WRF 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:

The Weather Research and Forecasting (WRF) Model is a next-generation mesoscale numerical weather prediction system designed to serve both operational forecasting and atmospheric research needs. It features multiple dynamical cores, a 3-dimensional variational (3DVAR) data assimilation system, and a software architecture allowing for computational parallelism and system extensibility. WRF is suitable for a broad spectrum of applications across scales ranging from meters to thousands of kilometers.

3. Version Information:

Download WRF 3.8 at:

http://www2.mmm.ucar.edu/wrf/src/WRFV3.8.TAR.gz

Download WRF benchmarks at:

http://box.mmm.ucar.edu/wrf/WG2/benchv2

4. Prerequisites:

4.1 Hardware:

The instructions from this best practice have been tested on the HPC Advisory Council, Dell™ PowerEdge™ R730 32-node cluster

- Dual Socket Intel® Xeon® 14-core CPUs E5-2697 V3 @ 2.60 GHz
- Mellanox ConnectX-4 EDR 100Gb/s InfiniBand adapters
- Mellanox Switch-IB SB7700 36-Port 100Gb/s EDR InfiniBand switches

4.2 Software:

a. OS: Red Hat Enterprise Linux 6.5+

b. Compilers: Intel compilers 2016

c. MPI: hpcx-v1.5.370-icc

d. Other:

- hdf5-1.8.16
- netcdf-4.4.0
- netcdf-fortran-4.4.3
- parallel-netcdf-1.7.0
- benchmark workload

5. Installation

5.0 Building OpenMPI using Intel Compiler 2016

```
---
source /opt/intel/compilers_and_libraries_2016.1.150/
linux/bin/compilervars.sh intel64


export CC=icc
export CXX=icpc
export FC=ifort
export F90=ifort


cd openmpi-1.10.2
make clean


./configure --prefix=${HPCX_HOME}/ompi-v1.10-i16
--with-knem=${HPCX_HOME}/knem \
        --with-fca=${HPCX_HOME}/fca --with-
mxm=${HPCX_HOME}/mxm \
        --with-hcoll=${HPCX_HOME}/hcoll \
        --with-platform=contrib/platform/mellanox/opti-
mized


make -j 16 all
make -j 16 install
---
# Create a module file for openmpi
% cd /opt/hpcx-v1.5.370-icc-MLNX_OFED_LINUX-3.2-
2.0.0.0-redhat6.5-x86_64/modulefiles
% cp hpcx-ompi-v1.10 hpcx-ompi-v1.10-i16
% sed "s/ompi-v1.10/ompi-v1.10-i16/g" –i hpcx-ompi-
v1.10-i16
```

5.1 Building hdf5

Download hdf5 from http://www.hdfgroup.org/ftp/HDF5/current/src/hdf5-1.8.16.tar.gz.

```
---
source /opt/intel/compilers_and_libraries_2016.1.150/linux/bin/compilervars.sh intel64
module use /opt/hpcx-v1.5.370-icc-MLNX_OFED_LINUX-3.2-2.0.0.0-redhat6.5-x86_64/modulefiles
module load hpcx-ompi-v1.10-i16

export CC=mpicc
export CXX=mpic++
export FC=mpif90
export F90=mpif90

./configure --prefix= /application/tools/i16/hdf5-1.8.16/install-hpcx --enable-parallel --enable-shared
make -j 28 install
```

5.2 Building parallel-netcdf

Download parallel-netcdf from http://cucis.ece.northwestern.edu/projects/PnetCDF/download.html.

```
---
source /opt/intel/compilers_and_libraries_2016.1.150/linux/bin/compilervars.sh intel64
module use /opt/hpcx-v1.5.370-icc-MLNX_OFED_LINUX-3.2-2.0.0.0-redhat6.5-x86_64/modulefiles
module load hpcx-ompi-v1.10-i16

export CC=mpicc
export CXX=mpicxx
export FC=mpif90
export F77=mpif90
export F90=mpif90

export OMPI_MPICC=icc
export OMPI_MPICXX=icpc
export OMPI_MPIFC=ifort
export OMPI_MPIF77=ifort
export OMPI_MPIF90=ifort

export CFLAGS=’-g -O2 -fPIC’
export CXXFLAGS=’-g -O2 -fPIC’
export FFLAGS=’-g -fPIC’
export FCFLAGS=’-g -fPIC’
export FLDFLAGS=’-fPIC’
export F90LDFLAGS=’-fPIC’
export LDFLAGS=’-fPIC’

./configure --prefix= /application/tools/i16/ parallel-netcdf-1.7.0/install-hpcx --enable-fortran --enable-large-file-test
make -j 28 install
```

5.3 Building netcdf-C and netcdf-Fortran

Download netcdf-C and netcdf-Fortran from http://www.unidata.ucar.edu/downloads/netcdf/index.jsp.

```
--- netcdf-C
source /opt/intel/compilers_and_libraries_2016.1.150/linux/bin/compilervars.sh intel64
module use /opt/hpcx-v1.5.370-icc-MLNX_OFED_LINUX-3.2-2.0.0.0-redhat6.5-x86_64/modulefiles
module load hpcx-ompi-v1.10-i16

export CC=mpicc
export CXX=mpicxx
export FC=mpif90
export F77=mpif90
export F90=mpif90

export OMPI_MPICC=icc
export OMPI_MPICXX=icpc
export OMPI_MPIFC=ifort
export OMPI_MPIF90=ifort

HDF5=/application/tools/i16/hdf5-1.8.16/install-hpcx
PNET=/application/tools/i16/parallel-netcdf-1.7.0/install-hpcx

export CPPFLAGS=”-I$HDF5/include -I$PNET/include”
export CFLAGS=”-I$HDF5/include -I$PNET/include”
export CXXFLAGS=”-I$HDF5/include -I$PNET/include”
export FCFLAGS=”-I$HDF5/include -I$PNET/include”
export FFLAGS=”-I$HDF5/include -I$PNET/include”
export LDFLAGS=”-I$HDF5/include -I$PNET/include -L$PNET/lib “

export WRFIO_NCD_LARGE_FILE_SUPPORT=1
```

```
./configure --prefix= /application/tools/i16/netcdf-4.4.0/install-hpcx --enable-fortran --disable-static --enable-shared --with-pic --enable-parallel-tests -enable-pnetcdf --enable-large-file-tests --enable-largefile
make
make install


--- netcdf-Fortran
module purge


source /opt/intel/compilers_and_libraries_2016.1.150/linux/bin/compilervars.sh intel64
module use /opt/hpcx-v1.5.370-icc-MLNX_OFED_LINUX-3.2-2.0.0.0-redhat6.5-x86_64/modulefiles
module load hpcx-ompi-v1.10-i16


export CC=mpicc
export CXX=mpicxx
export FC=mpif90
export F77=mpif90
export F90=mpif90


export OMPI_MPICC=icc
export OMPI_MPICXX=icpc
export OMPI_MPIFC=ifort
export OMPI_MPIF90=ifort


export WRFIO_NCD_LARGE_FILE_SUPPORT=1


HDF5=/application/tools/i16/hdf5-1.8.16/install-hpcx
NCDIR=/application/tools/i16/netcdf-4.4.0/install-hpcx
export LD_LIBRARY_PATH=${NCDIR}/lib:${LD_LIBRARY_PATH}


export CPPFLAGS=”-I$HDF5/include -I$NCDIR/include”
export CFLAGS=”-I$HDF5/include -I$NCDIR/include”
export CXXFLAGS=”-I$HDF5/include -I$NCDIR/include”
export FCFLAGS=”-I$HDF5/include -I$NCDIR/include”
export FFLAGS=”-I$HDF5/include -I$NCDIR/include”
export LDFLAGS=”-I$HDF5/include -I$NCDIR/include -L$NCDIR/lib “


./configure --prefix=$NCDIR --disable-static --enable-shared --with-pic --enable-parallel-tests --enable-large-file-tests --enable-largefile


make
make install
```

5.4 Building WRF-3.8

```
source /opt/intel/compilers_and_libraries_2016.1.150/linux/bin/compilervars.sh intel64
module use /opt/hpcx-v1.5.370-icc-MLNX_OFED_LINUX-3.2-2.0.0.0-redhat6.5-x86_64/modulefiles
module load hpcx-ompi-v1.10-i16


export PHDF5=/application/tools/i16/hdf5-1.8.16/install-hpcx
export NETCDF=/application/tools/i16/netcdf-4.4.0/install-hpcx
export PNETCDF=/application/tools/i16/parallel-netcdf-1.7.0/install-hpcx
export WRFIO_NCD_LARGE_FILE_SUPPORT=1


cat <<EOF > answer
67              # 67. (dm+sm)   INTEL (ifort/icc): HSW/BDW


EOF


./clean -a
./configure < answer
rm -f answer
./compile -j 32 wrf
```

6. Running WRF with HPCX

```
source /opt/intel/compilers_and_libraries_2016.1.150/linux/bin/compilervars.sh intel64
module use /opt/hpcx-v1.5.370-icc-MLNX_OFED_LINUX-3.2-2.0.0.0-redhat6.5-x86_64/modulefiles
module load hpcx-ompi-v1.10-i16

export WRFIO_NCD_LARGE_FILE_SUPPORT=1
export LD_LIBRARY_PATH=/application/tools/i16/netcdf-4.4.0/install-hpcx/lib:$LD_LIBRARY_PATH
USE_HCOLL=1
USE_MXM=1

FLAGS=””
HCA=mlx5_0
FLAGS+=”-mca btl openib,sm,self “
FLAGS+=”-mca btl_openib_if_include $HCA:1 “
FLAGS+=”-x MXM_RDMA_PORTS=$HCA:1 “
FLAGS+=”-mca rmaps_base_dist_hca $HCA:1 “
FLAGS+=”-x HCOLL_MAIN_IB=$HCA:1 “
FLAGS+=”-x HCOLL_IB_IF_INCLUDE=$HCA:1 “
FLAGS+=”-mca coll_fca_enable 0 “

if [[ “$USEKNEM” == “1” ]]; then
    FLAGS+=”-mca btl_sm_use_knem 1 “
    FLAGS+=”-x MXM_SHM_KCOPY_MODE=knem “
else
    FLAGS+=”-mca btl_sm_use_knem 0 “
fi

if [[ “$USE_HCOLL” == “1” ]]; then
    FLAGS+=”-mca coll_hcoll_enable 1 “
    FLAGS+=”-mca coll_hcoll_np 0 “
else
    FLAGS+=”-mca coll_hcoll_enable 0 “
fi

if [[ “$USE_MXM” == “1” ]]; then
    FLAGS+=”-mca pml yalla “
    FLAGS+=”-mca mtl_mxm_np 0 “
    FLAGS+=”-x MXM_TLS=$TPORT,shm,self “
    FLAGS+=”-x HCOLL_ENABLE_MCAST_ALL=1 “
else
    FLAGS+=”-mca mtl ^mxm “
    FLAGS+=”-mca pml ob1  “
fi

FLAGS+=”-hostfile <machinefile> “
FLAGS+=”-report-bindings “
FLAGS+=”--bind-to core “
FLAGS+=”-map-by node “

mpirun -np 1024 $FLAGS wrf.exe
```

6.1 Running WRF using Parallel netcdf

In the namelist.input, the following settings support pNetCDF by setting value to 11:

io_form_boundary

io_form_history

io_form_auxinput2

io_form_auxhist2

Set nocolons = .true. in the section &time_control of namelist.input.