Logistics

- If you haven’t already done so, you may work through the setup steps (1-6) in the google document during the presentations
- If possible, open 2 docker containers in separate terminal/powershell windows with the docker run command (see step 6). One will be used for running ELM, the other for plotting.
- Please use the slack channel:

    Introduce yourself in the intro thread if you would like

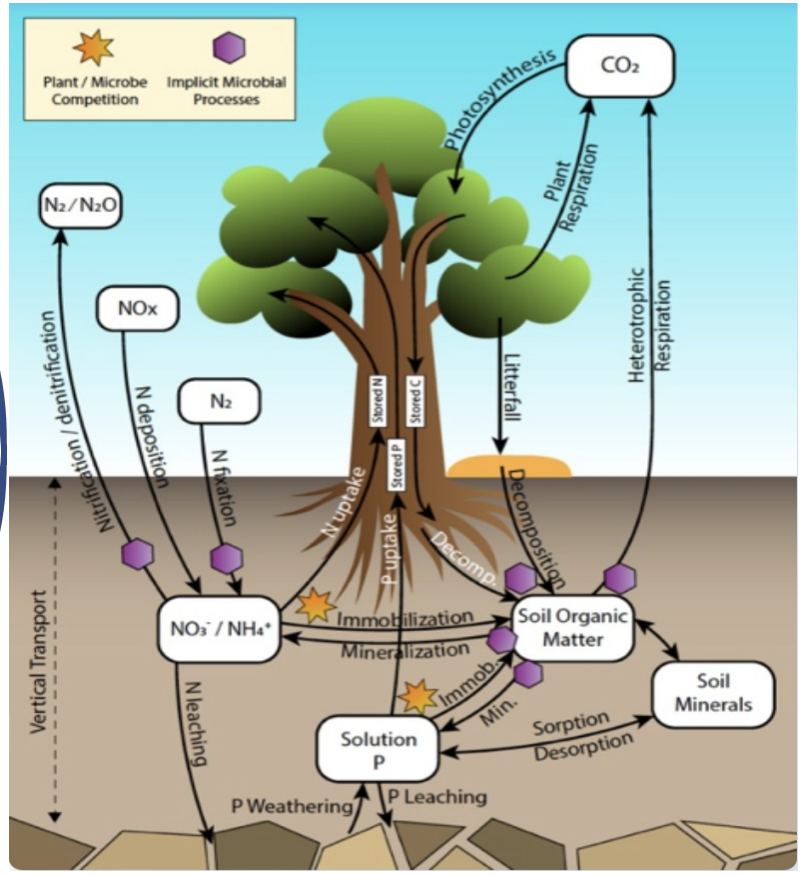
    Troubleshooting (we can help each other!)

    These are developing workflows. Helpful suggestions are very welcome!

ELM and the Offline Land Model Testbed (OLMT)

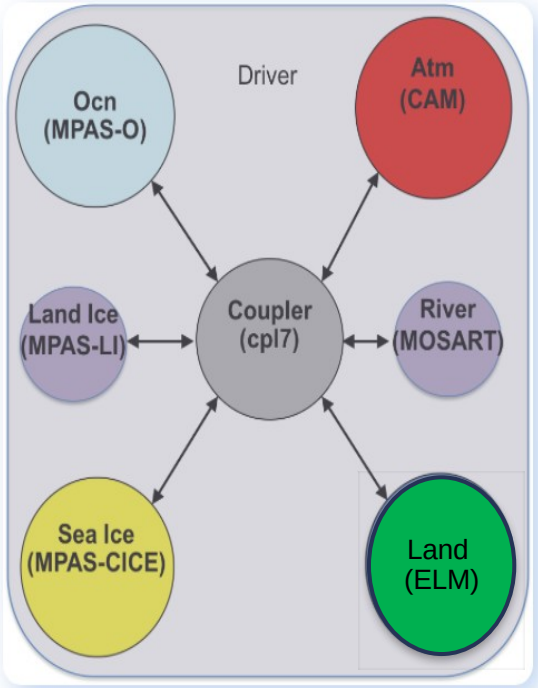
Daniel Ricciuto, Oak Ridge National Laboratory

MDI hackathon 4/13/2023

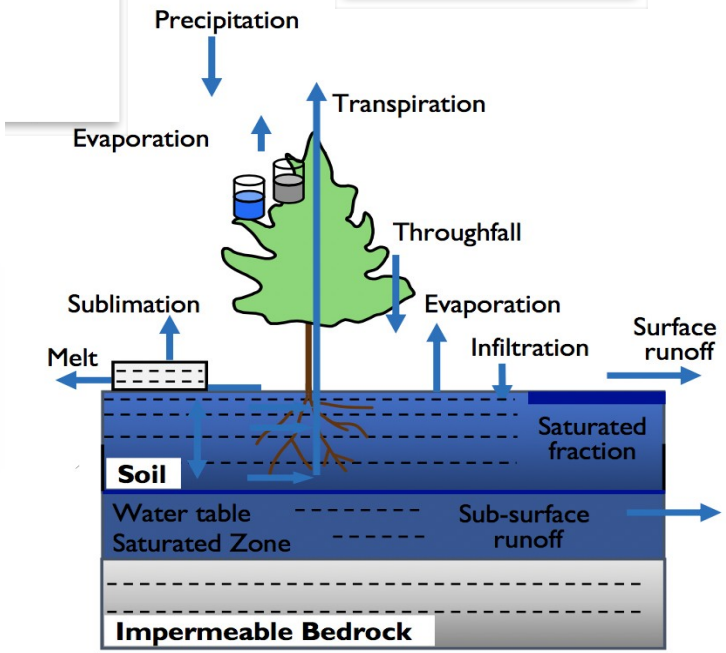


Modeling using E3SM and ELM

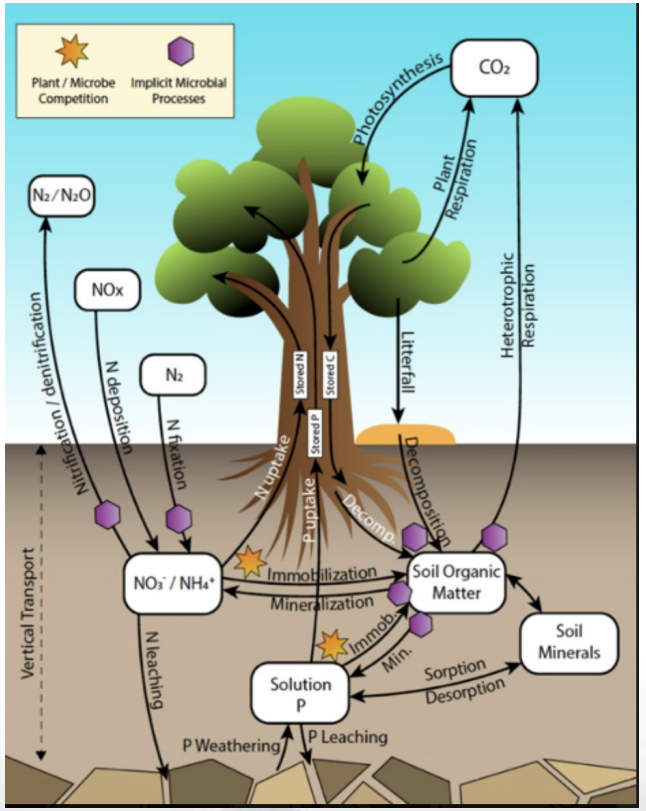
- E3SM initiated 2014; released to public April 2018
  - Version 2 coupled experiments ongoing, version 3 code to be frozen soon.
- E3SM land model (ELM) branched from CLM4.5
  P cycling, plant hydraulics, 2-way coupling with MOSART, topographic units, FATES
- DOE ESS investments, improve ELM through MODEX (e.g. NGEEs, SPRUCE, AmeriFlux, COMPASS)
- Improve model accessibility and "MODEX" workflows.



E3SM



ELM hydrology (10 soil layers (3.8m), 5 additional (35m)



ELM carbon and nutrients (N/P)

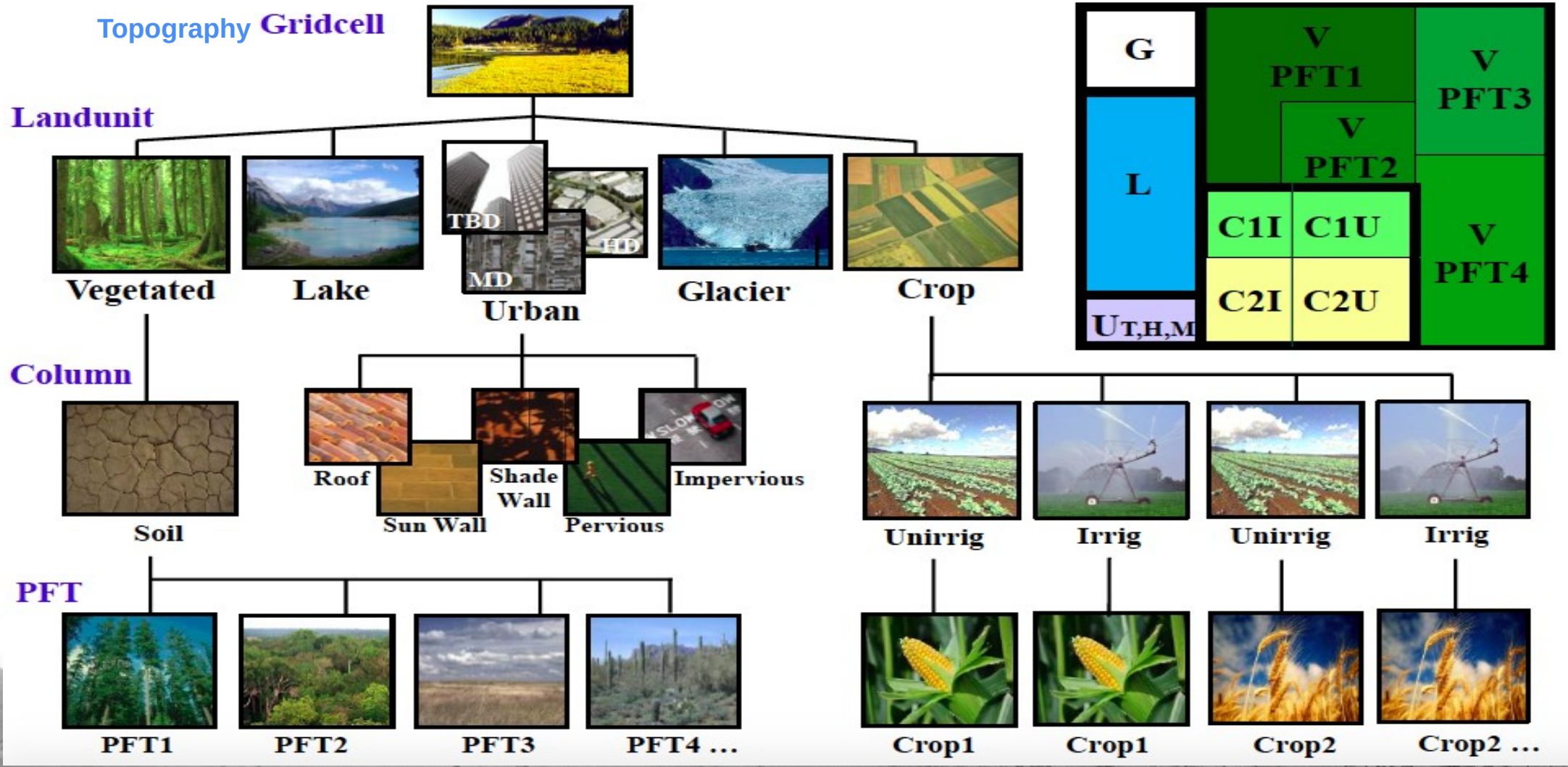
Plant pools (up to 17 PFTs):
Leaf
Live stem
Dead stem
Live Croot
Dead Croot
Fine root

Soil pools (10 layers):
SOM (4)
Litter (3)
CWD

C,N,P tracked for all of these.

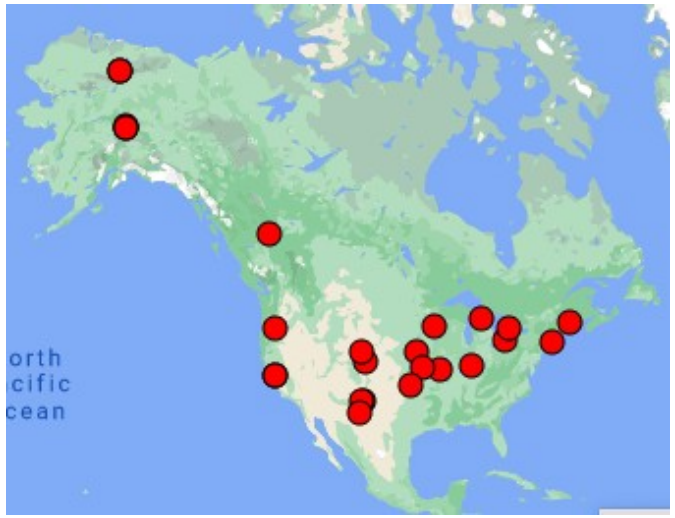
What is in an ELM gridcell?

- No lateral connectivity between gridcells.
- Resolution: 1km to 500km (25-100km most common for global simulations)
- Rectangular or unstructured grids – need not be contiguous



What does OLMT do?

- Automated framework for ELM simulations
  - ELM-SP, ELM-BGC, ELM-crop, ELM-FATES, CLM5*
  - Single site, multi-site and regional/global
  - Full BGC simulation (ad_spinup, regular spinup, transient)
  - Simulations have these **requirements**:
    - Soil texture, location and plant functional type
    - Input driver data
      - AmeriFlux/FLUXNET gap-filled drivers
      - Reanalysis (GSWP3, CRU-NCEP, Daymet, downscaled ESMs*)
    - Other inputs from global datasets (Ndep, fire inputs)
  - Scaling/manipulation of inputs
  - Multiple sites run together with a single script
  - Regional/global simulations
- Site, multi-site* ensembles/ uncertainty quantification (UQ)

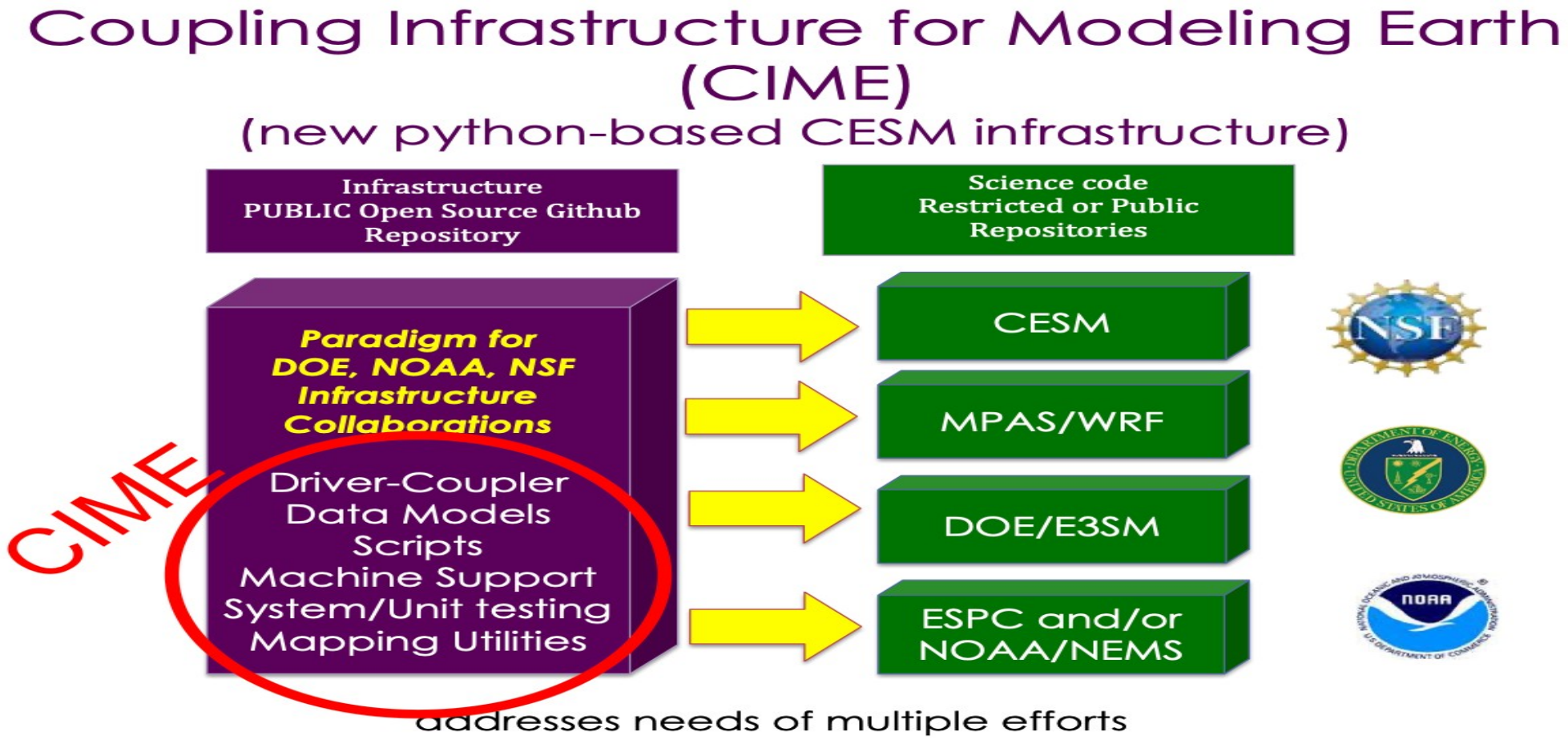


AmeriFlux sites with > 10 years of gap-filled data (ameriflux.lbl.gov). Additional data are available via FLUXNET

* - under development

E3SM simulation infrastructure

OLMT is built upon CIME:



For more info: https://www.cesm.ucar.edu/events/tutorials/2022

CIME steps: How we would run without docker/OLMT

Download the E3SM code to your machine:

- git clone -b <ref> --recursive https://github.com/E3SM-Project/E3SM.git

Navigate to the scripts directory:

> cd E3SM/cime/scripts

- ./create_newcase **--case** cmip6-picontrol **--compset** A_WCYCL1850S_CMIP6 **--res** ne30_oECv3_ICG **--mach** cori-knl

Set up the case

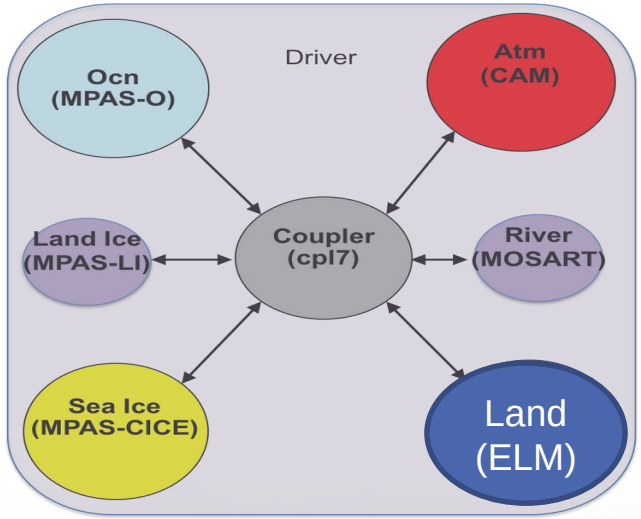
- ./case.setup

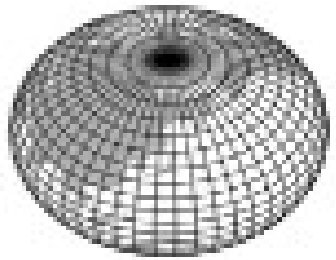
Compile the E3SM code

- ./case.build

Submit the job to the machine:

- ./case.submit





A global run is easier to launch than a site-level simulation!

https://e3sm.org/model/running-e3sm/e3sm-quick-start/

What's in a compset?

```
<!---I E3SMWCv2 compsets-->
<compset>
  <alias>I1850WCCNP</alias>
  <lname>1850_DATM%E3SMWCv2_ELM%CNPRDCTCBC_SICE_SOCN_MOSART_SGLC_SWAV</lname>
</compset>
```

time atm lnd ice ocn river lnd-ice wave

- Offline land model compset (I = offline land)
- 1850 = preindustrial forcings (CO2, Ndep, aerosol, land use); 20TR = transient
- DATM (data atmosphere) reads forcing data and inputs to land model through coupler. SATM allows reading from ELM directly ("coupler bypass"). These methods use slightly different formats.
- ELM%CNPRDCTCBC: A specific version of ELM with carbon-nitrogen-phosphorus, relative demand, converging trophic cascade, black carbon.
- SICE, SOCN,SGLC,SWAV --> stub models (inactive)
- MOSART – river routing model
- Note: Each phase of the simulation (ad_spinup, regular spinup, transient) uses a new case and compset.

Customizing a case

```
cases/b.day1.0> pwd
/glade/u/home/fischer/cases/b.day1.0
cases/b.day1.0> ls -1
archive_metadata
Buildconf
case.build
case.cmpgen_namelists
case.qstatus
case.setup
case.submit
check_case
check_input_data
env_archive.xml
env_batch.xml
env_build.xml
env_case.xml
env_mach_pes.xml
env_mach_specific.xml
env_run.xml
LockedFiles
pelayout
preview_namelists
preview_run
README.case
SourceMods
Tools
xmlchange
xmlquery
cases/b.day1.0>
```

create_newcase

script to check required input data files and download them, if necessary

User Customizable case XML files

User defined source code modifications (advanced!)

script to change XML settings

script to query XML settings

In ELM, there are a large number of potential combinations of options that are not covered by compsets. To enable them, customization is required.

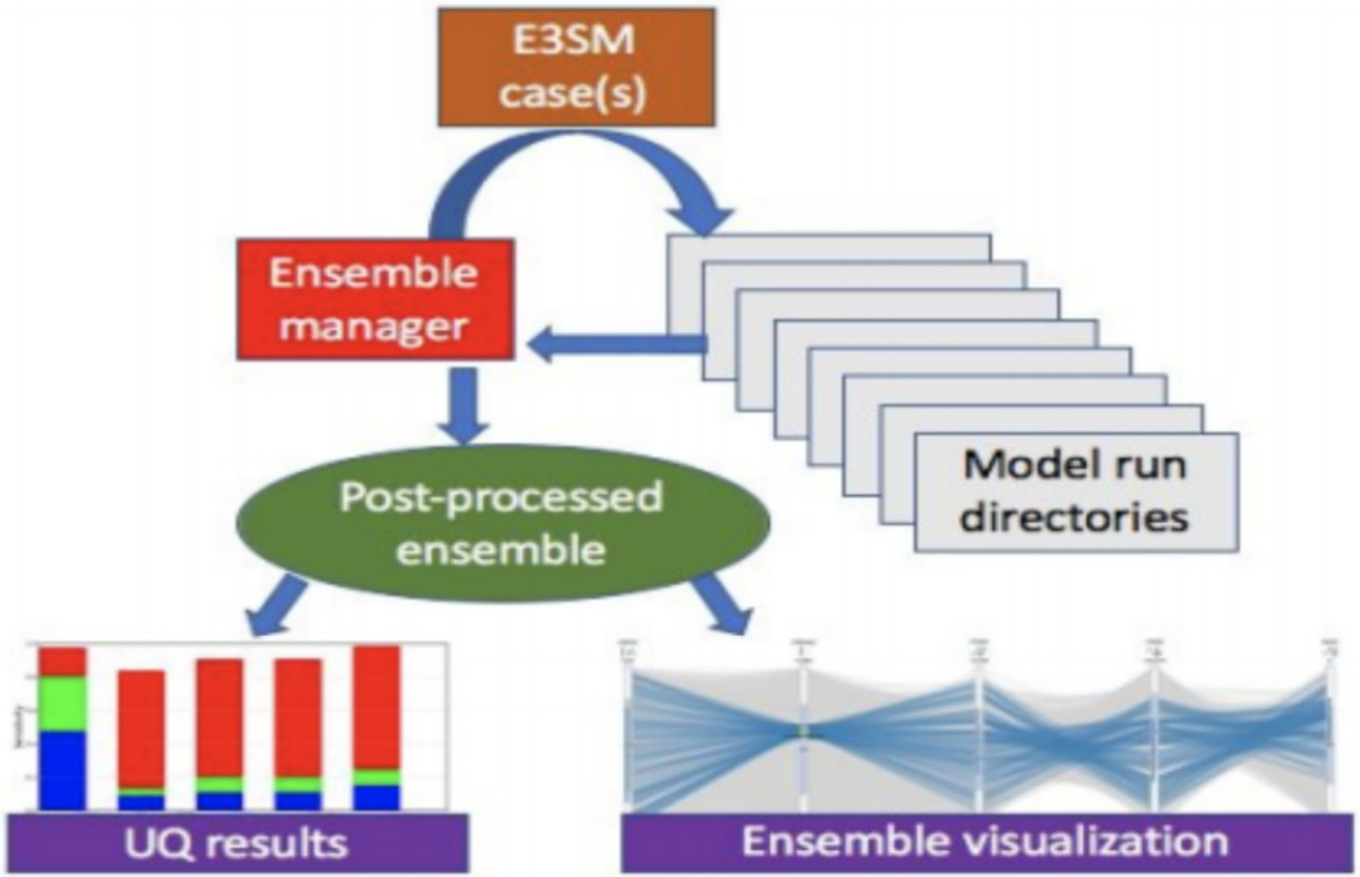
- For example, active fire, active crops, choice of decomposition model, etc.

Motivation for OLMT

OLMT automates CIME steps to enable MODEX applications in ELM

- Launching a case using a supported resolution and compset is straightforward, but customization of model options and parameters can be cumbersome and requires in-depth knowledge of CIME file locations, scripts and syntax.
- Setting up a single site run requires defining a new custom “resolution”. It is time-consuming to do this for every new site, so OLMT is configured to do this on the fly.
- Need to manage the three simulation phases (spinup + transient) and associated cases/compsets.
- Infrastructure for setting up, managing and post-processing model ensembles. This can be used for sensitivity analysis and model parameter calibration.

OLMT – ensemble and UQ capabilities



UQ capabilities

- Uses mpi4py for evaluation of serial ELM jobs (up to 1000 simultaneous simulations where parameters are varied)
  - Requirement: Prior uncertainty distribution or ranges.
- Post-processing and visualization
- Surrogate model construction
- Sensitivity analysis
- Model parameter calibration (using Markov Chain Monte Carlo. - MCMC)

Utility of the docker implementation

- Accessibility: Getting up and running on any new machine with E3SM can be challenging. The container has everything needed to run E3SM successfully on multiple platforms.
- Efficient site-level testing
- While regional ELM simulations are slow, we don’t have to worry about queue times and can quickly build and run short tests.
- Efficient analysis and visualization
- OLMT also contains some scripts for running on NERSC (Perlmutter)

Summary

OLMT enables efficient ELM simulation and UQ for sites and regions

- Simplification of steps required by CIME simulation infrastructure
- Quick and easy customization for site-level simulations, including modifying parameters and forcings
- Post-processing out ensemble model outputs for visualization and UQ
- Automated surrogate model construction for parameter sensitivity analysis and calibration.

Questions?

Resources

- E3SM
  - Github: https://github.com/E3SM-Project/E3SM
  - Project information: https://e3sm.org
  - Parameters and outputs: https://docs.google.com/spreadsheets/d/1LAzaHAkZ_mydxyohxUiV5JQ3nheV49tJ/edit#gid=1198779237
- OLMT
  - https://github.com/dmricciuto/OLMT
- FASST simulation framework
  - https://github.com/FASSt-simulation/fasst_simulation_tools