Instructions to Build and Run Earthworks compsets on Perlmutter

Code Changes

1. Code goes in your *.bashrc* file

```
export LMOD_REDIRECT=yes                # Send command output to STDOUT so
it can pipe more easily.
export LMOD_IGNORE_CACHE=1              # Try this for now, given that
we're constantly updating.
export LMOD_TMOD_FIND_FIRST=1           # Ignore assigned precendence and
use path-ordering instead.
                                        # This is essential for PrgEnv's to
adjust precedence.

export   NVIDIA=$CFS/nvendor/nvidia                    # Project space.
export SHAREDIR=$NVIDIA/SHARE.perlmutter               # Used for
installs.

module use --prepend $SHAREDIR/Modules/Deprecated      # Assign these in
reverse-order, to give
module use --prepend $SHAREDIR/Modules/Legacy          # precedence to the
most current.
module use --prepend $SHAREDIR/Modules/Latest

module use --append $SHAREDIR/Modules/Bundles          # New framework.
module use --append $SHAREDIR/Modules/PrgEnv/*/*
```

2. Code changes to ccs_config

- Add Perlmutter to your ccs_config/machines/config_machines.xml

```
  <machine MACH="perlmutter">
    <DESC>NERSC EX AMD EPYC, os is CNL, 64 pes/node, batch system is
Slurm</DESC>
    <NODENAME_REGEX>$ENV{NERSC_HOST}:perlmutter</NODENAME_REGEX>
    <OS>CNL</OS>
    <COMPILERS>nvhpc</COMPILERS>
    <MPILIBS>openmpi</MPILIBS>
    <PROJECT>m4180</PROJECT>
    <CIME_OUTPUT_ROOT>$ENV{SCRATCH}</CIME_OUTPUT_ROOT>
    <DIN_LOC_ROOT>/global/cfs/cdirs/ccsm1/inputdata</DIN_LOC_ROOT>

<DIN_LOC_ROOT_CLMFORC>/global/cfs/cdirs/ccsm1/inputdata/atm/datm7</DIN_LOC_
ROOT_CLMFORC>
    <DOUT_S_ROOT>$CIME_OUTPUT_ROOT/archive/$CASE</DOUT_S_ROOT>
```

```
    <BASELINE_ROOT>/global/cfs/cdirs/ccsm1/ccsm_baselines</BASELINE_ROOT>

<CCSM_CPRNC>/global/cfs/cdirs/ccsm1/tools/cprnc.perlmutter/cprnc</CCSM_CPRN
C>
    <GMAKE_J>8</GMAKE_J>
    <BATCH_SYSTEM>slurm</BATCH_SYSTEM>
    <SUPPORTED_BY>cseg</SUPPORTED_BY>
    <MAX_TASKS_PER_NODE>128</MAX_TASKS_PER_NODE>
    <MAX_GPUS_PER_NODE>4</MAX_GPUS_PER_NODE>
    <MAX_MPITASKS_PER_NODE>64</MAX_MPITASKS_PER_NODE>
    <PROJECT_REQUIRED>TRUE</PROJECT_REQUIRED>
    <mpirun mpilib="default">
      <executable>srun</executable>
      <arguments>
        <arg name="label"> --label</arg>
        <arg name="num_tasks" > -n {{ total_tasks }}</arg>
        <arg name="binding"> -c {{ srun_binding }}</arg>
      </arguments>
    </mpirun>
  <module_system type="module">
      <init_path lang="perl">/usr/share/lmod/lmod/init/perl</init_path>
      <init_path
lang="python">/usr/share/lmod/lmod/init/env_modules_python.py</init_path>
      <init_path lang="sh">/usr/share/lmod/lmod/init/sh</init_path>
      <init_path lang="csh">/usr/share/lmod/lmod/init/csh</init_path>
      <cmd_path lang="perl">/usr/share/lmod/lmod/libexec/lmod
perl</cmd_path>
      <cmd_path lang="python">/usr/share/lmod/lmod/libexec/lmod
python</cmd_path>
      <cmd_path lang="sh">module</cmd_path>
      <cmd_path lang="csh">module</cmd_path>
      <modules>
        <command name="rm">PrgEnv-nvidia</command>
        <command name="rm">PrgEnv-cray</command>
        <command name="rm">PrgEnv-aocc</command>
        <command name="rm">PrgEnv-gnu</command>
        <command name="rm">nvidia</command>
        <command name="rm">cce</command>
        <command name="rm">gnu</command>
        <command name="rm">aocc</command>
        <command name="rm">cray-parallel-netcdf</command>
        <command name="rm">cray-hdf5-parallel</command>
        <command name="rm">cray-libsci</command>
        <command name="rm">cray-mpich</command>
        <command name="rm">cray-hdf5</command>
        <command name="rm">cray-netcdf-hdf5parallel</command>
        <command name="rm">cray-netcdf</command>
        <command name="rm">craype</command>
      </modules>

      <modules compiler="nvhpc">
        <command name="purge"/>
        <command name="load">PrgEnv/PGI+OpenMPI/2024-01-05</command>
        <command name="load">esmf</command>
```

```
      </modules>
      <modules>
        <command name="load">cmake/3.24.3</command>
      </modules>
    </module_system>
    <environment_variables>
      <env name="OMP_STACKSIZE">256M</env>
<!--      <env name="OMP_PROC_BIND">spread</env>
      <env name="OMP_PLACES">threads</env> -->
      <env
name="ESMFMKFILE">$ENV{ESMF}/lib/libO/Linux.nvhpc.64.openmpi.default/esmf.m
k</env>
      <env name="CC">nvc</env>
      <env name="FC">nvfortran</env>
      <env name="NETCDF_C_PATH">$ENV{NETCDF_C}</env>
      <env name="NETCDF_FORTRAN_PATH">$ENV{NETCDF_F}</env>
    </environment_variables>
  </machine>
```

- Create/Add the following to your ccs_config/machines/cmake_macros/nvhpc_perlmutter.cmake

```
string(APPEND CFLAGS " -gopt  -time")
if (compile_threaded)
  string(APPEND CFLAGS " -mp")
endif()
if (NOT DEBUG)
  string(APPEND CFLAGS " -O")
  string(APPEND FFLAGS " -O")
endif()
string(APPEND CFLAGS " -Mnofma")
string(APPEND FFLAGS " -Mnofma")

string(APPEND CPPDEFS " -DFORTRANUNDERSCORE -DNO_SHR_VMATH -DNO_R16  -
DCPRPGI")
set(CXX_LINKER "CXX")
set(FC_AUTO_R8 "-r8")
string(APPEND FFLAGS " -i4 -gopt  -time -Mextend -byteswapio -Mflushz -
Kieee")
if (compile_threaded)
  string(APPEND FFLAGS " -mp")
endif()
if (DEBUG)
  string(APPEND FFLAGS " -O0 -g -Ktrap=fp -Mbounds -Kieee")
endif()
if (COMP_NAME STREQUAL datm)
  string(APPEND FFLAGS " -Mnovect")
endif()
if (COMP_NAME STREQUAL dlnd)
  string(APPEND FFLAGS " -Mnovect")
endif()
if (COMP_NAME STREQUAL drof)
  string(APPEND FFLAGS " -Mnovect")
```

```
endif()
if (COMP_NAME STREQUAL dwav)
  string(APPEND FFLAGS " -Mnovect")
endif()
if (COMP_NAME STREQUAL dice)
  string(APPEND FFLAGS " -Mnovect")
endif()
if (COMP_NAME STREQUAL docn)
  string(APPEND FFLAGS " -Mnovect")
endif()
set(FFLAGS_NOOPT "-O0")
set(FIXEDFLAGS "-Mfixed")
set(FREEFLAGS "-Mfree")
set(HAS_F2008_CONTIGUOUS "FALSE")
set(LDFLAGS "-time -Wl,--allow-multiple-definition")
if (compile_threaded)
  string(APPEND LDFLAGS " -mp")
endif()
set(MPICC "mpicc")
set(MPICXX "mpicxx")
set(MPIFC "mpif90")
set(SCC "nvc")
set(SCXX "nvc++")
set(SFC "nvfortran")
if (GPU_TYPE STREQUAL v100 AND GPU_OFFLOAD STREQUAL openacc)
   string(APPEND GPUFLAGS  " -acc -gpu=cc70,lineinfo,nofma -Minfo=accel ")
endif()
if (GPU_TYPE STREQUAL v100 AND GPU_OFFLOAD STREQUAL openmp)
   string(APPEND GPUFLAGS  " -mp=gpu -gpu=cc70,lineinfo,nofma -Minfo=accel
")
endif()
if (GPU_TYPE STREQUAL v100 AND GPU_OFFLOAD STREQUAL combined)
   string(APPEND GPUFLAGS  " -acc -gpu=cc70,lineinfo,nofma -mp=gpu -
Minfo=accel ")
endif()
if (GPU_TYPE STREQUAL a100 AND GPU_OFFLOAD STREQUAL openacc)
   string(APPEND GPUFLAGS  " -acc -gpu=cc80,lineinfo,nofma -Minfo=accel ")
endif()
if (GPU_TYPE STREQUAL a100 AND GPU_OFFLOAD STREQUAL openmp)
   string(APPEND GPUFLAGS  " -mp=gpu -gpu=cc80,lineinfo,nofma -Minfo=accel
")
endif()
if (GPU_TYPE STREQUAL a100 AND GPU_OFFLOAD STREQUAL combined)
   string(APPEND GPUFLAGS  " -acc -gpu=cc80,lineinfo,nofma -mp=gpu -
Minfo=accel")
endif()
```

Build and Run instructions

Use the scripts from the Perlmutter Scripts