CESM with 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:

Community Earth System Model (CESM) is a coupled climate model for simulating the earth's climate system. It is composed of four separate models simultaneously simulating: Earth's atmosphere, ocean, land surface and sea-ice. CESM allows researchers to conduct fundamental research into the earth's past, present and future climate states. Note that CESM1.0.3 supersedes CCSM4.0.

For more information about CESM:

http://www.cesm.ucar.edu/models/cesm1.0/

3. Version Information:

Software version being used: CESM 1.0.3

Registration is required for obtaining the source code. Refer to the above URL for the registration.

To check out the source code with subversion:

$ svn co https://svn-ccsm-release.cgd.ucar.edu/model_versions/cesm1_0_3 cesm1_0_3

4. Prerequisites:

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

4.1 Hardware:

- Dell PowerEdge M610 38-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 RHEL 5.5
- Mellanox OFED 1.5.2 InfiniBand Software Stack
- Mellanox Fabric Collectives Accelerator™ 2.1
- Compilers: Intel compilers 11.1
- MPI: Open MPI 1.5.3 with KNEM 0.9.6, Platform MPI 8.0.1
- Application: CESM 1.0.3
- Libraries: NetCDF 4.1.3
- Benchmark workload: B1850CN

5. Building CESM

After the source code has been checked out, you would need to create the case.:

a. Create case

First, use this command to define the case:

```
$ cd ~/cesm_1.0.3/scripts
$ ./create_newcase -case /home/user/cesm1_0_3/B1850CN \
 -mach generic_linux_intel \
-compset B_1850_CN \
-res 0.9x1.25_gx1v6 \
-din_loc_root_csmdata  /home/user/cesm1_0_3/dataset-B1850CN \
-scratchroot /home/user/cesm1_0_3/B1850CN/scratchroot \
-max_tasks_per_node 12
```

6. Modifying the configuration files

a. Modify env_mach_pes.xml

```
$ cd ~/cesm_1.0.3/B1850CN
```

Modify env_mach_pes.xml and change all reference of 128 to the number of processes you intend to run. In my case, I will run with a job with 384 MPI processes, I replace all references of 128 to 384.

Every time you want to run a job with a different number of processes, you will need to change this and recompile at the end.

b. Modify Macros.generic_linux_intel

You will need to modify this file for the library paths:

```
$ vim Macros.generic_linux_intel
NETCDF_PATH   := /application/netcdf-4.1.3/install-intel/
INC_NETCDF    := $(NETCDF_PATH)/include
LIB_NETCDF    := $(NETCDF_PATH)/lib
MOD_NETCDF    := $(NETCDF_PATH)/include
MPICH_PATH    := /usr/mpi/intel/openmpi-1.3.3
```

```
INC_MPI      := $(MPICH_PATH)/include
LIB_MPI      := $(MPICH_PATH)/lib
MPI_LIB_NAME  := mpi
PNETCDF_PATH  :=
INC_PNETCDF   :=
LIB_PNETCDF   :=
LAPACK_LIBDIR :=
```

For Platform MPI, you will need this:

```
MPICH_PATH := /opt/platform_mpi
LIB_MPI    := $(MPICH_PATH)/lib/linux_amd64
LIB_MPI    := $(MPICH_PATH)/MPICH2.0/lib/linux_amd64
INC_MPI    := $(MPICH_PATH)/include
INC_MPI    := $(MPICH_PATH)/MPICH2.0/include
MPI_LIB_NAME  := mpi
MPI_LIB_NAME  := mpich
```

c. Modify Tools/Makefile

You may need this change if your netcdf creates a fortran library (libnetcdff.so)

```
98 # System libraries (netcdf, mpi, pnetcdf, esmf)
 99 ifeq ($(strip $(SLIBS)),)
100    SLIBS := -L$(LIB_NETCDF) -lnetcdf –lnetcdff
101 else
102    SLIBS += -L$(LIB_NETCDF) -lnetcdf –lnetcdff
103 endif
```

d. Modify env_mach_specific

At the bottom of this file, you will need to define the path to the MPI directory and NETCDF directory.

```
$ vim env_mach_specific:
setenv MPICH_PATH /opt/platform_mpi
setenv NETCDF_PATH /application/netcdf-4.1.3/install-intel
```

7. Configure the case

After the configurations are set, you need to create the build/run/clean/submit scripts. As mentioned before, changing the number of processes to run requires rebuilding of the case.

```
$ cd ~/cesm_1.0.3/B1850CN
$ ./configure -case
```

8. Building the case

After the build script is generated, run this to compile.

```
$ ./B1850CN.generic_linux_intel.build
```

Sometimes, building would fail because it needs certain datasets. If the system has internet access, the dataset would be automatically. In cases when it is not, you will need to download them manually:

```
$ mkdir -p ~/cesm_1.0.3/dataset-1850CN/ccsm4_init/b40.1850.track1.1deg.006/0863-01-01
$ cd ~/cesm_1.0.3/dataset-1850CN/ccsm4_init/b40.1850.track1.1deg.006
$ svn export --force https://svn-ccsm-inputdata.cgd.ucar.edu/trunk/inputdata/ccsm4_init/b40.1850.track1.1deg.006/0863-01-01
```

The executable is under: ~ /cesm1_0_3_nfs/B1850CN/scratchroot/B1850CN /run/ccsm.exe

9. Running the case

Before running the script to start the job, you will need to define the mpirun command line. Near the end of the run script, comment out the lines for instructions and exit, then define your mpirun command:

```
$ vim B1850CN.generic_linux_intel.run
#echo "GENERIC_USER: Put the correct mpirun command in your *.run script, then remove this echo/exit"
#exit 2
# Command for Platform MPI:
mpirun -np 384 -IBV -hostfile ~/hostfile -prot -i profile -e MPI_IB_MULTIRAIL=1 -srq -IBV -aff=automatic /home/user/cesm1_0_3 /B1850CN/scratchroot/B1850CN/run/ccsm.exe >&! outputfile
# Command for Open MPI:
mpirun -np 384 -mca btl self,sm,openib  -mca btl_openib_if_include mlx4_0 -mca btl_sm_use_knem 1 -hostfile ~/hostfile --bind-to-core -mca coll_fca_enable 1 ./ccsm.exe  >&! outputfile
```

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