Introduction to NorESM2.3

14 - 16 Jan. 2026

Alok K. Gupta, Yanchun He, Ada Gjermundsen, Tomas Torsvik,
Steve Goldhaber, Mariana Vertenstein, Matvey Debolskiy

Forskningsrådet

INES

Essential steps to use NorESM

For this workshop we will demonstrate how to set up, build and run the NorESM2 model.

1. Downloading the NorESM source code
2. Basic steps to run NorESM
3. NorESM log files
4. First look at model output
5. Overview of model structure
6. Specify the compset
7. Specify model resolution
8. Understanding your case
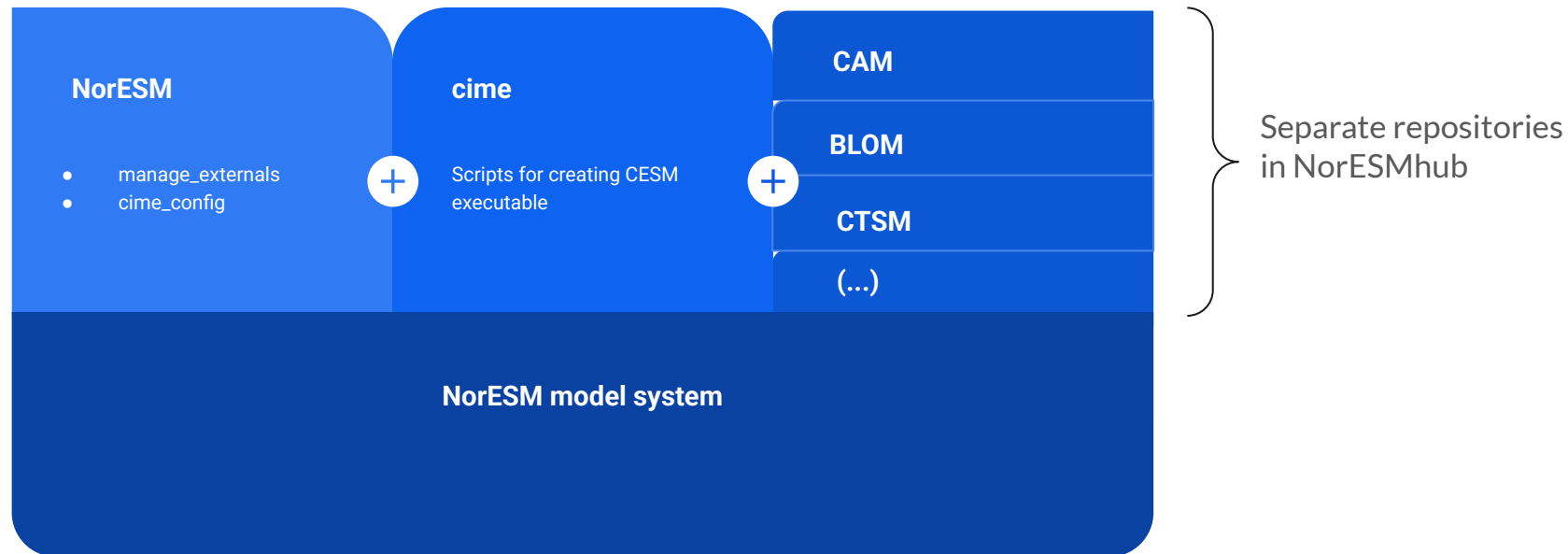9. Check your case
10. Reproducible development with scripts

Downloading the NorESM source code

- **A recipe for downloading the NorESM2.3 version of NorESM**

NorESM model structure

NorESM

- manage_externals
- cime_config

+

cime

Scripts for creating CESM executable

+

CAM

BLOM

CTSM

(...)

Separate repositories in NorESMhub

NorESM model system

Downloading the NorESM code

Recipe:

1. Log on to betzy

2. Make a directory which will contain the NorESM code and the experiments

3. Enter the folder

4. Make a directory called cases. Here you will store all your NorESM experiments.

5. Get the NorESM code from GitHub

6. Enter the noresm repository

7. Checkout the NorESM2.3 version

8. Get the corresponding code for the model component and cime

---

1. `$ ssh username@betzy.sigma2.no`

2. `$ mkdir NorESMworkshop2026`

3. `$ cd NorESMworkshop2026`

4. `$ mkdir cases`

   Optional directory structure, used for examples to follow

5. `$ git clone https://github.com/NorESMhub/NorESM.git`

6. `$ cd NorESM`

7. `$ git checkout noresm2_3_beta01`

8. `$ ./manage_externals/checkout_externals`

Documentation: https://noresm-docs.readthedocs.io/en/noresm2/access/download_code.html

All the steps needed for downloading the CMIP6 version of NorESM2 from github

1. Check out source code from gitHub
2. Check out NorESM2.3 version

3. Download source code for components (a.k.a. "externals")

```
tomast@login-3.betzy:~
$ mkdir NorESMworkshop2026
tomast@login-3.betzy:~
$ cd NorESMworkshop2026
tomast@login-3.betzy:~/NorESMworkshop2026
$ git clone https://github.com/NorESMhub/NorESM.git
Cloning into 'NorESM'...
remote: Enumerating objects: 11884, done.
remote: Counting objects: 100% (2162/2162), done.
remote: Compressing objects: 100% (424/424), done.
remote: Total 11884 (delta 1839), reused 1738 (delta 1738), pack-reused 9722 (from 4)
Receiving objects: 100% (11884/11884), 41.86 MiB | 31.87 MiB/s, done.
Resolving deltas: 100% (7723/7723), done.
tomast@login-3.betzy:~/NorESMworkshop2026
$ cd NorESM/
tomast@login-3.betzy:~/NorESMworkshop2026/NorESM : (master)
$ git checkout noresm2_3_beta01
Note: switching to 'noresm2_3_beta01'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at c32d3e2 Merge pull request #756 from NorESMhub/pr-noresm2_3_beta01
tomast@login-3.betzy:~/NorESMworkshop2026/NorESM : ((HEAD detached at noresm2_3_beta01))
$
```

```
tomast@login-3.betzy:~/NorESMworkshop2026/NorESM : ((HEAD detached at noresm2_3_beta01))
$ ./manage_externals/checkout_externals
Processing externals description file : Externals.cfg
Checking status of externals: cam, cice, cime, cism, clm, mosart, pop, blom, rtm, ww3,
Checking out externals: cam, cice, cime, clm, mosart, blom,
Processing externals description file : Externals_CAM.cfg
Checking out externals: oslo_aero,
Processing externals description file : Externals_CLM.cfg
Checking out externals: fates, ptclm,
Processing externals description file : Externals_BLOM.cfg
Checking out externals: cvmix, m4ago,

tomast@login-3.betzy:~/NorESMworkshop2026/NorESM : ((HEAD detached at noresm2_3_beta01))
$
```

NOTE: The NorESM repository does not contain all the source code for the NorESM model, but mainly framework scripts and configuration files to manage and build NorESM from external sources.

Fetching the source: *checkout_externals* script

./manage_externals/checkout_externals   : Fetch model components defined in the Externals.cfg file

./checkout_externals -e [EXTERNALS] : Fetch model components defined in [EXTERNALS] file

./checkout_externals -S   : Check status of downloaded model components

./checkout_externals -h   : See all options for checkout_externals

Entries from **Externals.cfg** file :

```
[cam]
tag = cam_cesm2_3_v1.0.0d
protocol = git
repo_url = https://github.com/NorESMhub/CAM
local_path = components/cam
externals = Externals_CAM.cfg
required = True
```

```
[clm]
tag = release-clm5.0.14-Nor_v1.0.7
protocol = git
repo_url = https://github.com/NorESMhub/ctsm
local_path = components/clm
externals = Externals_CLM.cfg
required = True
```

NOTE: Components used for standard NorESM build are enabled with the "required = True" option.

Basic steps to run NorESM

- **The Newbies Guide**
  (see NorESM2 documentation for further details)

How set up and run a standard NorESM case by executing 4 steps:

1. create a new case (the **create_newcase** script)

2. configure case (the **case.setup** script)

3. build case (the **case.build** script)

4. submit case (the **case.submit** script)

It is recommended to encapsulate this procedure, and any configuration modifications, in a run shell script. We will look at examples of this later.

Basic steps to run NorESM: Create case

The **create_newcase** script is an executable python script located in cime/scripts

```
./create_newcase --case $PATH_TO_cases/$CASENAME --mach $MACHINE --res $RESOLUTION
--compset $COMPSET --project $PROJECT --pecount={size of processor count}
```

--case : path and name of the NorESM experiment you are creating
--mach : name of the HPC you are using e.g. betzy, olivia, nebula
--res : grid resolution, e.g. 1 degree (f09_tn14) or 2 degree (f19_tn14) atmosphere/land resolution
--compset : compset name, e.g. piControl (N1850frc2), historical, ssp585, abrupt-4xCO2
--project : which project provides the cpu hours you are using (e.g. nn9560k = INES2)

```
$ cd ~/NorESMworkshop2026/NorESM/cime/scripts/

$ ./create_newcase --case ~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260114 --mach betzy
--res f19_tn14 --compset N1850frc2 --project nn9560k  --pecount=S --run-unsupported
```

Required for undocumented model configurations

```
./create_newcase --help  will provide you all input options including a description
```

Basic steps to run NorESM: Create case

```
tomast@login-3.betzy:~/NorESMworkshop2026/NorESM : ((HEAD detached at noresm2_3_beta01))
$ cd cime/scripts/
tomast@login-3.betzy:~/.../NorESM/cime/scripts : ((HEAD detached at cime5.6.10_NorESM2_3_r7))
$ ./create_newcase --case ~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113 --mach betzy --res f19_tn14 --compset N1850frc2 --project nn9560k --pecount=S --run-unsupported
Compset longname is 1850_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS
Compset specification file is /cluster/home/tomast/NorESMworkshop2026/NorESM/cime/../cime_config/config_compsets.xml
Compset forcing is 1850
ATM component is CAM cam6 physics:
LND component is clm5.0:BGC (vert. resol. CN and methane) with prognostic crop:
ICE component is Sea ICE (cice) model version 5 :with NORESM modifications appropriate for CMIP6 experiments
OCN component is BLOM default:BLOM/Ecosystem:
ROF component is MOSART: MOdel for Scale Adaptive River Transport
GLC component is Stub glacier (land ice) component
WAV component is Stub wave component
ESP component is
Pes     specification file is /cluster/home/tomast/NorESMworkshop2026/NorESM/cime/../cime_config/config_pes.xml
Compset specific settings: name is RUN_STARTDATE and value is 0001-01-01
Could not find machine match for 'login-3.betzy' or 'login-3'
Machine is betzy
Pes setting: grid match    is a%1.9x2.5.+l%1.9x2.5.+oi%tnx1v4
Pes setting: machine match is betzy
Pes setting: pesize match  is S
Pes setting: grid          is a%1.9x2.5_l%1.9x2.5_oi%tnx1v4_r%r05_g%null_w%null_m%tnx1v4
Pes setting: compset       is 1850_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS
Pes setting: tasks         is {'NTASKS_ATM': 384, 'NTASKS_LND': 136, 'NTASKS_ROF': 128, 'NTASKS_ICE': 236, 'NTASKS_OCN': 123, 'NTASKS_GLC': 128, 'NTASKS_WAV': 12, 'NTASKS_CPL': 384}
Pes setting: threads       is {'NTHRDS_ATM': 1, 'NTHRDS_LND': 1, 'NTHRDS_ROF': 1, 'NTHRDS_ICE': 1, 'NTHRDS_OCN': 1, 'NTHRDS_GLC': 1, 'NTHRDS_WAV': 1, 'NTHRDS_CPL': 1}
Pes setting: rootpe        is {'ROOTPE_ATM': 0, 'ROOTPE_LND': 0, 'ROOTPE_ROF': 0, 'ROOTPE_ICE': 136, 'ROOTPE_OCN': 384, 'ROOTPE_GLC': 0, 'ROOTPE_WAV': 372, 'ROOTPE_CPL': 0}
Pes setting: pstrid        is {}
Pes other settings: {}
Pes comments: none
 Compset is: 1850_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS
 Grid is: a%1.9x2.5_l%1.9x2.5_oi%tnx1v4_r%r05_g%null_w%null_m%tnx1v4
 Components in compset are: ['cam', 'clm', 'cice', 'blom', 'mosart', 'sglc', 'swav', 'sesp', 'drv', 'dart']
No charge_account info available, using value from PROJECT
No project info available
cesm model version found: noresm2_3_beta01
Batch_system_type is slurm_nor
job is case.run USER_REQUESTED_WALLTIME None USER_REQUESTED_QUEUE None
job is case.st_archive USER_REQUESTED_WALLTIME None USER_REQUESTED_QUEUE None
 Creating Case directory /cluster/home/tomast/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113
This component includes user_mods /cluster/home/tomast/NorESMworkshop2026/NorESM/components/cice/cime_config/usermods_dirs/noresm-cmip6
Adding user mods directory /cluster/home/tomast/NorESMworkshop2026/NorESM/components/cice/cime_config/usermods_dirs/noresm-cmip6
tomast@login-3.betzy:~/.../NorESM/cime/scripts : ((HEAD detached at cime5.6.10_NorESM2_3_r7))
$
```

README

$ cd ~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260114/

$ less README.case (or "$ cat README.case")

You will find the information about your case in README.case

Including compset long name, grid files, components, git branch, git commit etc.

README.case can be very useful if you want to reproduce a case (either your own or somebody else's case)

Basic steps to run NorESM: Set up the case

Enter the case folder:

```
$ cd ~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260114/

$ ./case.setup

$ ./preview_run (optional: get info about node count, env variables, job walltime and account)
```

```
tomast@login-3.betzy:~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113
$ cd ~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113/
tomast@login-3.betzy:~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113
$ ./case.setup
Setting resource.RLIMIT_STACK to -1 from (8388608, -1)
/cluster/home/tomast/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113/env_mach_specific.xml already exists, delete to replace
job is case.run USER_REQUESTED_WALLTIME None USER_REQUESTED_QUEUE None
Creating batch scripts
Writing case.run script from input template /cluster/home/tomast/NorESMworkshop2026/NorESM/cime/config/cesm/machines/template.case.run
Creating file .case.run
Writing case.st_archive script from input template /cluster/home/tomast/NorESMworkshop2026/NorESM/cime/config/cesm/machines/template.st_archive
Creating file case.st_archive
Creating user_nl_xxx files for components and cpl
If an old case build already exists, might want to run 'case.build --clean' before building
You can now run './preview_run' to get more info on how your case will be run
tomast@login-3.betzy:~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113
$
```

Basic steps to run NorESM: Build the case

After running ./case.setup you should see your case in the noresm run directory

$ ls $USERWORK/noresm/N1850frc2_f19_tn14_test01_20260114/
     bld   : build directory
     run   : run directory

$USERWORK : alias for **/cluster/work/users/$USER/**

bld: the build folder. Here you find compiled source code for the model
run: the run folder. Here you find logs, output data, restart files etc. More on that later …

$ ./case.build

Creating an executable **cesm.exe**; found in $USERWORK/noresm/$CASENAME/bld

Basic steps to run NorESM: Building the case

case.build
- Create component namelist files
- Build components
- Build cesm executable

If everything builds correctly:
MODEL BUILD HAS FINISHED SUCCESSFULLY

Basic steps to run NorESM: Submitting the case

```
$ ./case.submit
```

```
tomast@login-3.betzy:~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113
$ ./case.submit
Setting resource.RLIMIT_STACK to -1 from (-1, -1)
Creating component namelists
    Calling /cluster/home/tomast/NorESMworkshop2026/NorESM/components/cam//cime_config/buildnml
CAM namelist copy: file1 /cluster/home/tomast/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113/Buildconf/camc
    Calling /cluster/home/tomast/NorESMworkshop2026/NorESM/components/clm//cime_config/buildnml
    Calling /cluster/home/tomast/NorESMworkshop2026/NorESM/components/cice//cime_config/buildnml
    Calling /cluster/home/tomast/NorESMworkshop2026/NorESM/components/blom//cime_config/buildnml
/cluster/home/tomast/NorESMworkshop2026/NorESM/components/blom/cime_config/namelist_definition_blom.xml validates
    Calling /cluster/home/tomast/NorESMworkshop2026/NorESM/components/mosart//cime_config/buildnml
    Calling /cluster/home/tomast/NorESMworkshop2026/NorESM/cime/src/components/stub_comps/sglc/cime_config/buildnml
    Calling /cluster/home/tomast/NorESMworkshop2026/NorESM/cime/src/components/stub_comps/swav/cime_config/buildnml
    Calling /cluster/home/tomast/NorESMworkshop2026/NorESM/cime/src/components/stub_comps/sesp/cime_config/buildnml
    Calling /cluster/home/tomast/NorESMworkshop2026/NorESM/cime/src/drivers/mct/cime_config/buildnml
Finished creating component namelists
Checking that inputdata is available as part of case submission
Setting resource.RLIMIT_STACK to -1 from (-1, -1)
Loading input file list: 'Buildconf/cam.input_data_list'
Loading input file list: 'Buildconf/cpl.input_data_list'
Loading input file list: 'Buildconf/blom.input_data_list'
Loading input file list: 'Buildconf/cice.input_data_list'
Loading input file list: 'Buildconf/mosart.input_data_list'
Loading input file list: 'Buildconf/clm.input_data_list'
Check case OK
submit_jobs case.run
Submit job case.run
Submitting job script sbatch --time 00:59:00 --account nn9560k .case.run --resubmit
Submitted job id is 1402181
Submit job case.st_archive
Submitting job script sbatch --time 0:59:00 --account nn9560k  --dependency=afterok:1402181 case.st_archive --resubmit
Submitted job id is 1402182
Submitted job case.run with id 1402181
Submitted job case.st_archive with id 1402182
tomast@login-3.betzy:~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113
$
```

case.submit
- Create final versions of component namelists
- Check input data

case.submit submits two jobs to the queue:

- Run job

- Archive job

Monitoring your jobs: some useful commands

squeue: overview of job(s) running and the job id(s)

scontrol: see more details about the job running

scancel: stop job from running

```
$ squeue -u $USER

$ squeue -p $PROJECT

$ scontrol show job $JOBID

$ scancel $JOBID
```

Note! squeue -u $USER: If you don't see your job, it has either finished or crashed!

```
tomast@login-3.betzy:~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113
$ squeue -u tomast
         JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
       1402181    normal N1850frc   tomast  R       0:57      4 b[3343-3346]
       1402182    preproc N1850frc   tomast PD       0:00      1 (Dependency)
tomast@login-3.betzy:~/NorESMworkshop2026/cases/N1850frc2_f19_tn14_test01_20260113
$ 
```

Monitoring jobs : https://documentation.sigma2.no/jobs/monitoring.html

1st hands-on session

Resources for NorESM workshop

Betzy for NorESM user workshop

During the user workshop we have access to Betzy.
Most user will have access to project nn9560k until 18th Jan.

ReservationName = `noresm_course` (only for today)
Accounts   = nn9560k
PartitionName   = normal

We are going to use mostly *normal* queue for simulation and it could have a long waiting time and you have access for around 5 days.

Also, login to betzy system: login.betzy.sigma2.no

code for hands-on-session

Cloning repo NorESM :

git clone https://github.com/NorESMhub/NorESM.git **NorESM_beta23**

Checking out tag noresm2_3_beta01 :

git checkout -b **noresm2_3_beta01** tags/noresm2_3_beta01

Download NorESM code spreads on different repositories:

./manage_externals/checkout_externals

Exercise: 1

Run compset **N1850frc2** with grid **f19_tn14** for **1 month**

Step 1:- create experiment

mkdir cases

cd cases

../cime/scripts/create_newcase --case **n1850_f19_tn14_23beta3M** --compset N1850frc2 --res f19_tn14 --project nn9560k --machine betzy --run-unsupported --pecount=S

Step-2 setup and build

./case.setup

./case.build

Step-3 set run length and submit

./xmlchange STOP_OPTION=nmonths

./xmlchange STOP_N=1

./case.submit   **check file** CaseStatus, README.case

Exercise 2:

Continue same run for next 11 months:

./xmlchange STOP_N=11

./xmlchange CONTINUE_RUN=TRUE

./xmlchange --subgroup case.run JOB_WALLCLOCK_TIME=04:00:00

./case.submit

Needed for exercises?

Configure run settings: env_run.xml

Setting the length of the simulation:

- STOP_OPTION : nseconds,nsecond,nminutes,nhours,nhour,ndays,nmonths, nyears
- STOP_N: numerical value

```
./xmlchange STOP_OPTION=nmonths,STOP_N=6
```

Writing restart files in middle of simulation:
Restart files are written end of the simulation by default; But, if you are having a long simulation of 100 years; for safety reason you want to write restart files at some frequency you can set below option

- REST_OPTION: nseconds,nsecond,nminutes,nminute,nhours,nhour,nmonths, nyears
- REST_N: numerical value
- DOUT_S_SAVE_INTERIM_RESTART_FILES: TRUE or FALSE. Set to TRUE if you want to archive all restart files and FALSE if you only want to archive restart files from the end of the simulation

Documentation:
- https://noresm-docs.readthedocs.io/en/noresm2/configurations/experiment_environment.html
- https://noresm-docs.readthedocs.io/en/noresm2/configurations/experiment_environment.html#some-common-configuration-settings

Configure run settings: env_run.xml

Continue a simulation: when you are having WALLCLOCK time limitation on system. For example, you want to have 200 years simulation and WALLCLOCK time limitation is 5 days; you are able to simulate 10 model years/day; to complete 200 model years simulation set RESUBMIT=3, STOP_N to 50 and STOP_OPTION to nyears

- **CONTINUE_RUN** in env_run.xml; TRUE or FALSE.
    - You need all restart files and rpointer.* files in run folder.
    - Please note that CONTINUE_RUN needs to be FALSE first time you submit an experiment.
    - Will automatically be set to TRUE if the job is automatically resubmitted, i.e. if RESUBMIT > 0

- **RESUBMIT** in env_run.xml ; an integer value.
    - will auto resubmit till specified value; you will have total simulation period STOP_N*(RESUBMIT+1)

Documentation:
- https://noresm-docs.readthedocs.io/en/noresm2/configurations/experiment_environment.html
- https://noresm-docs.readthedocs.io/en/noresm2/configurations/experiment_environment.html#some-common-configuration-settings

Configure run settings: env_run.xml

Branch run
In a branch run, all components are initialized using a consistent set of restart files from a previous run. Mostly used for tuning experiments and investigating parameter space
- RUN_TYPE to "branch"
- RUN_REFDIR directory containing reference data
- RUN_REFCASE name of reference case
- RUN_REFDATE Reference date branch run
- GET_REFCASE TRUE else you have to copy data to run folder

Hybrid run
Not as strict as a branch run, all components are initialized but can have reference files from several experiments. Used e.g. for a historical experiment starting from piControl
- RUN_TYPE to "hybrid"
- RUN_REFDIR directory containing reference data
- RUN_REFCASE name of reference case
- RUN_REFDATE Reference date branch run
- GET_REFCASE TRUE else you have to copy data to run folder
- RUN_STARTDATE set the date for the beginning of the simulation

Documentation:
- https://noresm-docs.readthedocs.io/en/noresm2/configurations/experiment_environment.html
- https://noresm-docs.readthedocs.io/en/noresm2/configurations/experiment_environment.html#some-common-configuration-settings

Configure run settings: env_run.xml

Two scripts which you will find helpful (located in the case folder):
1. xmlquery: provides the information and its value which are set in the *.xml files, e.g. env_run.xml
2. xmlchange: used to change values/parameters set in the *.xml files, e.g. in env_run.xml

```
$ ./xmlquery --value STOP_OPTION,STOP_N

ndays,5

$ ./xmlchange STOP_OPTION=nyears,STOP_N=1

$ ./xmlquery --value STOP_OPTION

nyears,1
```

If you want to look at the env_run.xml file:

```
$ vi env_run.xml
```

Documentation:
- https://noresm-docs.readthedocs.io/en/noresm2/configurations/experiment_environment.html
- https://noresm-docs.readthedocs.io/en/noresm2/configurations/experiment_environment.html#some-common-configuration-settings

Configure run setting: env_batch.xml

You also need to modify **env_batch.xml:**

- XML block for case.run
- env_batch.xml sets the arguments for the batch job commands
- Neet to set JOB_WALLCLOCK_TIME
- XML block for case.st_archive
- case.st_archive is a pending job which moves files from the run directory to the archive directory after a successful simulation.
- also here you need to set JOB_WALLCLOCK_TIME
- you can also modify project for CPU hours if required
  (usually it is set during experiment creation, but you may need to change it)

```
./xmlchange --subgroup case.st_archive JOB_WALLCLOCK_TIME=00:25:00
```

After setting the walltime for the two jobs, you can submit your case:

```
$ ./case.submit
```

Documentation:
https://noresm-docs.readthedocs.io/en/noresm2/configurations/experiment_environment.html#batch-job-environment

Log files

Log files produced by NorESM

NorESM produce log output for case creation, build process, run process and archiving process.

Log files in the case directory:

CaseStatus : Summary of all processes (case setup, build, job submission and job completion).
        Good place to start looking at the logs!

README.case : Log for case creation

software_environment.txt : System environments (modules) used during model build

logs/run_environment.txt.<id>.<time> : System environment (modules) loaded during model run.

<casename> : Log for job execution, CPU, memory and disk usage.

Documentation: https://noresm-docs.readthedocs.io/en/noresm2/output/noresm_logs.html

Log files produced by NorESM

Build logs : <workdir>/noresm/<casename>/bld/   (<workdir> = /cluster/work/users/<username>)

      cesm.bldlog.<timestamp>.gz   : Build the coupled model executable cesm.exe.
      <component>.bldlog.<timestamp>.gz   : Build log for individual model components.

Run logs : <workdir>/archive/<casename>/logs/   (assuming run and archive processes finished)

      cesm.log.<jobid>.<timestamp>.gz   : Run log for coupled model system.
      <component>.log.<jobid>.<timestamp>.gz   : Run log for individual model components.

If NorESM fails during run, you may find these log files in <workdir>/noresm/<casename>/run/ .

Archiving logs : <workdir>/archive/<casename>/

      archive.log.<timestamp>   : Log for short term archiving of model output.
      case.log   : Log for archiving of case directory.

NOTE: Archived *.gz files can be read directly using the "less" command, but can otherwise be expanded before reading: gunzip <filename>.gz

Experiment status and timing statistics

In your case directory: the file CaseStatus logs all information on what you have done and if it worked or not:

```
2021-11-10 14:45:01: case.setup starting
 ---------------------------------------------------
2021-11-10 14:45:02: case.setup success
 ---------------------------------------------------
2021-11-10 14:45:07: case.build starting
 ---------------------------------------------------
2021-11-10 14:51:31: case.build success
 ---------------------------------------------------
2021-11-10 14:52:04: case.submit starting
 ---------------------------------------------------
2021-11-10 14:52:12: case.submit error
ERROR: Command: 'sbatch --time 00:59:00 -q devel --account nn9560k .case.run --resubmit' failed with error 'sbatch: error: QOSGrpNodeLimit
sbatch: error: Batch job submission failed: Job violates accounting/QOS policy (job submit limit, user's size and/or time limits)' from dir '/cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test02_2021
1115'
 ---------------------------------------------------
2021-11-10 14:52:52: case.submit starting
 ---------------------------------------------------
2021-11-10 14:52:59: case.submit error
ERROR: Command: 'sbatch --time 00:29:00 -q devel --account nn9560k .case.run --resubmit' failed with error 'sbatch: error: QOSGrpNodeLimit
sbatch: error: Batch job submission failed: Job violates accounting/QOS policy (job submit limit, user's size and/or time limits)' from dir '/cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test02_2021
1115'
 ---------------------------------------------------
2021-11-10 14:53:32: case.submit starting
 ---------------------------------------------------
2021-11-10 14:53:39: case.submit error
ERROR: Command: 'sbatch --time 00:29:00 -q devel --account nn9560k .case.run --resubmit' failed with error 'sbatch: error: QOSGrpNodeLimit
sbatch: error: Batch job submission failed: Job violates accounting/QOS policy (job submit limit, user's size and/or time limits)' from dir '/cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test02_2021
1115'
 ---------------------------------------------------
2021-11-10 14:57:08: case.setup starting
 ---------------------------------------------------
2021-11-10 14:57:09: build.clean starting
 ---------------------------------------------------
2021-11-10 14:57:27: build.clean success
 ---------------------------------------------------
2021-11-10 14:57:28: case.setup success
 ---------------------------------------------------
2021-11-10 14:57:52: case.build starting
 ---------------------------------------------------
2021-11-10 14:59:37: case.build success
 ---------------------------------------------------
2021-11-10 15:00:26: case.submit starting
 ---------------------------------------------------
2021-11-10 15:00:34: case.submit success case.run:253485, case.st_archive:253486
 ---------------------------------------------------
2021-11-10 15:05:37: case.run starting
 ---------------------------------------------------
2021-11-10 15:05:46: model execution starting
 ---------------------------------------------------
2021-11-10 15:15:35: model execution success
 ---------------------------------------------------
2021-11-10 15:15:35: case.run success
 ---------------------------------------------------
2021-11-10 15:15:56: st_archive starting
 ---------------------------------------------------
2021-11-10 15:16:29: st_archive success
 ---------------------------------------------------
```

Experiment status and timing statistics

In your case directory: in the timing sub-directory; the file cesm_timing.$CASE provides information on grid type, run length, compset, processors configuration and many others.
Most important are timing statics : Model throughput, Model cost and run time

```
Overall Metrics:
  Model Cost:              985.36   pe-hrs/simulated_year
  Model Throughput:          3.12   simulated_years/day

  Init Time   :    201.411 seconds
  Run Time    :    379.632 seconds      75.926 seconds/day
  Final Time  :      0.183 seconds

  Actual Ocn Init Wait Time    :      5.450 seconds
  Estimated Ocn Init Run Time  :      2.344 seconds
  Estimated Run Time Correction :     0.000 seconds
    (This correction has been applied to the ocean and total run times)

Runs Time in total seconds, seconds/model-day, and model-years/wall-day
CPL Run Time represents time in CPL pes alone, not including time associated with data exchange with other components

  TOT Run Time:    379.632 seconds      75.926 seconds/mday        3.12 myears/wday
  CPL Run Time:      8.458 seconds       1.692 seconds/mday      139.93 myears/wday
  ATM Run Time:    233.509 seconds      46.702 seconds/mday        5.07 myears/wday
  LND Run Time:    106.004 seconds      21.201 seconds/mday       11.17 myears/wday
  ICE Run Time:     36.725 seconds       7.345 seconds/mday       32.23 myears/wday
  OCN Run Time:    281.239 seconds      56.248 seconds/mday        4.21 myears/wday
  ROF Run Time:     18.111 seconds       3.622 seconds/mday       65.35 myears/wday
  GLC Run Time:      0.000 seconds       0.000 seconds/mday        0.00 myears/wday
  WAV Run Time:      0.000 seconds       0.000 seconds/mday        0.00 myears/wday
  ESP Run Time:      0.000 seconds       0.000 seconds/mday        0.00 myears/wday
  CPL COMM Time:   127.679 seconds      25.536 seconds/mday        9.27 myears/wday
```

Model Throughput increases to
~20 simulated_years/day
if you use 4 nodes instead of 1

Experiment status and timing statistics

In your RUN directory: /cluster/work/users/$USER/noresm/$CASE/run/
You will find all the namelists; component_in files, the timing folder, restart files and rpointer.* and if your model simulation for some reason crashes; the log files.

- the cesm.log.$JOBID file can provide some hints of which component caused the crash
- the component.log.$JOBID file can provide information about which subroutine caused the crashed
- Log timestamps can be a useful way to keep track of run evolution after several model runs

If your model simulation is successful, the log files are archived in:
/cluster/work/users/$USER/archive/$CASE/logs/

For a successful simulation the cpl.log.$JOBID.<timestamp>.gz file will end with:

```
(seq_mct_drv): ===============          SUCCESSFUL TERMINATION OF CPL7-cesm ===============
(seq_mct_drv): ===============      at YMD,TOD =    00010106        0        ===============
(seq_mct_drv): ===============  # simulated days (this run) =        5.000   ===============
(seq_mct_drv): ===============  compute time (hrs)          =        0.105   ===============
(seq_mct_drv): ===============  # simulated years / cmp-day =        3.118   ===============
(seq_mct_drv): ===============  pes min memory highwater  (MB)      -0.001   ===============
(seq_mct_drv): ===============  pes max memory highwater  (MB)      -0.001   ===============
(seq_mct_drv): ===============  pes min memory last usage (MB)     521.612   ===============
(seq_mct_drv): ===============  pes max memory last usage (MB)    1174.614   ===============
```

Looking at model output with ncview

Looking at model output with ncview

ncview provides simple visualization of output netCDF files. Useful to check if output is produced and shows some sensible pattern. Other diagnostic tools should be used for in-depth analysis.

- Log in with X-forwarding:
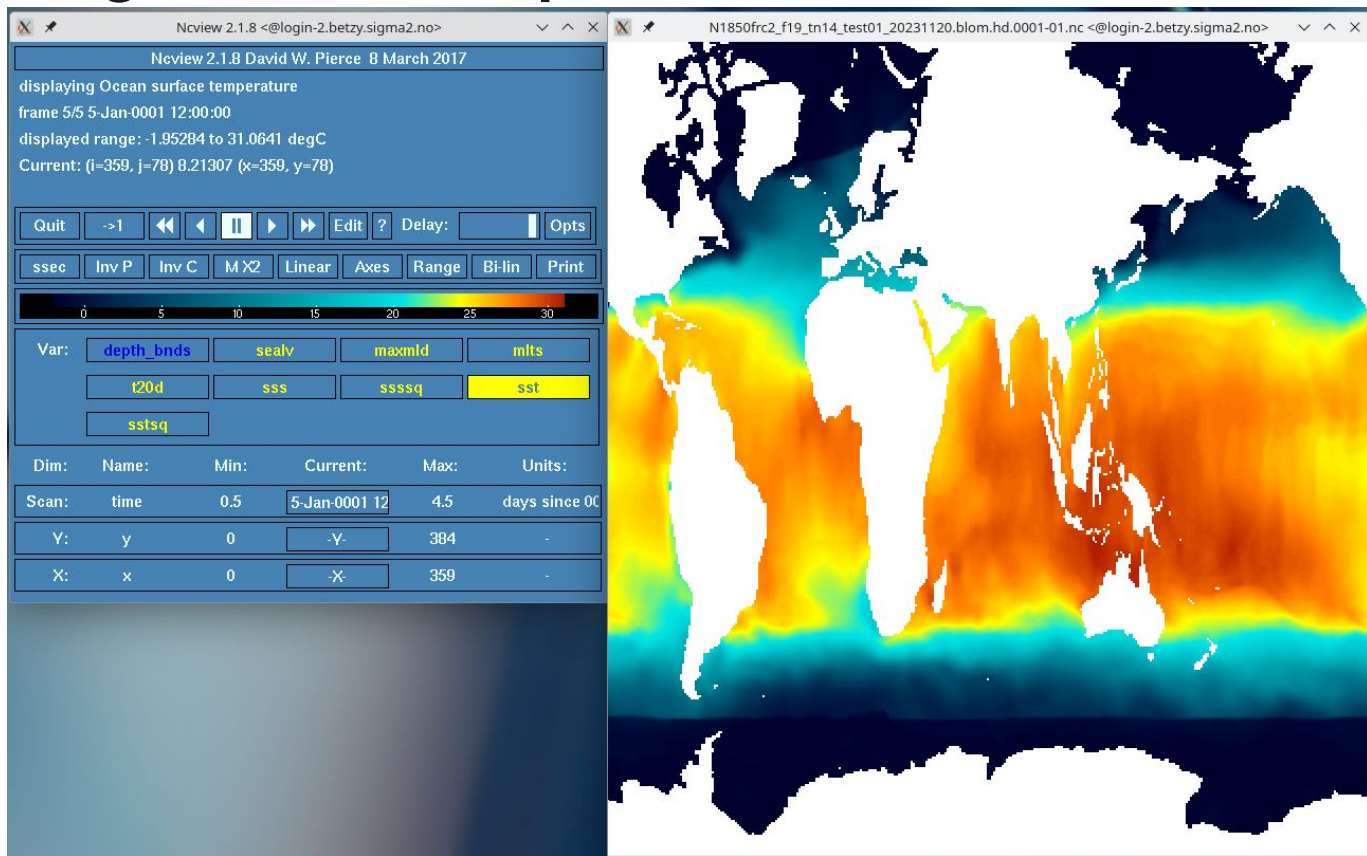 linux/mac: $ssh -Y username@betzy.sigma2.no

- Load module for ncview:
  $module load ncview/2.1.8-gompi-2022a

- Navigate to archive directory, ocean output, and launch ncview:
  $cd $USERWORK/archive/N1850frc2_f19_tn14_test01_20260114/ocn/hist
  $ncview N1850frc2_f19_tn14_test01_20260114.blom.hd.0001-01.nc
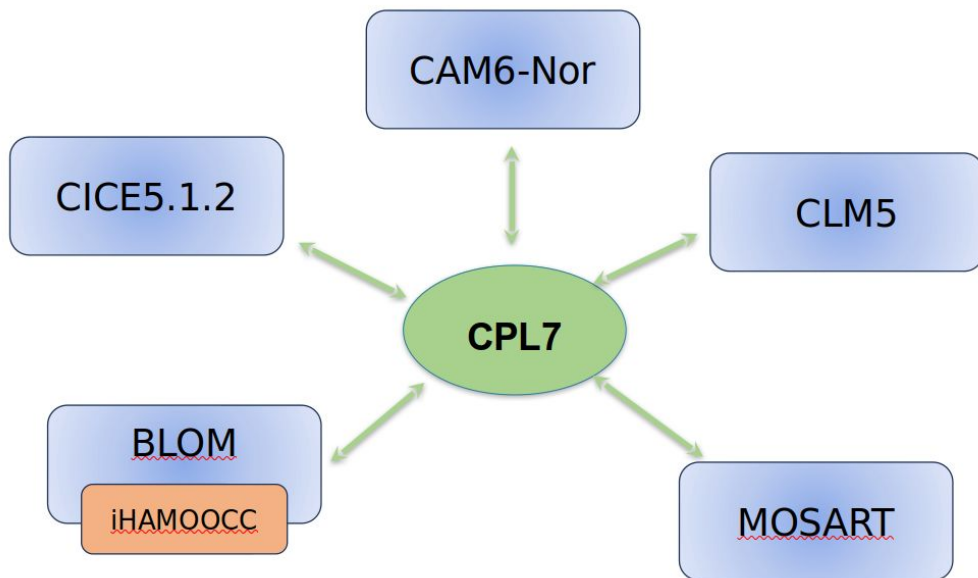
Looking at model output with ncview

Questions?

# Extra slides:
Overview of NorESM model system

NorESM communication framework

NorESM2 is based on the second version of the Community Earth System Model, CESM2, and share most of the CESM2 structure, but modifies model component.

Component communicate with each other through the COUPLER, named CPL7 for the NorESM2 model version.

cime_config directory

NorESM

cime_config/
- config_compsets.xml
- config_pes.xml

components/

blom/

cam/

cice/

clm/

mosart/
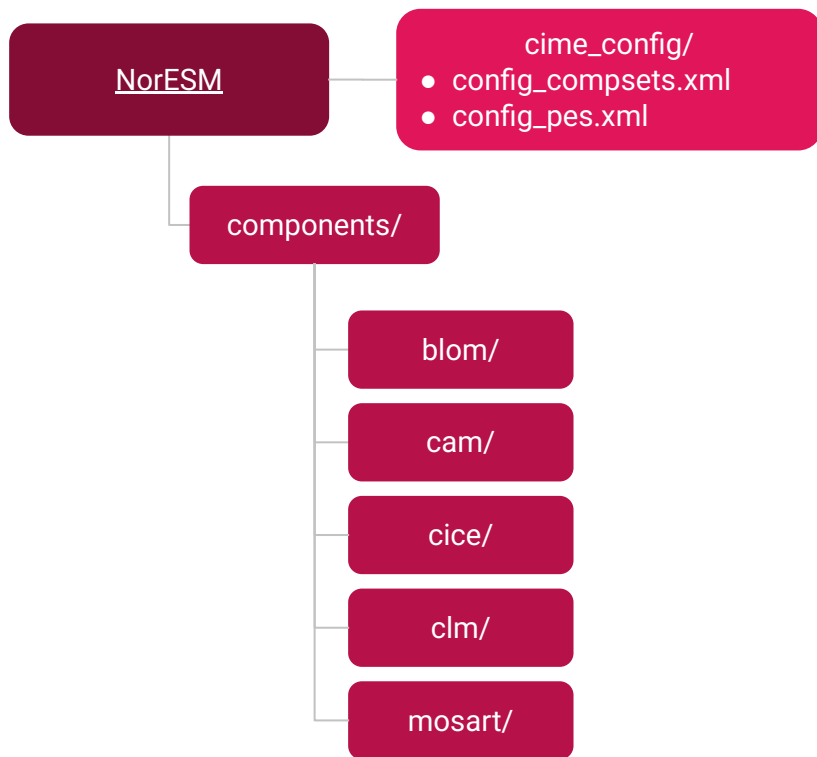
Files in cime_config/ :

config_compsets.xml : compset configuration file

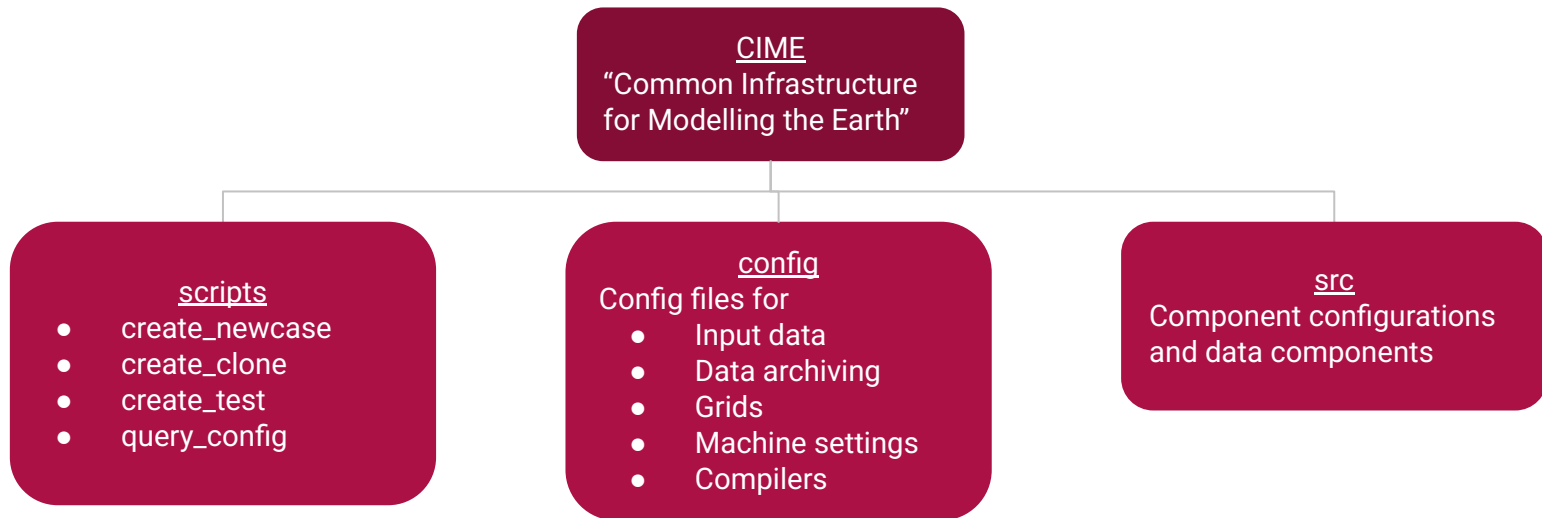**config_pes.xml** : processor config for particular compsets

Additional component scripts

buildnml : build component namelist

buildlib : build component library

cime

CIME
"Common Infrastructure for Modelling the Earth"

scripts
- create_newcase
- create_clone
- create_test
- query_config

config
Config files for
- Input data
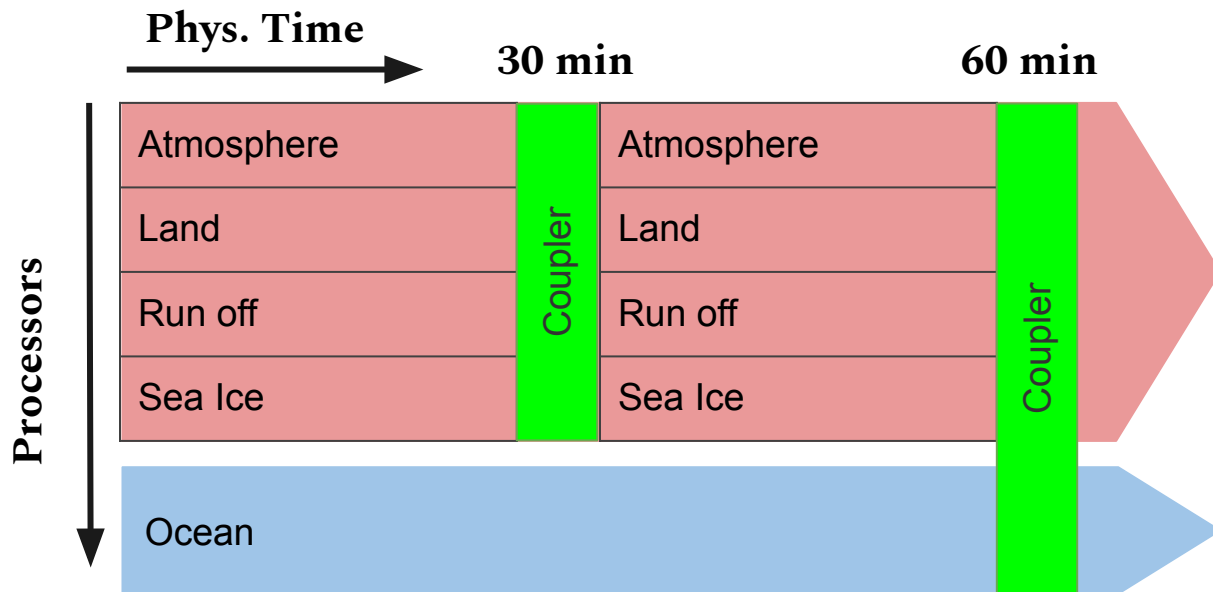- Data archiving
- Grids
- Machine settings
- Compilers

src
Component configurations and data components

Example: fully coupled model on multiple processors

Standard setup:

Ocean component usually runs separately from other components.

Atmosphere runs on 30 min physics time steps, and couples to other components except ocean each interval.

Atmosphere - Ocean coupling at 60 min physics time steps.

Extra slides:
HPC resources provided by Sigma2

Sigma2 HPC machines

Sigma2 provides 3 HPC systems:

- Betzy for large parallel jobs   (Normal queue: 4-512 nodes,   4 days max walltime)
- Olivia for intermediate parallel jobs   (Not yet configured for NorESM)
- Saga for serial or single node jobs   (Not configured for NorESM)

HPC job types: https://documentation.sigma2.no/jobs/choosing_job_types.html

	Betzy	Olivia
System	BullSequana XH2000	HPE Cray Supercomputing EX
CPU type	AMD Epyc 7742 ; 2.25GHz	AMD Epyc Turin
Nodes / Cores	1344 / 172032	252 / 64512
(core / mem) per node	128 / 256 GB	256 / 768 GB
	16 Nvidia A100 GPUs	

Fram and Betzy storage areas

Directory	Alias	Purpose
/cluster/home/$USER	$HOME	User data
/custer/work/users/$USER	$USERWORK	Staging and job data
/custer/work/jobs/$SLURM_JOB_ID	$SCRATCH	Per-job data
/cluster/projects/<project_name>		Project data
/cluster/shared/<folder_name>		Shared data

NorESM copies build and run files to subdirectories in $USERWORK. These files are not backed up,and are subject to automatic deletion after 42 days (21 days or 17 days if storage reduced to 30% or 10%, respectively). Therefore, make sure you copy output files you want to keep to a permanent storage area.

Documentation: https://documentation.sigma2.no/files_storage/clusters.html

NorESM shared resources on Fram and Betzy

Shared resources for NorESM are available on Betzy under directory:

/cluster/shared/noresm/

/cluster/shared/noresm/

WORKSHOP

inputdata/

diagnostics/

<specific_experiment_dirs>
Differs between Betzy and Fram

restart/
- common restart files

share/domains/
- grids

cpl/
- maps between grids

atm/

ocn/

lnd/

glc/

ice/

rof/

wav/