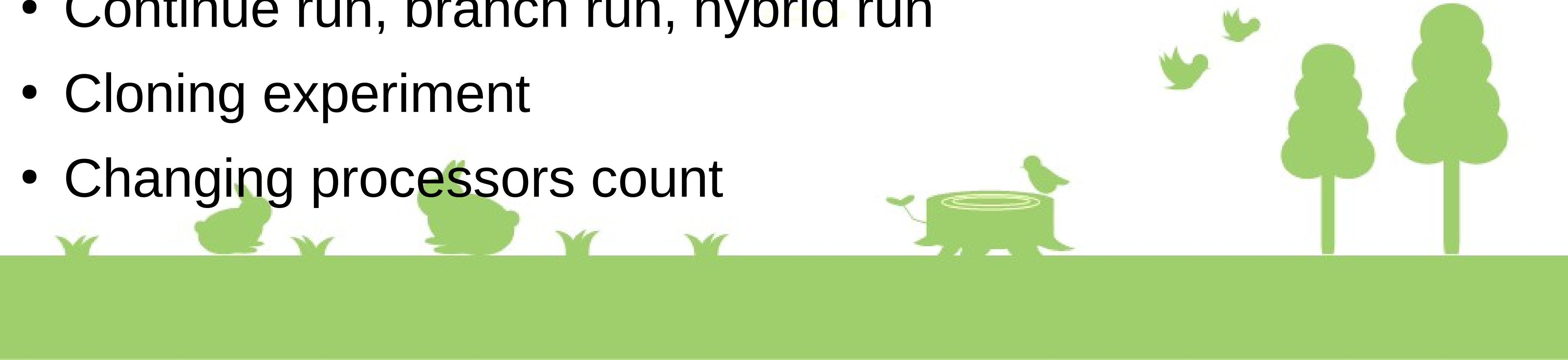
NorESM user workshop 2020

Alok K. Gupta, Yanchun He, Mats Bentsen

Agenda

- HPC infrastructures and NorESM workflow
- Downloading NorESM
- NorESM-2 structure
- Basic steps to execute model
- Few namelist variables to set up model and writing restart files
- user_nl_* files to modify namelist; SourceMods to debug source files
- Continue run, branch run, hybrid run
- Cloning experiment
- Changing processors count



HPC machines

- Fram

  System Lenovo NeXtScale nx360

  Number of Cores 32256

  Number of nodes 1006

  CPU type Intel E5-2683v4 2.1 GHz

  Intel E7-4850v4 2.1 GHz (hugemem)

  Per node 32 cores; 64 GiB memory

  Island topology; 9216 cores per island

- We are using Intel compilers on both machine

- Betzy

  BullSequana XH2000

  Number of Cores 172032

  Number of nodes 1344

  CPU type AMD® Epyc™ 7742 2.25GHz

  Per node 128 cores; 256 GiB memory

- Recently, in testing phase and have approx. 15-25% efficiency for most experiments of NorESM

Information taken from Sigma2 website :- https://www.sigma2.no/

Web services:

- **Research Data Archive** : **https://archive.norstore.no/**
- **Diagnostic output** : **http://ns2345k.web.sigma2.no/**
- **ESGF node** : **https://noresg.nird.sigma2.no/thredds/**
- **NorESM git repository** : **https://github.com/NorESMhub**
- **NIRD tootkit** : **https://apps.sigma2.no/nird**
- **NorESM documentation** : **https://noresm-docs.readthedocs.io/en/latest/**
- **NorESM inputdata Server** : **https://www.noresm.org/inputdata**

Recent technical development

- Distributed NorESM repository – Like as CESM- public
- Support for new system Betzy
- More documentations – NorESM user guide
- MICOM changes to BLOM – Bergen Layered Ocean Model
- user_nl_blom – consistent like as user_nl_cam ...
- Few bug fixes related to uninitialized variables
- Implementation of inputdata server:- CAM-OSLO and BLOM specific data can be downloaded automatically
- Implementation of CMIP6 compsets
- License

Workflow

Model git repository
cime, cam, blom, cice,
clm , rtm, ..

https://github.com/NorESMhub

Model Simulation

Mostly, shared folder

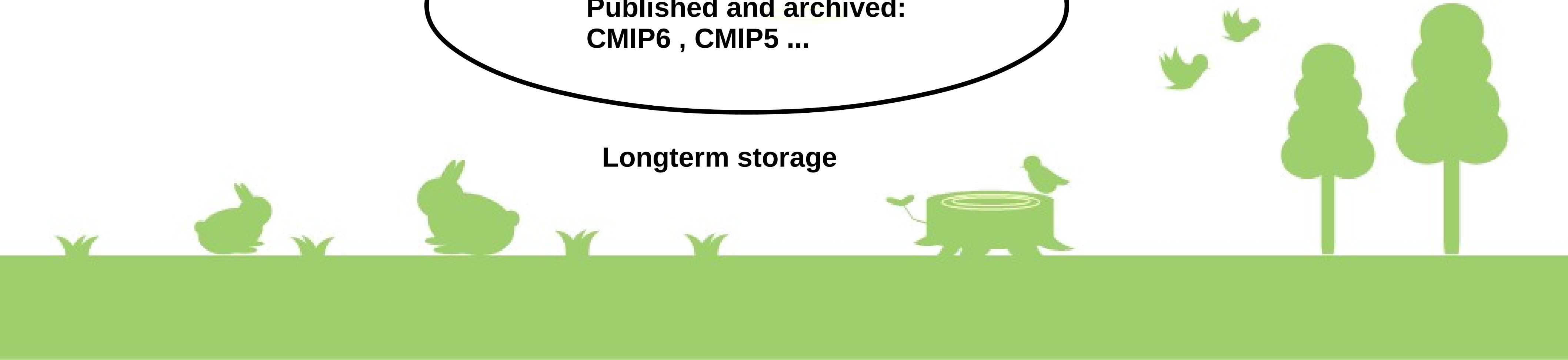
FRAM – Betzy – HPC Machines

Post-processing
Diagnostics,
CMORization

NIRD - National e-Infrastructure for Research Data

Published and archived:
CMIP6 , CMIP5 ...

Longterm storage



Downloading CMIP6 version of NorESM

- git clone – clone the remote repository locally

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

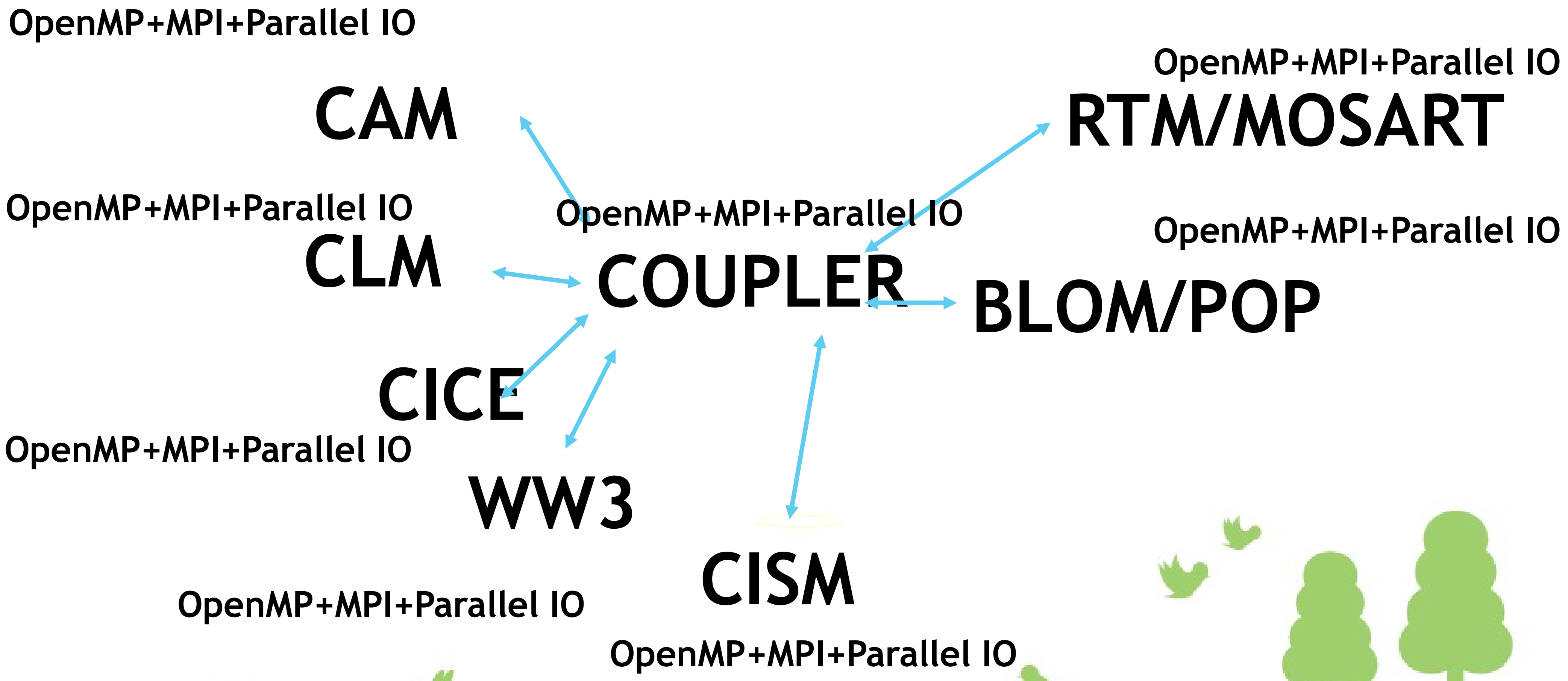
  cd NorESM

  git tag list all tags

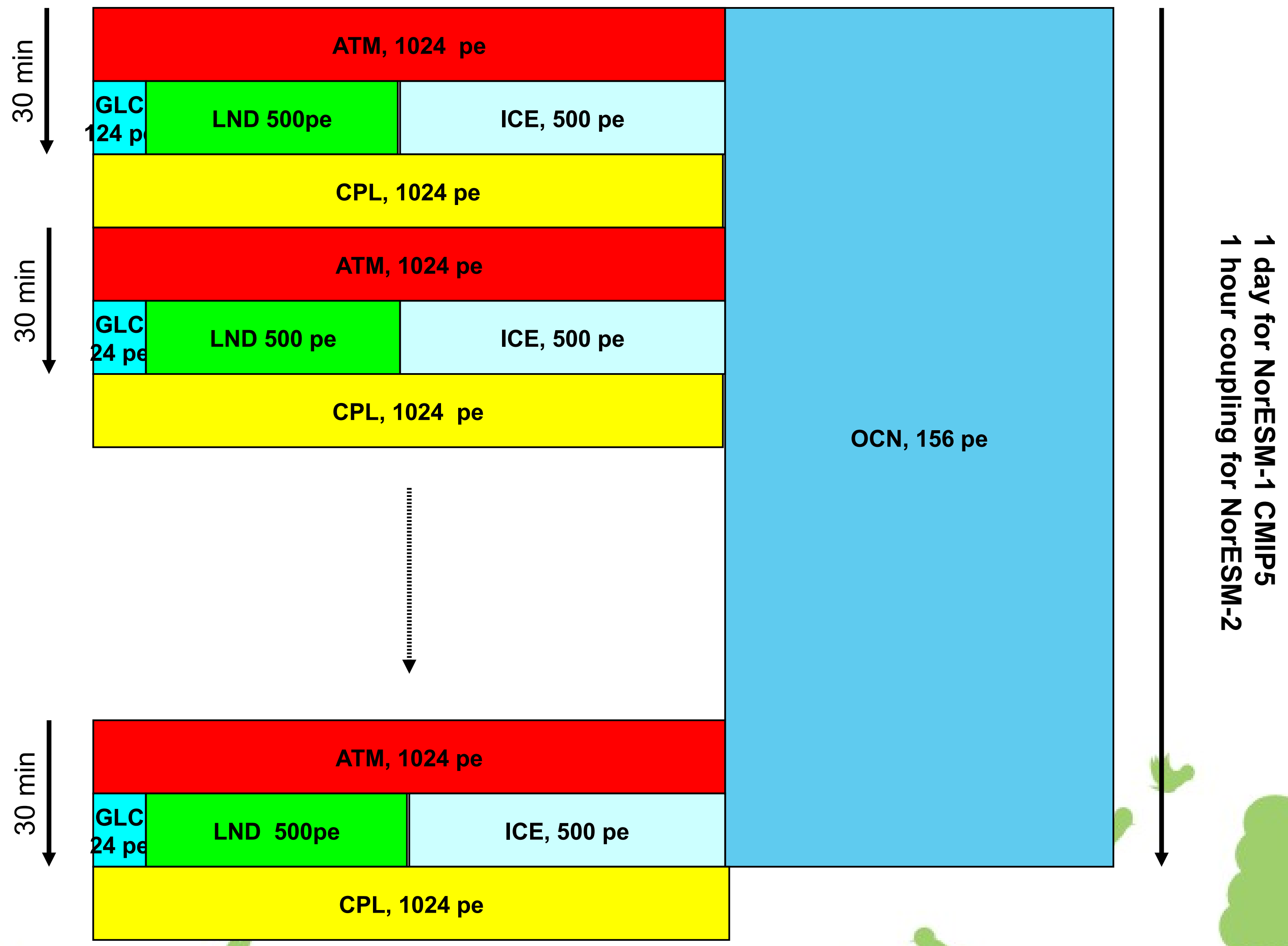
  git checkout tags/release-noresm2.0.2 -b noresm
- checkout tag **release-noresm2.0.2** to **noresm**
- **release-noresm2.0.2 –** version to reproduce CMIP6 results of NorESM2

  ./manage_externals/checkout_externals
- Also check once **External.cfg** as it specifies which tag will be checkedout
- **checkout_externals** will clone whole repository and checkout tags on path specified in **External.cfg**

NorESM framework



NorESM/CESM execution strategy



NorESM structure

cime

Most general scripts for creating, submitting, machines related setting, IO code (PIO), data components, coupler code, archiving scripts

doc

NorESM specific information
CMIP6 related key experiments
....

cime_config

Coupled experiments configurations and processors settings

components

All active components and their default namelist settings , processors and experiments setting when not fully coupled

cime

- ***scripts***

  create_newcase, **create_clone**, **query_config**

- ***scripts/Tools*:** mostly tools for setting, building and executing experiment, checks inputdata, archive data; some of these tools/scripts are linked to experiment directory when you create an experiment

- ***scripts/lib/CIME*:** contains mostly python scripts to execute these tools

cime/src/

- ***components***
  - ***data_comps*** - Data driven components
  - ***stub_comps*** - Stub components
  - ***xcpl_comps*** - Dead components- only for technical system testing
- ***drivers***
  - ***mct*** – model coupling tootkit
- ***externals***
  - ***pio*** – parallel IO code
- ***share***

  This module exists to collect code shared between various components

https://escomp.github.io/CESM/release-cesm2/cesm_configurations.html

components

- ***cam*** – atmosphere; ***cice*** – sea-ice; ***cism*** – ice sheet; ***clm*** – land ; **blom, *pop*** – ocean; ***mosart, rtm*** – river transport ; ***ww3*** – wave model

  Every component have ***src*** and ***cime_config*** directories

- ***cime_config***:

  buildlib – build particular component library

  buildnml – build particular component namelist

  config_pes.xml – processors configuration for particular compsets

  config_compsets.xml – compsets with some data components

Steps to execute an experiment

- Create an experiment – *create_newcase*
- Configure an experiment – *case.setup*
- Build an experiment – *case.build*
- *Set WALLTIME and SIMULATION RUN TIME in env_batch.xml and env_run.xml*
- Run an experiment – *case.submit*

Creating an experiment

Create an experiment using **create_newcase** *;* usually I create a directory *cases* where I want to keep experiments.
noresm/cime/scripts/create_newcase --help
Will provide you all the options;

../noresm/cime/scripts/create_newcase --case N1850OCBDRDDMS_f09_tn14 --compset N1850OCBDRDDMS --res f09_tn14 --machine fram --project nn2345k --run-unsupported

-- *case* name of experiment
-- compset *Short name of compsets*
-- *res   grid resolution*
-- *machine name of machine*
-- *project CPU hours account*
--run-unsupported scientifically unsupported resolution with compset

Compsets

- An experiment with some sets of components and forcing
- List of all compsets

  ./query_config --compsets list all existing compsets;

  ./query_config --compsets allactive

  ./query_config --compsets blom
- All compsets starting with N are NorESM related compsets

- The compset longname has the specified order atm, lnd, ice, ocn, river, glc wave cesm-options
- The notation for the compset longname is
- ***TIME_ATM[%phys]_LND[%phys]_ICE[%phys]_OCN[%phys]_ROF[%phys]_GLC[%phys]_WAV[%phys][_ESP%phys][_BGC%phys]***
- Where for the specific compsets below the following is supported
- TIME = Time period (e.g. 2000, HIST, RCP8...)
- ATM = [CAM40, CAM50, CAM54, CAM60] ;   LND = [CLM45, CLM50, SLND] ;   ICE = [CICE, DICE, SICE]
- OCN = [DOCN, ,AQUAP, SOCN, BLOM]; ROF = [RTM, MOSART, SROF] ; GLC = [CISM1, CISM2, SGLC]
- WAV = [WW3, DWAV, XWAV, SWAV];   ESP = [SESP] ; BGC = optional BGC scenario
- The OPTIONAL %phys attributes specify submodes of the given system
- For example DOCN%DOM is the data ocean model for DOCN
- ALL the possible %phys choices for each component are listed.
- ALL data models must have a %phys option that corresponds to the data model mode
- Each compset node is associated with the following elements
- - lname ; alias ; support (optional description of the support level for this compset)
- Each compset node can also have the following attributes
- - grid (optional regular expression match for grid to work with the compset)

- **Few coupled compsets:**

  TIME_ATM[%phys]_LND[%phys]_ICE[%phys]_OCN[%phys]_ROF[%phys]_GLC[%phys]_WAV[%phys][_ESP%phys][_BGC%phys]

- **N1850frc2** : *1850_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS*

- ***NHISTfrc2*** *: HIST_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS*

- ***NSSP126frc2*** *: SSP126_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS*

- ***N1850frc2 -short name of compset***

Defining compsets (experiment)/files

- Within ***components*** (not fully coupled – data atmosphere, stub atmosphere, stub or data ocean, ..)

  cam/cime_config/config_compsets.xml
  cice/cime_config/config_compsets.xml
  clm/cime_config/config_compsets.xml
  pop/cime_config/config_compsets.xml
  cism/cime_config/config_compsets.xml
  blom/cime_config/config_compsets.xml

  Fully coupled compsets:

- *cime_config/config_compsets.xml*

Grids:

- *./query_config --grids* provide description of all grids; *--long* provides deailed description.

```
alias: f19_tn14 (only for compsets that are not _POP )
non-default grids are: atm:1.9x2.5  lnd:1.9x2.5  ocnice:tnx1v4
mask is: tnx1v4

alias: T62_tn14 (only for compsets that are not _CAM )
        non-default grids are: atm:T62  lnd:T62  ocnice:tnx1v4
```

- NorESM supported grid: f19_tn14, f09_tn14, f09_tn0254, T62_tn14, TL319_tn14, TL319_tn0254, f09_f09_mg17, f19_f19_mg17...

Grid resolution

- **f19 1.9x2.5** = 1.9 degree latitude and 2.5 degree longitude resolution = 144x96
- **f09 0.9x1.25** = 0.9 degree latitude and 1.25 degree longitude resolution = 288x192
- **f02 0.23x0.31** =0.23 degree latitude and 0.31 degree longitude resolution = 1152x768
- **tnx1v1 ,tnx1v3, tnx1v4** = tripolar 1 degree grid= 360x384
- **tnx0.25v1 ,tnx0.25v3, tnx0.25v4** = tripolar quarter degree ocean grid =1440x1152 ; 1,3,4 version number

  NorESM2 is using mostly version 4 which not includes Caspian sea
- **T62, TL319** = approx. 2 degree Gaussian and 0.5 degree spectral grid used for data atmosphere

  cime/config/cesm/config_grids.xml

Mapping files

- There is different resolution for ocean, atmosphere and river runoff grid
- Coupler is used to interpolate/transfer fields from one grid to another grid
- Map files are generated for weight factors (***cime/tools***) and these factors are stored; for that purpose ESMF tool is used.
- Below some list of map files:

map_tnx1v4_to_fv0.9x1.25_aave_da_170609.nc

map_tnx1v4_to_fv1.9x2.5_aave_da_170609.nc

map_fv0.9x1.25_to_tnx1v4_aave_da_170609.nc

map_fv1.9x2.5_to_tnx1v4_aave_da_170609.nc

map_r05_to_tnx1v4_e1000r300_170609.nc

cime/config/cesm/config_grids.xml

Configuring experiment

- ./case.setup ; Creates namelists and various files and directories needed in order to build and run the case.
- Any changes to *env_mach_pes.xml* and *env_case.xml* must be made before running this.
- To run this initially for the experiment, simply run:

  ./case.setup
- To rerun after making changes to *env_mach_pes.xml*, run: **./case.setup --reset**

  ./case.setup --clean do not remove user_nl_* files and Macros file

Creating executable

- **./case.build**
- You can see all software module in *env_mach_specific.xml* and all compiler flags in *Macros.make*
- Processors configuration in *env_mach_pes.xml*
- ./case.build – it will create namelist files and compile all the required libraries (mct, gptl, csm_share and pio) and components (cam, blom, clm, cice, ..). Finally, build the binary **cesm.exe;**

  found in *$CIME_OUTPUT_ROOT/$CASE/bld*
- After this, you can modify only *env_batch.xml; env_run.xml* files and *user_nl_** files
- ./case.build --clean component removes object files of components
- ./case.build --clean-all removes bld directory

Setting simulation period and restart files option

- Modify *env_run.xml* in experiment directory
- *STOP_OPTION*

  nseconds,nsecond,nminutes,nminute,nhours,nhour,nmonths, nyears
- *STOP_N*

  numerical value

  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

Setting wallclock time

- Experiment directory *env_batch.xml*

  XML block for *case.run*

  in that set *JOB_WALLCLOCK_TIME*
- XML block for *case.st_archive*

  in that also set *JOB_WALLCLOCK_TIME*
- Here also you can modify *project* for CPU hours if required

  usually, it is set during experiment creation

Executing experiment

- **./case.submit**
- It will first update all the *namelist* if/or made any corrections after in *env_run.xml, env_batch.xml* and *user_nl_**
- Then, It will submit two jobs scripts ***.case.run*** and ***case.st_archive***

  .case.run – executes the model

  case.st_archive – is a dependent job on ***.case.run.*** On successful completion of ***.case.run*****,** it is executed and copy output to short term archiving
- Before calling it check all namelist parameters and job WALLCLOCK time

Checking inputdata

- ./check_input_data scripts check missing inputdata during ./case.build and ./case.submit
- It downloads missing data from NorESM/CESM server during ./case.submit by invoking ./check_input_data --download
- *BLOM* and *CAM-OSLO* related data exists only on *NorESM* server; if you are using older version of *NorESM* then, you have to copy these data.

Information about experiment and its status

- *README.case*

  it logs all the information regarding experiment

  which grids resolution, compsets, which NorESM repository tag and time step

- *CaseStatus*

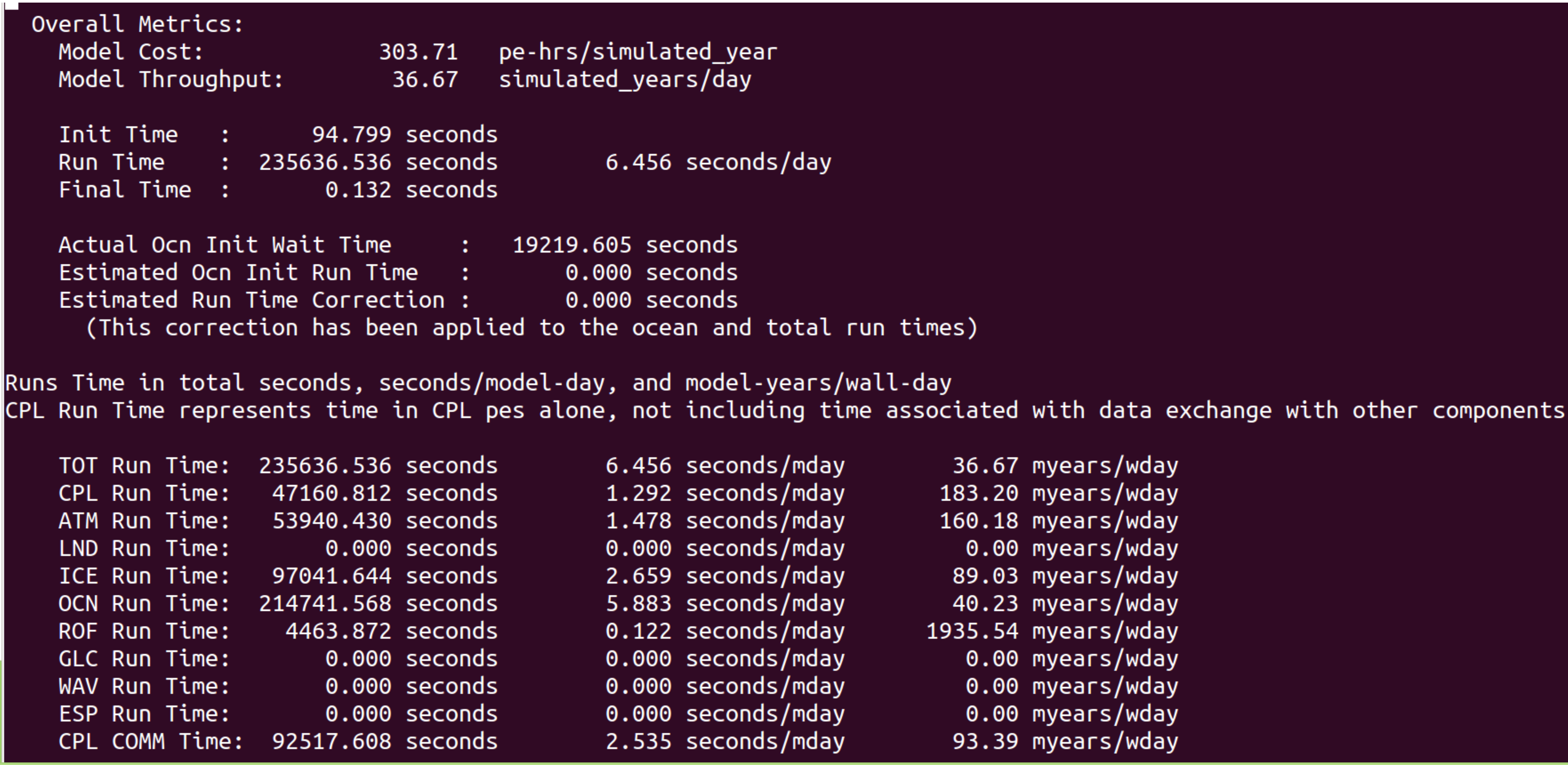
  it logs all information what you have done with experiment such as setting up experiment; building experiment ; executing experiment with time stamps

Successful completion

- Check file CaseStaus
- ---
- 2019-07-29 04:05:11: case.run starting
- ---
- 2019-07-29 04:05:26: model execution starting
- ---
- 2019-07-31 20:49:23: model execution success
- ---
- **2019-07-31 20:49:24: case.run success**
- ---
- 2019-08-01 10:14:28: st_archive starting
- ---
- 2019-08-01 10:17:12: st_archive success
- ---
- 2019-08-01 10:17:12: case.submit starting
- ---
- 2019-08-01 10:17:13: case.submit success case.run:380071.service2, case.st_archive:380072.service2
- ---
- 2019-08-02 22:00:00: case.run starting
- ---
- 2019-08-02 22:00:16: model execution starting
- ---
- 2019-08-05 02:02:45: model execution success
- ---
- **2019-08-05 02:02:45: case.run error**
- **ERROR: RUN FAIL: Command 'mpiexec_mpt -n 450 /work/agu002/noresm/NOICPL_N1850_f19_tn14**
- **_210619_sal4/bld/cesm.exe >> cesm.log.$LID 2>&1 ' failed**
- **See log file for details: /work/agu002/noresm/NOICPL_N1850_f19_tn14_210619_sal4/run/cesm.log.380071.service2.190802-220000**

Timing statistics

- Stored within *timing* sub-directory in experiment directory
- File name cesm_timing.$CASE.*
- It 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:             303.71   pe-hrs/simulated_year
    Model Throughput:        36.67   simulated_years/day

    Init Time   :      94.799 seconds
    Run Time    :  235636.536 seconds        6.456 seconds/day
    Final Time  :       0.132 seconds

    Actual Ocn Init Wait Time     :   19219.605 seconds
    Estimated Ocn Init Run Time   :       0.000 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:  235636.536 seconds        6.456 seconds/mday        36.67 myears/wday
    CPL Run Time:   47160.812 seconds        1.292 seconds/mday       183.20 myears/wday
    ATM Run Time:   53940.430 seconds        1.478 seconds/mday       160.18 myears/wday
    LND Run Time:       0.000 seconds        0.000 seconds/mday         0.00 myears/wday
    ICE Run Time:   97041.644 seconds        2.659 seconds/mday        89.03 myears/wday
    OCN Run Time:  214741.568 seconds        5.883 seconds/mday        40.23 myears/wday
    ROF Run Time:    4463.872 seconds        0.122 seconds/mday      1935.54 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:  92517.608 seconds        2.535 seconds/mday        93.39 myears/wday
```

Continuing an experiment

- *CONTINUE_RUN* in *env_run.xml;* TRUE or FALSE

  But, you need all restart files and *rpointer.** files in *run* folder

- *RESUBMIT* in *env_run.xml ;* an integer value

  it will auto resubmit till specified value; you will have total simulation period *STOP_N*(RESUBMIT+1)*

- 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*

Changing a namelist parameter

- Within experiment directory there are *user_nl_* files*
- If you plan to modify any default namelist parameter and want to use some extra namelist parameter then, you can add them to these relevant files
- For example you want to have extra cam output *user_nl_cam*

  ```
  fincl1 = 'FSN200','FSN200C','FLN200'
  ```

- http://www.cesm.ucar.edu/models/cesm1.2/cesm/doc/usersguide/x2172.html
- Example for *user_nl_clm*

  ```
  finidat = '/cluster/shared/noresm/inputdata/cesm2_init/b.e20.B1850.f09_g17.pi_control.all.297/0308-01-01/b.e20.B1850.f09_g17.pi_control.all.297.clm2.r.0308-01-01-00000.nc'

  use_init_interp = .true.

  reset_snow = .true.
  ```

- Best practice for tuning parameter

Changing source code

- Within experiment, there is a directory name ***SourceMods***; inside that there are sub-directories like as below; example from one data atmosphere experiment

  src.cice src.datm src.drof src.drv src.blom src.sglc src.share src.slnd src.swav

- BLOM will use f77 style header files (*.h) and if modified, then one must put all dependent source files to ***SourceMods/src.blom***
- put relevant source file to relevant model sub-directory and build the model again using ./case.build and then, submit
- Once satisfied, you can commit it to repository
- Best practice for modifying code

Namelist and SourceMods

- *--user-mods-dir* option in *create_newcase*

  --user-mods-dir USER_MODS_DIR

  USER_MODS_DIR name of directory which contains *user_nl_** and *SourceMods*
- If used it will copy whole structure to experiment.
- Usually, used when you want some typical settings for particular experiments.
- Exists under *cime_config/usermods_dirs*

  components/cime_config/usermods_dirs

xmlquery and xmlchange

- xmlquery :- provides the information and it is value which are sets in *.xml files
- xmlchange :- is used to change of values/parameters set in *.xml files
- xmlquery --help xmlchange --help
- Examples:

```
./xmlchange --id STOP_OPTION  --val nyears

./xmlquery --value STOP_OPTION

./xmlchange STOP_OPTION=nyears,STOP_N=1
```

Branch run

- http://www.cesm.ucar.edu/models/cesm1.2/cesm/doc/usersguide/x1894.html
- Mostly used for tuning experiments and investigating parameter affects
- *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

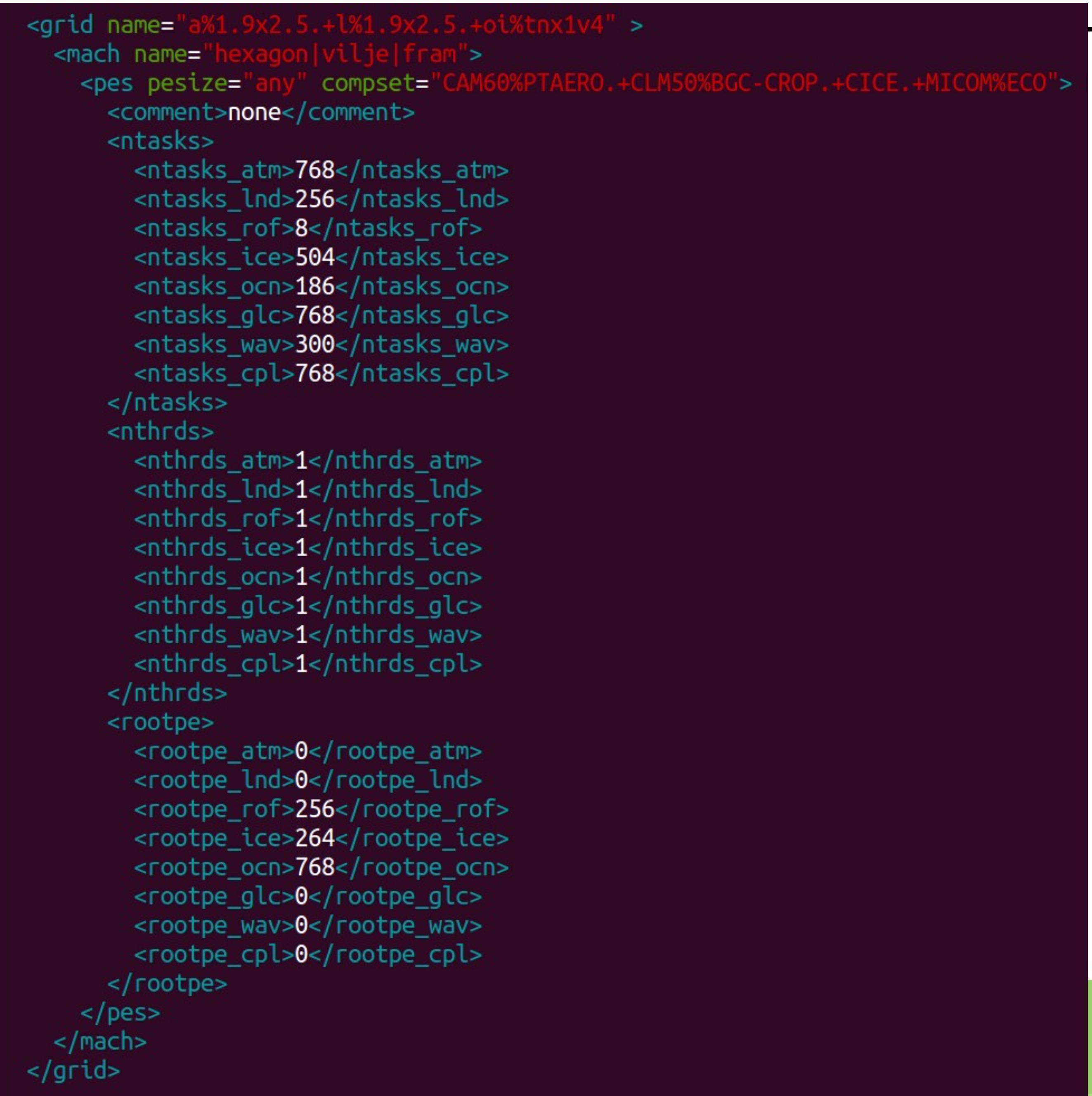
Hybrid run

- It could have reference files from many experiments
- *RUN_TYPE* to “*hybrid*”
- *RUN_REFDIR* directory containing reference data
- *RUN_REFCASE* name of reference case
- *RUN_REFDATE* Reference date for hybrid run
- *GET_REFCASE TRUE* else you have to copy data to run folder
- *RUN_STARTDATE* Run start date (yyyy-mm-dd).
- Can be used for together combinations of initial/restart files
- In a hybrid initialization, the ocean model does not start until the second ocean coupling

Cloning run

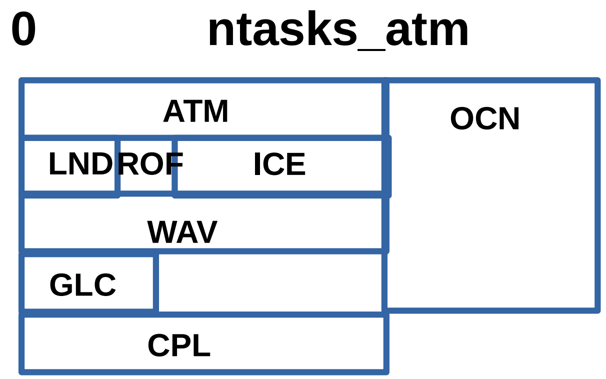
- It creates an exact copy of the experiment
- ***create_clone --case abcdef --clone old_case***
- You can do desired modifications before building and submitting executable
- You should not modify *env_case.xml, env_build.xml*
- *README.case* will be having same information as old_case; so edit it if have made some modifications or used different model version to backtrace

Setting processors count example for fully coupled code



```
<grid name="a%1.9x2.5.+l%1.9x2.5.+oi%tnx1v4" >
  <mach name="hexagon|vilje|fram">
    <pes pesize="any" compset="CAM60%PTAERO.+CLM50%BGC-CROP.+CICE.+MICOM%ECO">
      <comment>none</comment>
      <ntasks>
        <ntasks_atm>768</ntasks_atm>
        <ntasks_lnd>256</ntasks_lnd>
        <ntasks_rof>8</ntasks_rof>
        <ntasks_ice>504</ntasks_ice>
        <ntasks_ocn>186</ntasks_ocn>
        <ntasks_glc>768</ntasks_glc>
        <ntasks_wav>300</ntasks_wav>
        <ntasks_cpl>768</ntasks_cpl>
      </ntasks>
      <nthrds>
        <nthrds_atm>1</nthrds_atm>
        <nthrds_lnd>1</nthrds_lnd>
        <nthrds_rof>1</nthrds_rof>
        <nthrds_ice>1</nthrds_ice>
        <nthrds_ocn>1</nthrds_ocn>
        <nthrds_glc>1</nthrds_glc>
        <nthrds_wav>1</nthrds_wav>
        <nthrds_cpl>1</nthrds_cpl>
      </nthrds>
      <rootpe>
        <rootpe_atm>0</rootpe_atm>
        <rootpe_lnd>0</rootpe_lnd>
        <rootpe_rof>256</rootpe_rof>
        <rootpe_ice>264</rootpe_ice>
        <rootpe_ocn>768</rootpe_ocn>
        <rootpe_glc>0</rootpe_glc>
        <rootpe_wav>0</rootpe_wav>
        <rootpe_cpl>0</rootpe_cpl>
      </rootpe>
    </pes>
  </mach>
</grid>
```

Total number of processors = ntasks_atm + ntasks_ocn



rootpe_ocn=ntasks_atm
rootpe_rof=ntasks_lnd
rootpe_ice=ntasks_lnd+ntasks_rof

Processors count

- *env_mach_pes.xml*; within experiment
- *config_pes.xml*; within **cime_config** or **components/*/cime_config**
- For *atmosphere*, you can set maximum number of processors *nlat/3*atmosphere levels;* for NorESM2 32 atmosphere levels…..
- For *blom*, you have to look relevant directory
- **components/blom/bld/”gridname”**

  there are *patch.input.** files, You should use last numerical values of these files as a processors count
- *Land* is quite trivial, you can set whatever number you want
- *CICE* uses auto decomposition and you can set any processors count and based on that it will choose parallel method

Configuration files for porting to new machines

- ***cime/config/cesm***

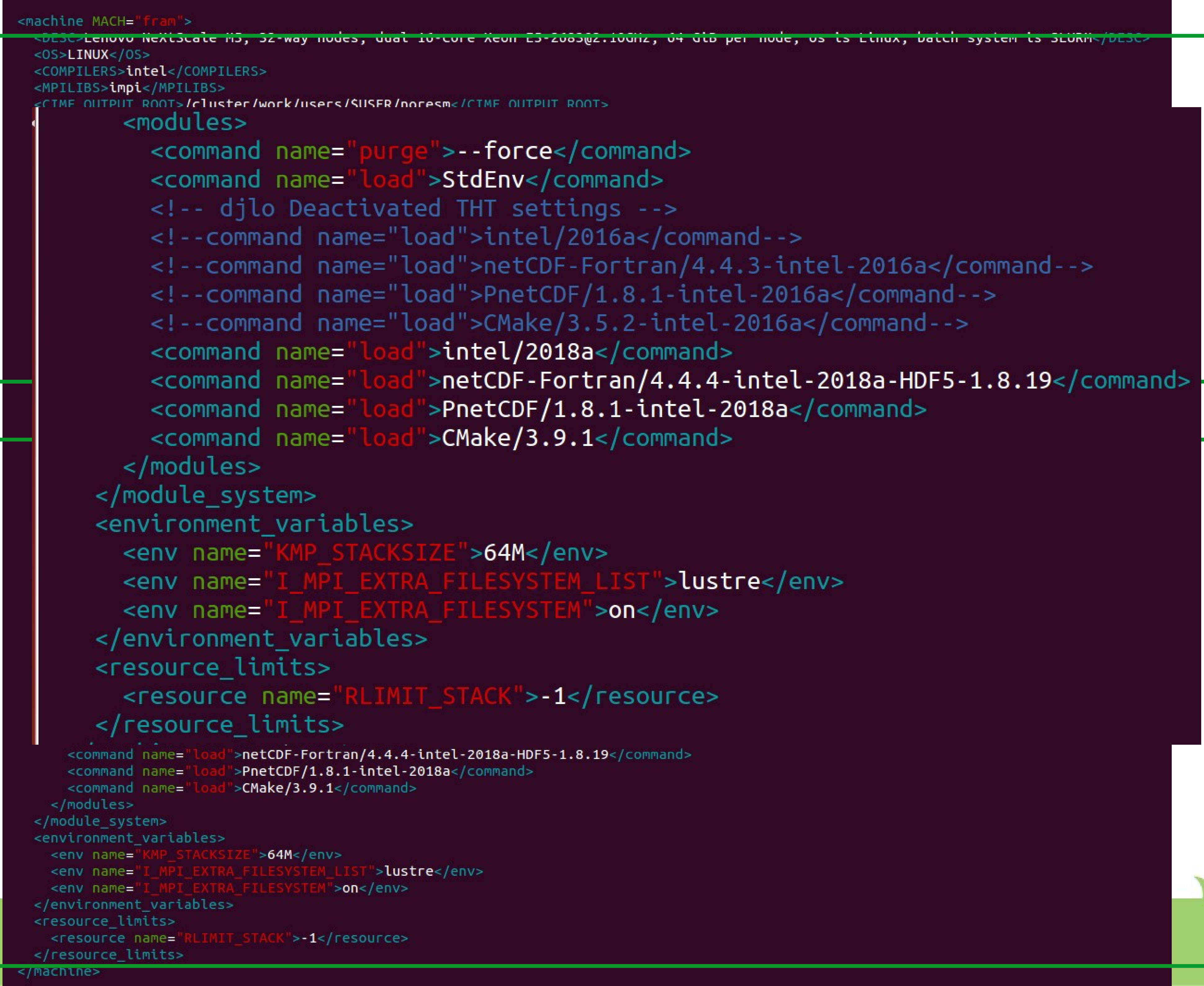
  config_machines.xml - machine related settings, where are inputdata or where to execute model or copy output, which modules to use…

  config_batch.xml - setting batch queue setting (slurm, PBS, ..)

  config_compilers.xml - compilers related settings

- ***cime_config/ components/*/cime_config***

  config_pes.xml – processors distribution settings



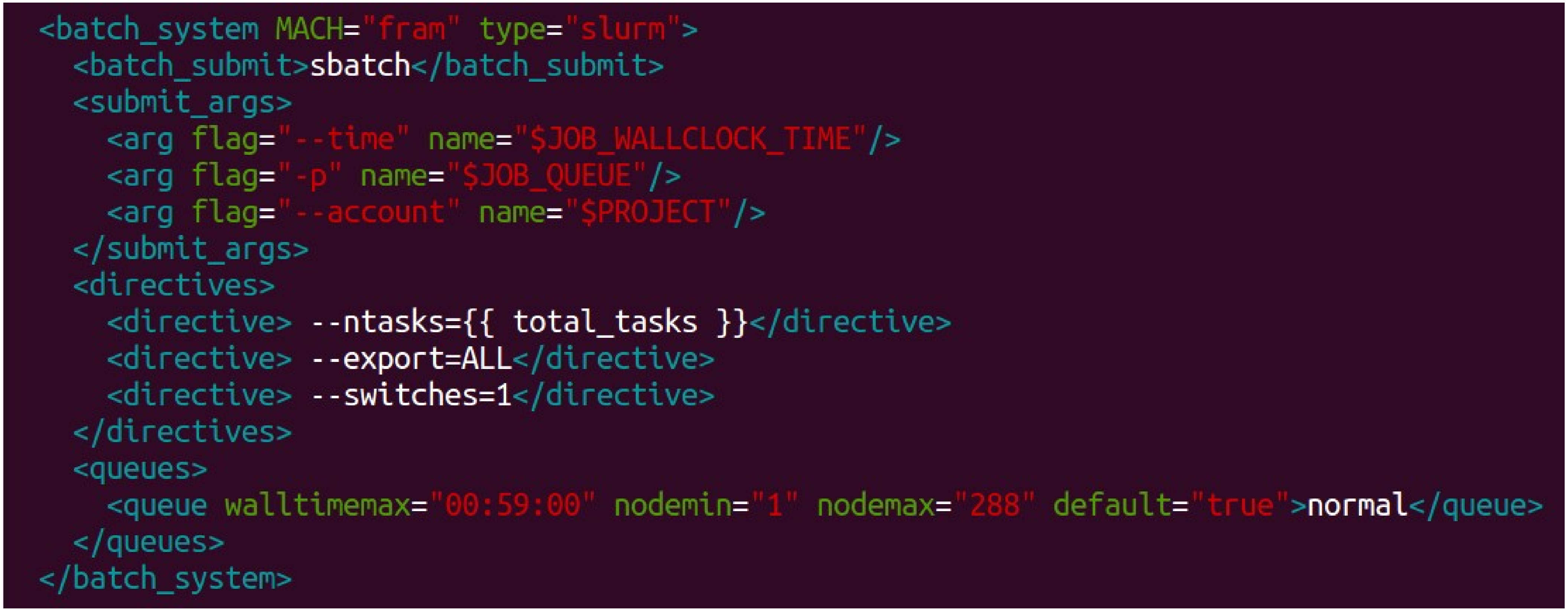
```
<machine MACH="fram">
  <DESC>Lenovo NeXtScale M5, 32-way nodes, dual 16-core Xeon E5-2683@2.10GHz, 64 GiB per node, os is Linux, batch system is SLURM</DESC>
  <OS>LINUX</OS>
  <COMPILERS>intel</COMPILERS>
  <MPILIBS>impi</MPILIBS>
  <CIME_OUTPUT_ROOT>/cluster/work/users/$USER/noresm</CIME_OUTPUT_ROOT>
```

```
    <modules>
      <command name="purge">--force</command>
      <command name="load">StdEnv</command>
      <!-- djlo Deactivated THT settings -->
      <!--command name="load">intel/2016a</command-->
      <!--command name="load">netCDF-Fortran/4.4.3-intel-2016a</command-->
      <!--command name="load">PnetCDF/1.8.1-intel-2016a</command-->
      <!--command name="load">CMake/3.5.2-intel-2016a</command-->
      <command name="load">intel/2018a</command>
      <command name="load">netCDF-Fortran/4.4.4-intel-2018a-HDF5-1.8.19</command>
      <command name="load">PnetCDF/1.8.1-intel-2018a</command>
      <command name="load">CMake/3.9.1</command>
    </modules>
  </module_system>
  <environment_variables>
    <env name="KMP_STACKSIZE">64M</env>
    <env name="I_MPI_EXTRA_FILESYSTEM_LIST">lustre</env>
    <env name="I_MPI_EXTRA_FILESYSTEM">on</env>
  </environment_variables>
  <resource_limits>
    <resource name="RLIMIT_STACK">-1</resource>
  </resource_limits>
```

```
      <command name="load">netCDF-Fortran/4.4.4-intel-2018a-HDF5-1.8.19</command>
      <command name="load">PnetCDF/1.8.1-intel-2018a</command>
      <command name="load">CMake/3.9.1</command>
    </modules>
  </module_system>
  <environment_variables>
    <env name="KMP_STACKSIZE">64M</env>
    <env name="I_MPI_EXTRA_FILESYSTEM_LIST">lustre</env>
    <env name="I_MPI_EXTRA_FILESYSTEM">on</env>
  </environment_variables>
  <resource_limits>
    <resource name="RLIMIT_STACK">-1</resource>
  </resource_limits>
```

```
</machine>
```

config_batch.xml



```xml
<batch_system MACH="fram" type="slurm">
  <batch_submit>sbatch</batch_submit>
  <submit_args>
    <arg flag="--time" name="$JOB_WALLCLOCK_TIME"/>
    <arg flag="-p" name="$JOB_QUEUE"/>
    <arg flag="--account" name="$PROJECT"/>
  </submit_args>
  <directives>
    <directive> --ntasks={{ total_tasks }}</directive>
    <directive> --export=ALL</directive>
    <directive> --switches=1</directive>
  </directives>
  <queues>
    <queue walltimemax="00:59:00" nodemin="1" nodemax="288" default="true">normal</queue>
  </queues>
</batch_system>
```

config_compilers.xml

```
<compiler MACH="fram">
  <CPPDEFS>
   <append> -D$(OS) </append>
  </CPPDEFS>
  <FFLAGS>
   <append> -xCORE-AVX2 -no-fma </append>
  </FFLAGS>
  <NETCDF_PATH>$(EBROOTNETCDFMINFORTRAN)</NETCDF_PATH>
  <PNETCDF_PATH>$(EBROOTPNETCDF)</PNETCDF_PATH>
  <MPI_PATH>$(MPI_ROOT)</MPI_PATH>
  <MPI_LIB_NAME>mpi</MPI_LIB_NAME>
  <FFLAGS>
    <append DEBUG="FALSE"> -O2 </append>
    <append MODEL="blom"> -r8 </append>
  </FFLAGS>
  <MPICC> mpiicc </MPICC>
  <MPICXX> mpiicpc </MPICXX>
  <MPIFC> mpiifort </MPIFC>
  <PIO_FILESYSTEM_HINTS>lustre</PIO_FILESYSTEM_HINTS>
  <SLIBS>
    <append>-mkl=sequential -lnetcdff -lnetcdf</append>
  </SLIBS>
</compiler>
```

Set compilers FLAGS
NETCDF and PNETCDF path
Compilers/wrapper used on machine
MODEL related flags