WRF and CESM on Janus

Davide Del Vento - 2012 June 21
From CESM release 1.0.4 Janus is a supported machine.

At present, PGI compiler is the only supported in CESM.

Use the standard CESM scripts to create, build, and run cases.
CESM example

For a 2-degree FV core, B_1850 compset

```bash
export CASE=testcase
export CASEROOT=/lustre/janus_scratch/$USER/$CASE
export COMPSET=B_1850
export RES=1.9x2.5_gx1v6
export MACH=janus
./create_newcase
    -case $CASEROOT
    -mach $MACH
    -compset $COMPSET
    -res $RES
cd /lustre/janus_scratch/$USER/testcase/
configure -case
./testcase.janus.build
qsub testcase.janus.run
```
 CESM performance

CCSM Tag : cesm1_0_4 (best guess)
grid : 1.9x2.5_gx1v6
compset : B_1850 (B1850)
stop_option : ndays, stop_n = 5
run_length : 5 days (4 for ocean)

Contribution of individual models to performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Run Time</th>
<th>m/day</th>
<th>simulated_years/day</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOT</td>
<td>172.8s</td>
<td>34.5s</td>
<td>6.85</td>
</tr>
<tr>
<td>LND</td>
<td>10.7s</td>
<td>2.1s</td>
<td>110.07</td>
</tr>
<tr>
<td>ICE</td>
<td>31.2s</td>
<td>6.2s</td>
<td>37.92</td>
</tr>
<tr>
<td>ATM</td>
<td>49.2s</td>
<td>9.8s</td>
<td>24.01</td>
</tr>
<tr>
<td>OCN</td>
<td>74.8s</td>
<td>14.9s</td>
<td>15.81</td>
</tr>
<tr>
<td>GLC</td>
<td>0.0s</td>
<td>0.0s</td>
<td>0.00</td>
</tr>
<tr>
<td>CPL</td>
<td>11.9s</td>
<td>2.3s</td>
<td>99.01</td>
</tr>
</tbody>
</table>

Model Throughput: 6.85 simulated_years/day
WRF

- Works with both PGI and Intel compilers
- But there are some tricks you have to know and use
- A NetCDF rebuild at the end of May, broke the old WRF executable you may have done
Building WRF with Intel Compiler

- This used to work out-of-the-box until mid-May
- An update in HDF5, used by NetCDF broke it
- You may fix it manually, to use an older dotkit
- We will maintain a NCAR-WRF-Parallel-Intel dotkit
Building WRF with Intel Compiler

use NCAR-WRF-Parallel-Intel

export WRFIO_NCD_LARGE_FILE_SUPPORT=1

./configure

➲ Then select 11, namely
  11. Linux x86_64 i486 i586 i686, ifort compiler with icc (dmpar)

➲ Then select nesting (currently tested only without nesting)

./compile em_real
Building WRF with PGI Compiler

- WRF does not build with PGI C compiler, on any platform

- In fact, it uses PGI Fortran compiler plus gcc

- The naming conventions and paths used on Janus require workarounds

- The same NetCDF/HDF5 issue mentioned for Intel compiler applies

- We will maintain a NCAR-WRF-Parallel-PGI dotkit, but you will responsible for the workarounds
Building WRF with PGI Compiler

use NCAR-WRF-Parallel-PGI
export WRFIO_NCD_LARGE_FILE_SUPPORT=1
./configure

- Then select 7, namely
  7. Linux x86_64, PGI accelerator compiler with gcc (dmpar)
- Select nesting (currently tested only without nesting)
- Edit the configure.wrf file as follows:
  DM_CC=/curc/tools/free/redhat_5_x86_64/openmpi-1.4.3_gcc-4.5.2_torque-2.5.8_ib/bin/mpicc
  -DMPI2_SUPPORT

./compile em_real
WRF scaling on CONUS 12km
WRF scaling on CONUS 2.5km