Quantum ESPRESSO 5.3.0 Installation Best Practices

HPC ADVISORY COUNCIL
NETWORK OF EXPERTISE
BEST PRACTICES

1. Introduction:

The following best practices document is provided as courtesy of the HPC Advisory Council.

2. Application Description:

Quantum ESPRESSO is an integrated suite of computer codes for electronic-structure calculations and materials modeling at the nanoscale. It is based on density-functional theory, plane waves, and pseudopotentials (both norm-conserving and ultrasoft).

3. Version Information:

Download Quantum ESPRESSO v5.3.0 (or newer) at:
http://www.quantum-espresso.org

4. Prerequisites:

The instructions from this best practice have been tested on the following configuration:
Hardware:

- Dell PowerEdge R730 32-node (1024-core) "Thor" cluster.
- Dual-Socket 16-Core Intel E5-2697v4 @ 2.60 GHz CPUs
- Mellanox ConnectX-4 EDR InfiniBand adapters
- Mellanox SwitchX SB7700 VPI InfiniBand and Ethernet switches

OS and software:

- RHEL 7.2, MLNX_OFED_LINUX-3.4-1.0.0.0 InfiniBand SW stack
- MPI: Mellanox HPC-X v1.7.0-405
- Compilers: Intel Composer 2016
- Application: Quantum ESPRESSO 5.3.0 and elpa-2015.11.001
- Benchmarks:
  - Unified European Application Benchmark Suite (UEABS) DEISA pw benchmark Test Case A, AUSURF112 - Gold surface (112 atoms) DEISA pw benchmark
  - Benchmark input cases can be downloaded from PRACE:
  - http://www.prace-ri.eu/ueabs/#Quantum_Espresso

5. Installation

5.1 Building ELPA

The following script is used for compiling and installing ELPA:

```
#!/bin/bash
module purge
module load intel/compiler/2016.4.258

WHICHMPI=intelmpi
WHICHMPI=openmpi
if [ "$WHICHMPI" == "intelmpi" ]; then
        module load intel/impi/5.1.3.258
        export I_MPI_CC=icc
        export I_MPI_CXX=icpc
        export I_MPI_FC=ifort
        export I_MPI_F90=ifort
elif [ "$WHICHMPI" == "openmpi" ]; then
```

```
        module load hpcx/icc-2016
        export OMPI_MPICC=icc
        export OMPI_MPICXX=icpc
        export OMPI_MPIFC=ifort
        export OMPI_MPIF90=ifort
fi

BASE=$PWD
./configure --prefix=$BASE/../espresso-5.3.0/ELPA-$WHICHMPI \
        SCALAPACK_LDFLAGS="-L$MKLROOT/lib/intel64 -lmkl_scalapack_lp64 -
lmkl_intel_lp64 \
                        -lmkl_sequential -lmkl_core -lmkl_blacs_${WHICHMPI}_lp64 \
                        -lpthread -lm -Wl,-rpath,$MKLROOT/lib/intel64" \
        SCALAPACK_FCFLAGS="-I$MKLROOT/include/intel64/lp64" \
        CFLAGS="-O2 -xCORE-AVX2" CXXFLAGS="-O2 -xCORE-AVX2" FCFLAGS="-O2 -xCORE-AVX2"
\
        CC=mpicc CXX=mpicxx FC=mpif90 F90=mpif90

make clean
make -j 16
make install
```

5.2 Building Quantum ESPRESSO

The following makefile is used for compiling Quantum ESPRESSO (note that there is a patch file at the end of this document):

```
#!/bin/bash
module purge
module load intel/compiler/2016.4.258

WHICHMPI=intelmpi
WHICHMPI=openmpi
if [ "$WHICHMPI" == "intelmpi" ]; then
        module load intel/impi/5.1.3.258
        export I_MPI_CC=icc
        export I_MPI_CXX=icpc
        export I_MPI_FC=ifort
        export I_MPI_F90=ifort
elif [ "$WHICHMPI" == "openmpi" ]; then
        module load hpcx/icc-2016
        export OMPI_MPICC=icc
        export OMPI_MPICXX=icpc
        export OMPI_MPIFC=ifort
        export OMPI_MPIF90=ifort
fi

BASE=$PWD
make distclean
rm -f make.sys
./configure --enable-parallel --prefix=$BASE/bin-$WHICHMPI \
        --with-elpa=$BASE/ELPA-${WHICHMPI}  \
        --with-scalapack=intel  \
        CFLAGS="-O3 -I$MKLROOT/include/fftw -xCORE-AVX2" CXXFLAGS="-O3 -
I$MKLROOT/include/fftw -xCORE-AVX2" \
        FCFLAGS="-O3 -I$MKLROOT/include/fftw -xCORE-AVX2" F90FLAGS="-O3 -
I$MKLROOT/include/fftw -xCORE-AVX2" \
        CC=mpicc CXX=mpicxx FC=mpif90 F90=mpif90

sed "s/mkl_blacs_openmpi_lp64/mkl_blacs_${WHICHMPI}_lp64/g" -i make.sys
sed "s/mkl_blacs_intelmpi_lp64/mkl_blacs_${WHICHMPI}_lp64/g" -i make.sys
```

```
sed "s/D__FFTW/D__FFTW3/g" -i make.sys
sed "s/FFT_LIBS       =.*/FFT_LIBS       = -mkl/g" -i make.sys
sed "s/LDFLAGS        =.*/LDFLAGS        =/g" -i make.sys
sed "s/include\/elpa/include\/elpa-2015.11.001/g" -i make.sys
sed "s/type f/type f | grep -v mpi/g" -i Makefile

patch -f -p0 < espresso-drv.patch

rm -f $BASE/bin-$WHICHMPI
make pw -j 32
make install
```

6. Running QUANTUM ESPRESSO

The input data 'ausurf.in' is staged on /dev/shm of each node.

Note: The "-nd" value usually is best when $NP/2 or $NP/4.

Running using HPC-X:

```
$ module load intel/compiler/2016.4.258
$ module load hpcx/icc-2016
$ mpirun -np 1024 -bind-to core -mca btl_sm_use_knem 1 -mca plm rsh -mca
coll_fca_enable 0 -mca coll_hcoll_enable 1 -mca coll_hcoll_np 0 -x
HCOLL_ENABLE_MCAST_ALL=1 -x HCOLL_MCAST_NP=1 -x HCOLL_CONTEXT_CACHE_ENABLE=1 -mca pml
yalla -mca mtl_mxm_np 0 -x MXM_TLS=ud,shm,self -x MXM_RDMA_PORTS=mlx5_0:1 -mca
btl_openib_if_include mlx5_0:1 -x MALLOC_MMAP_MAX_=0 -x MALLOC_TRIM_THRESHOLD_=-1 -x
HCOLL_ML_DISABLE_ALLTOALL=1 -x HCOLL_ML_DISABLE_ALLTOALLV=1 -mca
coll_tuned_use_dynamic_rules 1 -mca coll_tuned_alltoallv_algorithm 1 -mca
coll_tuned_alltoall_algorithm 1 ./bin-openmpi/pw.x -inp ausurf.in -nk 2 -nt 4 -nd 512
```

Running using Intel MPI:

```
$ module load intel/compiler/2016.4.258
$ module load intel/impi/5.1.3.258
mpirun -np 1024 -genv I_MPI_PIN on -genv DAT_OVERRIDE /etc/dat.conf -genv
I_MPI_DAT_LIBRARY /usr/lib64/libdat2.so -IB -genv MV2_USE_APM 0 -genv I_MPI_FABRICS
shm:ofa -genv I_MPI_OFA_USE_XRC 1 -genv I_MPI_OFA_NUM_ADAPTERS 1 -genv
I_MPI_OFA_ADAPTER_NAME mlx5_0 -genv I_MPI_OFA_NUM_PORTS 1 -genv MALLOC_MMAP_MAX_ 0 -
genv MALLOC_TRIM_THRESHOLD_ -1 ./bin-intelmpi/pw.x -inp ausurf.in -nk 2 -nt 4 -nd 512
```

7. espresso-drv.patch

This patch is needed to make the ELPA library work with Quantum ESPRESSO.

```
diff -Naur Modules/dspev_drv.f90 Modules-new/dspev_drv.f90
--- Modules/dspev_drv.f90       2016-01-09 00:24:18.000000000 -0800
+++ Modules-new/dspev_drv.f90   2016-11-07 15:43:29.533943796 -0800
@@ -679,6 +679,8 @@
       INTEGER     :: i
  #if defined(__ELPA)
       INTEGER     :: nprow,npcol,my_prow, my_pcol,mpi_comm_rows, mpi_comm_cols
+      INTEGER     :: mpierr
+      LOGICAL     :: success
  #endif

       IF( SIZE( s, 1 ) /= lds ) &
@@ -702,8 +704,10 @@

  #if defined(__ELPA)
```

```
      CALL BLACS_Gridinfo(ortho_cntx,nprow, npcol, my_prow,my_pcol)
-     CALL get_elpa_row_col_comms(ortho_comm, my_prow, my_pcol,mpi_comm_rows,
mpi_comm_cols)
-     CALL solve_evp_real(n,  n,   s, lds,    w,  vv, lds     ,nb  ,mpi_comm_rows,
mpi_comm_cols)
+     mpierr = get_elpa_row_col_comms(ortho_comm, my_prow, my_pcol,mpi_comm_rows,
mpi_comm_cols)
+     success = solve_evp_real(n,  n,   s, lds,    w,  vv, lds     ,nb,
nb,mpi_comm_rows, mpi_comm_cols)
+!     CALL get_elpa_row_col_comms(ortho_comm, my_prow, my_pcol,mpi_comm_rows,
mpi_comm_cols)
+!     CALL solve_evp_real(n,  n,   s, lds,    w,  vv, lds     ,nb  ,mpi_comm_rows,
mpi_comm_cols)

      IF( tv )  s = vv
      IF( ALLOCATED( vv ) ) DEALLOCATE( vv )
diff -Naur Modules/zhpev_drv.f90 Modules-new/zhpev_drv.f90
--- Modules/zhpev_drv.f90       2016-01-09 00:24:18.000000000 -0800
+++ Modules-new/zhpev_drv.f90   2016-11-07 15:43:49.782943792 -0800
@@ -1491,6 +1491,8 @@
      CHARACTER   :: jobv
 #if defined(__ELPA)
      INTEGER     :: nprow,npcol,my_prow, my_pcol,mpi_comm_rows, mpi_comm_cols
+     INTEGER     :: mpierr
+     LOGICAL     :: success
 #endif

      !
@@ -1505,9 +1507,12 @@

 #if defined(__ELPA)
      CALL BLACS_Gridinfo( ortho_cntx, nprow, npcol, my_prow, my_pcol )
-     CALL get_elpa_row_col_comms(ortho_comm, my_prow,
my_pcol,mpi_comm_rows,mpi_comm_cols)
-     CALL solve_evp_complex(n, n, h, size(h,1), w, v, size(h,1), nb, &
+     mpierr = get_elpa_row_col_comms(ortho_comm, my_prow,
my_pcol,mpi_comm_rows,mpi_comm_cols)
+     success = solve_evp_complex(n, n, h, size(h,1), w, v, size(h,1), nb, nb, &
                            mpi_comm_rows, mpi_comm_cols)
+!     CALL get_elpa_row_col_comms(ortho_comm, my_prow,
my_pcol,mpi_comm_rows,mpi_comm_cols)
+!     CALL solve_evp_complex(n, n, h, size(h,1), w, v, size(h,1), nb, &
+!                            mpi_comm_rows, mpi_comm_cols)
      h = v

      CALL mp_comm_free( mpi_comm_rows )
```