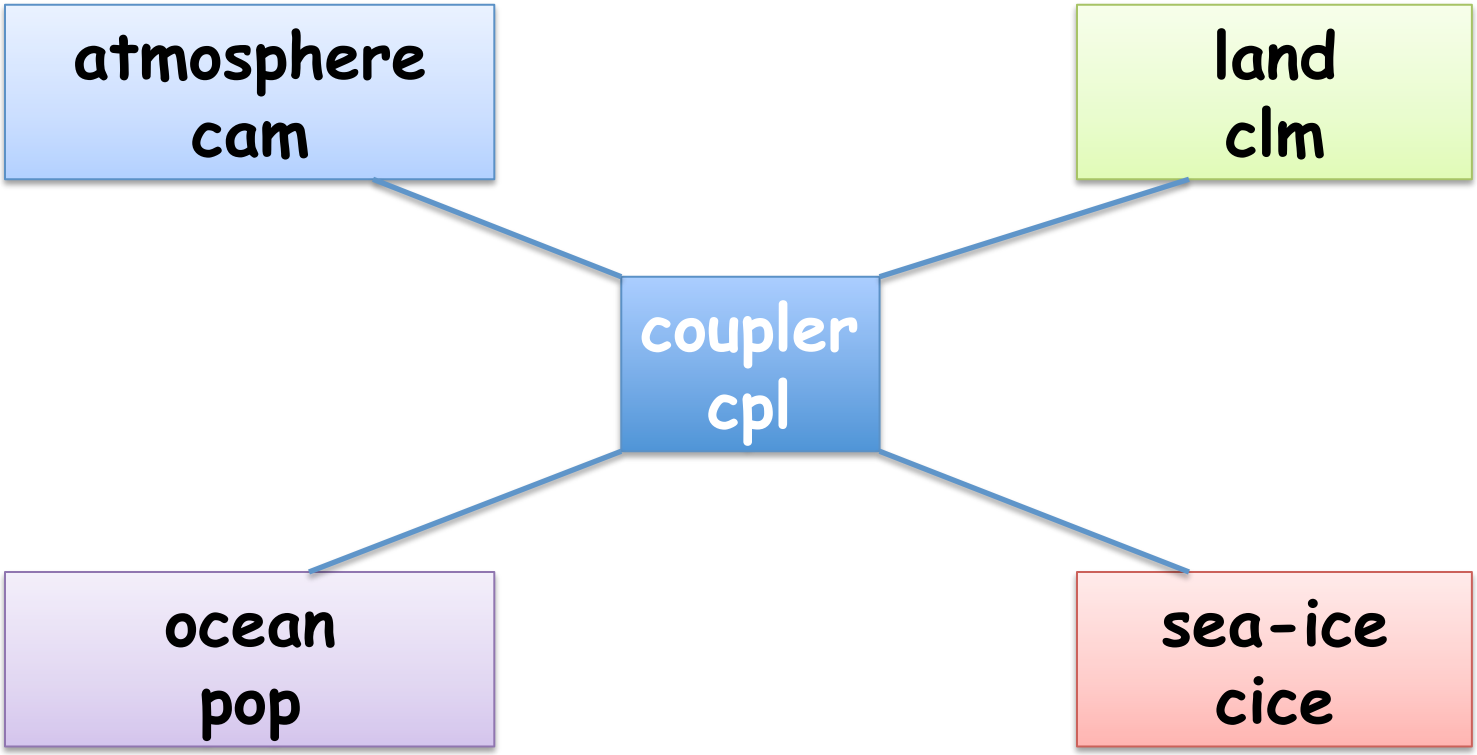
CCSM Tutorial

CCSM Overview

CCSM: Community Climate System Model



Steps in running CCSM

1. create new case

run “create_newcase”
set machine, case name, model resolution, component set

2. configure

edit env_conf.xml (namelist variables) &
    env_mach_pes.xml (MPI settings)
create scripts to generate component namelists, build & run scripts

3. build

modify scripts generating namelists (Buildconf), if necessary
put your own mods in SourceMods
generate component namelists in run directory,
compile the model

4. submit the run script

edit env_run.xml & run script
submit run script

model resolution (-res)

atmosphere, land

4x5: 46 latitudes, 72 longitudes

1.9x2.5: 96 latitudes, 144 longitudes

ocean, ice

gx3v5: 116 latitudes, 110 longitudes

gx1v5: 384 latitudes, 320 longitudes

-res 4x5_gx3v5

component sets (-compset)

B = CAM CLM POP CICE CPL (fully coupled model)

F = CAM CLM docn cice CPL (similar as standalone CAM)

-compset B_2000_TRACK1

Running CCSM: step 1 ("create_newcase")

go to the directory of the source code (ccsm4_0_beta19)

get into the "scripts" directory ($USER/ccsm4_0_beta19/scripts)

run "create_newcase" by executing

```
./create_newcase -mach bluefire
-case /blhome/$USER/$CASENAME
-res 4x5_gx3v5
-compset B_2000_TRACK1
-skip_rundb
```

Running CCSM: step 2 ("configure")

go to /blhome/$USER/$CASENAME

edit env_conf.xml & env_mach_pes.xml, if necessary

configure the model by executing

configure -case

configure -cleanall (if you make a mistake…)

scripts will be created in Buildconf &
build script & run script

RUN_TYPE (env_conf.xml)

startup: everything starts from initial conditions

branch: a model case where a new run is started but uses restart files from another case.

hybrid: a combination of startup and branch. The atm/lnd components start from initial conditions. The ocn/ice components start from restart files.

Running CCSM: step 3 (build the model)

optionally, edit scripts in "Buildconf" – e.g. cam.buildnml.csh

put your own code, if any, in "SourceMods"

build the model by executing

```
./$CASENAME.bluefire.build
```

(the model will be compiled and component namelists will be generated in run directory)

Running CCSM: step 4
submit the job

edit env_run.xml

edit the run script ($CASENAME.bluefire.run)

submit the job by executing

```
bsub < $CASENAME.bluefire.run
```

modifying xml files (xmlchange)

you may use vi or emacs

use env_run.xml as an example

```
xmlchange -file env_run.xml
-id CONTINUE_RUN -val FALSE
-id RESUBMIT -val 0
-id STOP_OPTION -val ndays
-id STOP_N -val 5
```

Archiving data
(set in env_run.xml)

short term archiving (DOUT_S)

    history files moved to
        /ptmp/$USER/archive/$CASENAME

long term archiving (DOUT_L_MS)
history files moved to NCAR Mass Storage system

edit run script

The run script is $CASENAME.bluefire.run.

#BSUB -q XXXX (standby, economy, regular, premium)

#BSUB -W YYYY (6:00 as maximum)

#BSUB -P 37591047

Example 1: vanilla run

log on bluefire

```
cp –R /fs/cgd/csm/collection/ccsm4_0_beta19 ~$USER/.

cd ccsm4_0_beta19/scripts

./create_newcase –mach bluefire
-case /blhome/$USER/ccsm_vanilla
-res 4x5_gx3v5
-compset B_2000_TRACK1
-skip_rundb

cd /blhome/$USER/ccsm_vanilla
```

Example 1: vanilla run

configure –case

./ccsm_vanilla.bluefire.build

modify run script (project number, etc)

bsub –U reservation-id < ccsm_vanilla.run

check status of your submission (bjobs)

check status of your run
run directory: /ptmp/$USER/$CASENAME/run
(in “run” directory, tail –f ccsm.log.xxxx)