WRF 3.2.1 Best Practices for Intel® Cluster Ready

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 3.2.1 at:

http://www.mmm.ucar.edu/wrf/users/download/get_source.html

Download WRF 3.2.1 benchmarks at:

http://www.mmm.ucar.edu/WG2bench/conus12km_data_v3

Single domain, medium size. 12km CONUS, Oct. 2001

Download NetCDF 4.1.1 at:

http://www.unidata.ucar.edu/downloads/netcdf/netcdf-4_1_1/index.jsp

4. Prerequisites:

4.1 Hardware:

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

- Dell PowerEdge M610 14-node cluster
- Intel Xeon X5670 CPUs @ 2.93 MHz
- Memory: 24GB per node @ 1333MHz
- Mellanox ConnectX-2 QDR InfiniBand Adapters
- Mellanox QDR InfiniBand Switch

4.2 Software:

- Intel® Cluster Ready running CentOS 5 Update 4
- Application: WRF 3.2.1
- Compilers: Intel compilers, GNU compilers
- MPI: Intel MPI 4 U1, Open MPI 1.5, Platform MPI 8.0.1
- Miscellaneous: NetCDF 4.1.1
- Benchmark workload: conus12km

5. Building WRF

Extract WRF

```
% tar xvfz WRFV3.2.1.TAR.gz
% cd ~/WRFV3
```

Building WRF

WRF would require NetCDF to be installed first. You should set the NETCDF environment variable before running the "configure" script. When you run the configure script, it will show you a list of supported platform. The "dmpar" option is for strictly running with MPI and the "dm+sm" is the mode of running with the OMP and MPI Hybrid. The configure script will also generate a file called "configure.wrf" for you to customize the paths and compiler flags for your environment.

```
% export NETCDF=/application/netcdf-4.1.1
% ./configure
% vim configure.wrf
% ./compiler wrf
```

Editing configure.wrf for Platform MPI, Open MPI

You will need to change the DM_FC and DM_CC lines for MPI implementations other than Intel MPI:

```
#DM_FC          =       mpif90 -f90=$(SFC)
DM_FC           =       mpif90
#DM_CC          =       mpicc -cc=$(SCC) -DMPI2_SUPPORT
DM_CC           =       mpicc -DMPI2_SUPPORT
```

© Copyright 2010. HPC Advisory Council. All rights reserved.

Editing configure.wrf for GNU Compilers

There appears to be bugs in the GNU compilers 4.1 that are included with RHEL/CentOS. WRF will encounter segmentation fault at runtime. It is best to either use the GCC 4.4.0 in RHEL or CentOS 5 or build your own GCC toolchain. To use GNU 4.4.0 from RHEL/CentOS, simply yum install or RPM install the following GCC packages:

gcc44-4.4.0-6.el5

gcc44-c++-4.4.0-6.el5

gcc44-gfortran-4.4.0-6.el5

libgfortran44-4.4.0-6.el5

libgfortran44-4.4.0-6.el5

libstdc++44-devel-4.4.0-6.el5

Below is a snapshot of the crucial lines in configure.wrf.

Note that "--ffast-math" flag is added to FCOPTIM, and "--lcurl" is added because NetCDF was built with curl support.

```
DMPARALLEL      =       1
OMPCPP          =       # -D_OPENMP
OMP             =       # -fopenmp
OMPCC           =       # -fopenmp
SFC             =       gfortran44
SCC             =       gcc44
CCOMP           =       gcc44
DM_FC           =       mpif90
DM_CC           =       mpicc -DMPI2_SUPPORT
FC              =       $(DM_FC)
CC              =       $(DM_CC) -DFSEEKO64_OK
LD              =       $(FC)
LDFLAGS_LOCAL   =       -lcurl
FCOPTIM         =       -O3 -ffast-math -ftree-vectorize
-ftree-loop-linear -funroll-loops
```

During compile time, you may run into this error.

module_soil_pre.f90:539: internal compiler error: in gfc_typenode_for_spec, at fortran/trans-types.c:652

Modify share/module_soil_pre.F to comment out these 6 lines that triggers the fortran error to work around the issue:

```
531     !WRITE ( num_cat_count , FMT = '(I3)' ) num_
        soil_top_cat
532     !WRITE ( message , FMT = '('//num_cat_
        count//'(i3,1x))'      ) (ints(l),l=1,num_soil_top_cat)
533     !CALL wrf_debug(1,message)
534     !WRITE ( message , FMT = '('//num_cat_
        count//'(i3,1x))'      ) &
535     !   nint(soil_top_cat(i,ints(1:num_soil_top_
        cat),j)*100)
536 !   !   nint(soil_top_cat(i,(ints(k),k=1:num_soil_top_
        cat),j    )*100)
```

Building WRF with Intel Compilers and Intel MPI

```
DMPARALLEL      =       1
OMPCPP          =       # -D_OPENMP
OMP             =       # -openmp -fpp –auto
OMPCC           =       # -openmp -fpp –auto
SFC             =       ifort
SCC             =       icc
CCOMP           =       icc
DM_FC           =       mpif90 -f90=$(SFC)
DM_CC           =       mpicc -cc=$(SCC) -DMPI2_
SUPPORT
FC              =       $(DM_FC)
CC              =       $(DM_CC)
LD              =       $(FC)
CFLAGS_LOCAL    =       -w -O3 –ip
LDFLAGS_LOCAL   =       -ip –lcurl
FCOPTIM         =       -O3
FCREDUCEDOPT    =       $(FCOPTIM)
FCNOOPT         =       -O0 -fno-inline -fno-ip
FCDEBUG         =       # -g $(FCNOOPT) –traceback
BYTESWAPIO      =       -convert big_endian
FCBASEOPTS_NO_G =       -w -ftz -align all
-fno-alias -fp-model precise $(FORMAT_FREE)
$(BYTESWAPIO)
```

© Copyright 2010. HPC Advisory Council. All rights reserved.

Additionally, the –xS SSE compiler optimization may be used to get better performance depends on the machine architecture.

Building WRF with Intel Compilers and Intel MPI with OpenMP Hybrid

Here is the example of the configure.wrf for compiling WRF with the (dm+sm) support.

```
DMPARALLEL      =       1
OMPCPP          =       -D_OPENMP
OMP             =       -openmp -fpp –auto
OMPCC           =       -openmp -fpp –auto
SFC             =       ifort
SCC             =       icc
CCOMP           =       icc
DM_FC           =       mpif90 -f90=$(SFC)
DM_CC           =       mpicc -cc=$(SCC) -DMPI2_
SUPPORT -DMPI2_THREAD_SUPPORT
FC              =       $(DM_FC)
CC              =       $(DM_CC) -DFSEEKO64_OK
LD              =       $(FC)
CFLAGS_LOCAL    =       -w -O3 -ip -xHost -fp-model
fast=2 -no-prec-div -no-prec-sqrt –openmp
LDFLAGS_LOCAL   =       -ip –lcurl
FCOPTIM         =       -O3 -xHost -ip -fp-model fast=2
-no-prec-div -no-prec-sqrt –openmp
BYTESWAPIO      =       -convert big_endian
FCBASEOPTS_NO_G =       -w -ftz -align all -fno-alias
$(FORMAT_FREE) $(BYTESWAPIO)
```

6. Running WRF

After you finished compiling WRF, that you verify the wrf.exe executable is generated in run/wrf.exe, there are a few steps you will need before running the application.

Setting up the dataset

First you will need to extract the conus 12km dataset to a directory.

```
% cd ~/WRFV3.2.1/test/em_real
% tar xvfz ~/conus12km_data_v3-2.tar.gz
```

Copy the dataset to WRFV3.2.1/test/em_real directory where you will be running WRF at.

```
% cp ~/WRFV3.2.1/run/RRTM_DATA ~/WRFV3.2.1/
test/em_real
```

These are the examples of running WRF. Make sure you are in the "em_real" directory when you run WRF.exe.

Running with Intel MPI

```
% mpdboot -r ssh -f ~/mpd.hosts.ib.14 -n 14
% mpiexec -np 168 -IB ~/WRFV3.2.1/run/wrf.exe
%mpdallexit
```

When running the OpenMP + MPI Hybrid version, make sure that you define the appropriate environment variables like the following: KMP_AFFINITY, OMP_NUM_THREADS, I_MPI_PIN_DOMAIN

Running with Open MPI

```
%mpirun -np 168 -mca btl self,sm,openib –hostfile /
home/demo/hostfile-ompi.14 -mca mpi_paffinity_
alone 1 ~/WRFV3.2.1/run/wrf.exe
```

Running with Platform MPI

```
% mpirun -np 168 -IBV -cpu_bind -prot -hostfile ~/
hostfile-ib14 ~/WRFV3.2.1/run/wrf.exe
```

HPC ADVISORY COUNCIL
NETWORK OF EXPERTISE
350 Oakmead Pkwy, Sunnyvale, CA 94085
Tel: 408-970-3400 • Fax: 408-970-3403
www.hpcadvisorycouncil.com

© Copyright 2010. HPC Advisory Council. All rights reserved.