。
- module 名は `amber/18/bugfix12` です。
- ファイルは `/local/apl/lx/amber18-bf12` 以下にインストールされています。
- サンプルは `samples/` 以下にあります。
- 環境設定スクリプト(`amber.sh`, `amber.csh`)は `/local/apl/lx/amber18-bf12` 直下にあります。CUDA-9.1のパス設定はこのスクリプトでは行われなことに注意してください。
- CUDA版のバイナリを使う際には、CUDA-9.1のパスを明示的に指定する必要があります。samples/以下にあるgpu用サンプルを参考にしてください
- テストのログはlogs/以下に置いてあります。cuda以外のテストはフロントエンドノードで行っています。