

Quantum Espresso 6.7 with GPU support

ウェブページ

<https://www.quantum-espresso.org/>
<https://gitlab.com/QEF/q-e-gpu/-/releases>

バージョン

6.7-gpu

ビルド環境

- PGI 20.4
- MKL 2019.0.5 (intel 2019 update 5)
- CUDA 10.1

ビルドに必要なファイル

- q-e-gpu-qe-gpu-6.7.tar.gz
- openmpi-4.0.2.tar.bz2
- (/local/apl/lx/pbs14 以下の PBS Pro 関連ファイル)

ビルド手順

```
#!/bin/sh

VERSION=6.7
FULLVER=${VERSION}
BASEDIR=/home/users/${USER}/Software/QE/${VERSION}-gpu
TARBALL=${BASEDIR}/q-e-gpu-qe-gpu-${FULLVER}.tar.gz
INSTDIR=/local/apl/lx/espresso67-gpu

# openmpi
WORKDIR=/work/users/${USER}
OMPIVER=4.0.2
OMPITARBALL=/home/users/${USER}/Software/OpenMPI/${OMPIVER}/openmpi-${OMPIVER}.tar.bz2
OMPIROOT=${INSTDIR}/openmpi-${OMPIVER}
PBSROOT=/local/apl/lx/pbs14

PARALLEL=12

# -----
umask 0022

module purge
module load pgi/20.4
module load mkl/2019.0.5

export LANG=C
export LC_ALL=C

ulimit -s unlimited

# build openmpi first
cd ${WORKDIR}
if [ -d openmpi-${OMPIVER} ]; then
  mv openmpi-${OMPIVER} openmpi-erase
  rm -rf openmpi-erase &
fi

tar jxf ${OMPITARBALL}
cd openmpi-${OMPIVER}

mkdir rccs && cd rccs
```

```

CC=pgcc CXX=pgc++ FC=pgfortran \
  ./configure --prefix=${OMPIROOT} \
    --with-tm=${PBSROOT} \
    --enable-mpi-cxx \
    --enable-mpi1-compatibility \
    --with-psm2
make -j ${PARALLEL} && make install && make check

# openmpi setting
export OMPI_MCA_btl_openib_allow_ib=1
export CPATH="${OMPIROOT}/include:${CPATH}"
export LIBRARY_PATH="${OMPIROOT}/lib:${LIBRARY_PATH}"
export LD_LIBRARY_PATH="${OMPIROOT}/lib:${LD_LIBRARY_PATH}"
export PATH="${OMPIROOT}/bin:${PATH}"

# qe build
cd ${INSTDIR}
if [ -d q-e-gpu-qe-gpu-${FULLVER} ]; then
  mv q-e-gpu-qe-gpu-${FULLVER} q-e-gpu-qe-gpu-erase
  rm -rf q-e-gpu-qe-gpu-erase &
fi

tar xzf ${TARBALL}
cd q-e-gpu-qe-gpu-${FULLVER}
mv *.[a-zA-Z]* ../
cd ../ && rmdir q-e-gpu-qe-gpu-${FULLVER}

export MPIF90=mpif90
export MPIFC=mpif90
export MPIF77=mpif90
export MPICC=mpicc
export MPICXX=mpicxx

# --with-cuda should point cuda bundled with pgi... but i couldn't do it...
FC=pgfortran F90=pgfortran F77=pgfortran CC=pgcc CXX=pgc++ \
  ./configure --enable-openmp \
    --enable-parallel \
    --with-cuda=/local/apl/lx/cuda-10.1 \
    --with-cuda-cc=60 \
    --with-cuda-runtime=10.1

# force to add cc70 support
sed -i -e "s/cc60/cc60,cc70/" make.inc

make -j${PARALLEL} pw cp
cd test-suite

make run-tests-pw-serial
make run-tests-cp-serial
make clean
make run-tests-pw-parallel
make run-tests-cp-parallel
cd ..

```

メモ

- openmpi の atomic_cmpset* のテストでエラーが出ていますが、QE のテストは全て成功しています
- pw についてはかなりの部分が GPU 対応になっているようです
- (GPU 版の cp については現時点ではあまり実用的ではないと思われます)
 - fft などかなり限定された箇所のみ GPU 対応になっているようです
 - conjugate gradient (cg) は現時点ではサポートされていないようです
- OpenMP 有効の条件でビルドしています
- P100 と V100 のどちらでも利用できます
- MPI 並列時は、個々のプロセスに 1 枚の GPU が割り当てられます
 - 複数のプロセスが一つの GPU を共有することは可能ですが、効率が下がる可能性があります。

- MPI 並列時は /local/apl/lx/espresso67-gpu/openmpi-4.0.2 以下の mpirun をご利用下さい。(サンプルもそのように作ってあります)
- GPU に対応していない計算もあるかもしれません。公式情報を確認したり、実際に実行するなどして(アウトプットの最後のタイミングからある程度判断できると思われます)ご確認ください。
- 小さい計算では GPU 版を使うメリットは薄いと思われます。
 - 大きすぎる計算を実行すると(おそらくはGPUの)メモリ不足でクラッシュすることもあります。その場合は MPI 並列で複数の GPU を使うようにすればエラーを回避できるかもしれません。