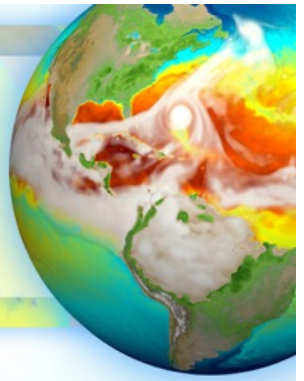
E3SM Tools

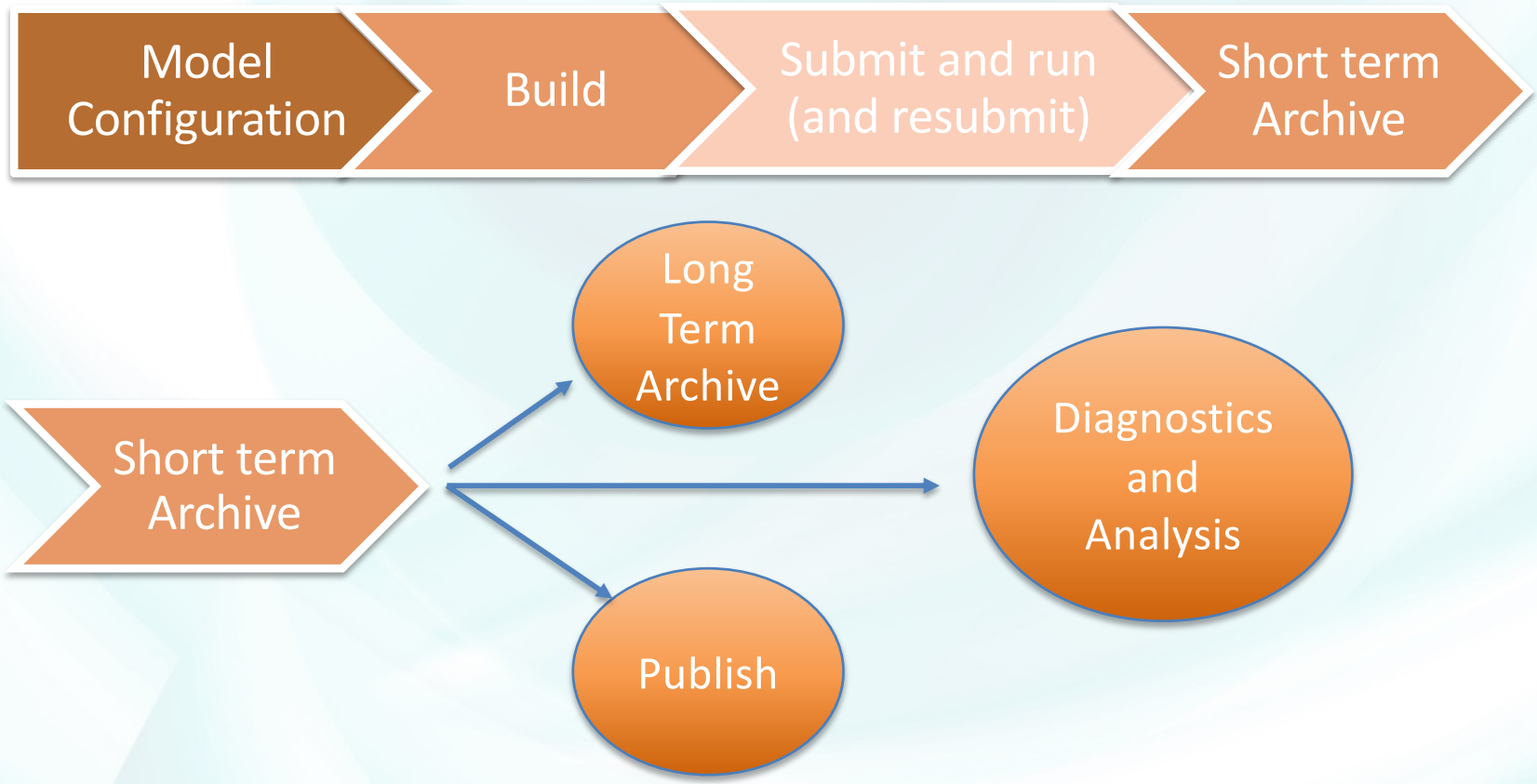


Robert Jacob
Charlie Zender
Chengzhu (Jill) Zhang
Xylar Asay-Davis
Sterling Baldwin
Ryan Forsyth
Sarat Sreepathi

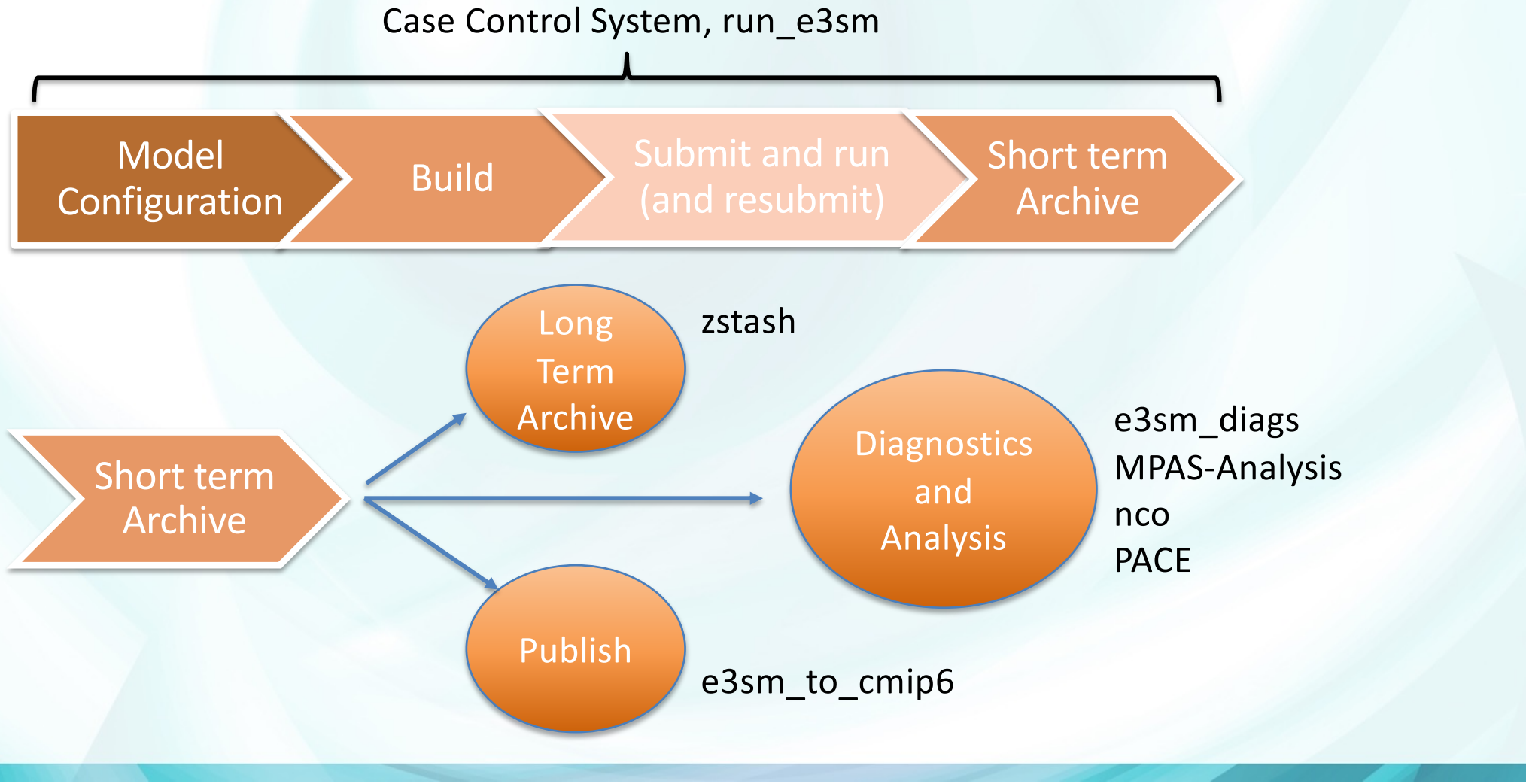
ESMD-E3SM PI Meeting

October 29, 2020

Typical ESM workflow



E3SM SFA provides tools for every phase



The CIME Case Control System

- E3SM/CESM collaborate on the Common Infrastructure for Modeling the Earth (CIME). (Now also being used by NOAA.)
- A major component of CIME is the Case Control System (CCS)
- What is the CCS?
  - CCS is the workflow control code for creating an experiment in a CIME-enabled model like E3SM (and CESM).
  - Creates and configures a case (experiment)
  - Handles access to input data and disposition of output data
  - Performs configuration and compilation of the model and included libraries (MCT, SCORPIO)
  - Abstracts the batch interface with HPC system
  - Tests of model validity
  - Written in python; XML files contain all the information required to run a given experiment on a given machine using a given compiler.

(CCS description from presentation by Jim Edwards, 2017 CESM Annual Workshop)

User Interface Scripts in CCS

In E3SM/cime/scripts:

- create_newcase: Create an experiment (case) directory for a given machine, compiler and model configuration
- query_case: list all known component sets, grids, machines. (Many more then are verified)

In each created “case directory”

- case.setup: Generate batch scripts in a case for a given pe-layout
- case.build: Compile model code and included libraries (MCT, SCORPIO)
- case.submit: Submit a job or set of jobs to the batch system
- case.st_archive: Move and sort into subdirectories output in ”run” directory to another directory (the ”short term archive”)
- xmlquery: examine case data settings in local XML files
- xmlchange: modify case data settings in local XML files

CCS Concepts

- “component set” or “compset”: A string that specifies which components\ models are combined to form the coupled model.
  - Ex: `1850_EAM%CMIP6_ELM%SPBC_MPASSI%SPUNUP_MPASO%SPUNUP_MOSART_SGLC_SWAV`
  - Shortened with a “compset alias”: A_WCYCL1850S_CMIP6
- “grid alias” describes the resolution of each component
  - Ex: `ne30pg2_r05_EC30to60E2r2-1900_ICG`
- “works out-of-the-box”: a case you can configure and run with only the basic commands (no xmlchange commands or edits to namelists)
  - Ex: ./create_newcase --case Ftest --compset FC5AV1C-L --res ne4_ne4 --mach compy
- **With a specific source code version (git hash), the compset and grid is all you need to exactly reproduce an out-of-the-box case**.
- See documentation at: http://esmci.github.io/cime

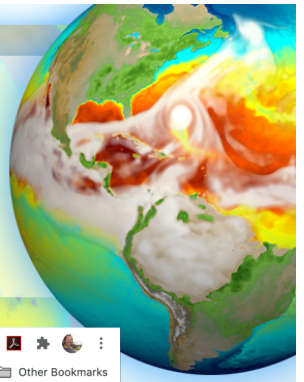
The run_e3sm script

- Normal CCS development workflow: create an out-of-the-box case and then customize it for your science
  - work in the case directory and issue xmlchange commands, edit “user_nl” files, change pe_layout, submit (for short runs) and examine results. Repeat. When ready, submit for a long production run.
  - But the resulting case directory is not portable:
    - Can’t move it to another machine and run it
    - A colleague can’t copy it and run it
    - If you want to keep that case but configure/run a modified version, have to repeat steps, copy files (user_nl)
- The run_e3sm script (written in csh) solves many of these issues
  - Put all the commands to create and modify the base case in the script.
  - Copy script to another machine or give to colleague to re-run. Commit to repo to archive.
  - Edit script and re-run to make slightly different case.
  - Specify version (git hash) of source to checkout.
  - Does a little to much now

E3SM-Unified: conda package for analysis tools.

- All of the diagnostics tools discussed here are available through a single conda package: e3sm_unified
  - On supported machines, activate with a simple command that **puts all tools in your path**:
    ```
    > ex:  source /compyfs/software/e3sm-unified
    /load_latest_e3sm_unified.sh
    ```
  - On other machines, install with “conda create” command
- Coordinates several “main” packages
  - python, **nco**, **e3sm_diags, mpas-analysis, ilamb**, **livvkit, geometric_features, mpas_tools, tempest-remap,** ncl**, cdat, zstash,** jupyter ipython, globus-cli, matplotlib, scipy, more!
  - 336 dependencies; all mutually compatible
- Available on all officially supported platforms (compy, cori, theta, summit, anvil)

NCO Quickstart: ncclimo, ncremap
Charlie Zender, UC Irvine



NCO pre+postprocesses netCDF datasets

Obtain NCO from E3SM-Unified Conda package

Use `ncclimo` to generate climos and/or timeseries from output

`ncclimo` can regrid, though `ncremap` gives more control.

Documentation…

E3SM Confluence | More | +

E3SM Documentation / Data and Analysis

Share

Regridding E3SM Data with ncremap

Created by Charlie Zender
Last updated Mar 16, 2020 • Analytics

This page is devoted to instruction in **ncremap**. It describes steps necessary to create grids, and to regrid datasets between different grids with **ncremap**. Some of the simpler regridding options supported by ncclimo are also described at Generate, Regrid, and Split Climatologies (climo files) with ncclimo. This page describes those features in more detail, and other, more boutique features often useful for custom regridding solutions.

The Zen of Regridding

Most modern climate/weather-related research requires a regridding step in its workflow. The plethora of geometric and spectral grids on which model and observational data are stored ensures that regridding is usually necessary to scientific insight, especially the focused and variable resolution studies that E3SM models conduct. Why does such a common procedure seem so complex? Because a mind-boggling number of options are required to support advanced regridding features that many users never need. To defer that complexity, this HOWTO begins with solutions to the prototypical regridding problem, without mentioning any other options. It demonstrates how to solve that problem simply, including the minimal software installation required. Once the basic regridding vocabulary has been introduced, we solve the prototype problem when one or more inputs are "missing", or need to be created. The HOWTO ends with descriptions of different regridding modes and workflows that use features customized to particular models, observational datasets, and formats. The overall organization, including TBD sections (suggest others, or vote for prioritizing, below), is:

- Software Requirements:
- Prototypical Regridding I: Use Existing Map-file
- Prototypical Regridding II: Create Map-file from Known Grid-files
- Prototypical Regridding IV: Manual Grid-file Generation
- Intermediate Regridding:
- Intermediate Regridding I: Treatment of Missing Data

Not Secure | nco.sourceforge.net/...

Apps | Getting Started | Smart Bookmarks | Getting Started | Other Bookmarks

Next: ncecat netCDF Ensemble Concatenator, Previous: ncbo netCDF Binary Operator, Up: Reference Manual [Contents][Index]

4.4 ncclimo netCDF Climatology Generator

SYNTAX

```
ncclimo [-3] [-4] [-5] [-6] [-7]
[-a dec_md] [-C clm_md] [-c caseid]
[-d dbg_lvl] [--d2f]  [--dpt_fl=dpt_fl] [-E yr_prv] [-e yr_end]
[-f fml_nm] [--fl_fmt=fl_fmt] [--glb_avg] [-h hst_nm] [-i drc_in]
[-j job_nbr] [-L dfl_lvl] [-l lnk_flg] [-m mdl_nm] [-n nco_opt]
[--no_cll_msr] [--no_frm_trm] [--no_ntv_tms] [--no_stg_grd] [--no_stdin]
[-O drc_rgr] [-o drc_out] [-p par_typ] [--ppc=ppc_prc]
[-R rgr_opt] [-r rgr_map]
[-S yr_prv] [-s yr_srt] [--seasons=csn_lst] [--sgs_frc=sgs_frc]
[-t thr_nbr] [--tpd=tpd_dly] [--uio] [-v var_lst] [--version]
[--vrt_fl=vrt_fl] [--vrt_xtr=vrt_xtr]
[-X drc_xtn] [-x drc_prv] [--xcl_var]
[-Y rgr_xtn] [-y rgr_prv] [--ypf=ypf_max]
```

DESCRIPTION

In climatology generation mode, ncclimo ingests "raw" data consisting of interannual sets of files, each containing sub-daily (diurnal), daily, monthly, or yearly averages, and from these produces climatological daily, monthly, seasonal, and/or annual means. Alternatively, in timeseries reshaping (aka "splitter") mode, ncclimo will subset and temporally split the input raw data timeseries into per-variable files spanning the entire period. ncclimo can optionally (call ncremap to) regrid all output files in either mode. Unlike the rest of NCO, ncclimo and ncremap are shell scripts, not compiled binaries[66]. As of NCO 4.9.2 (February, 2020), the ncclimo and ncremap scripts export the environment variable HDF5_USE_FILE_LOCKING with a value of FALSE. This prevents failures of these operators that can occur with some versions of the underlying HDF library that attempt to lock files on file systems that cannot, or do not, support it.

There are five (usually) required options ('-c', '-s', '-e', '-i', and '-o')) to generate climatologies, and many more options are available to customize the processing. Options are similar to ncremap options. Standard ncclimo usage for climatology generation looks like

```
ncclimo            -c caseid -s srt_yr -e end_yr -i drc_in -o drc_out
ncclimo -m mdl_nm  -c caseid -s srt_yr -e end_yr -i drc_in -o drc_out
ncclimo -v var_lst -c caseid -s srt_yr -e end_yr -i drc_in -o drc_out
ncclimo --case=caseid --start=srt_yr --end=end_yr --input=drc_in --output=drc_out
```

In climatology generation mode, ncclimo constructs the list of input filenames from the arguments to the caseid, date, and model-type options. As of NCO version 4.9.4 (September, 2020), ncclimo can produce climatologies of high-frequency input data supplied via standard input, positional command-line options, or directory contents, all input methods traditionally supported only in splitter mode. Instead of using the caseid option to help generate the input filenames as it does for normal (monthly) climos, ncclimo uses the caseid option, when provided, to rename the output files for high-frequency climos.

E3SM Energy Exascale Earth System Model

ARTMENT OF ERGY

Installation

Activate E3SM-Unified (recommended) which bundles NCO

> source <activation_path>/load_latest_e3sm_unified.sh

--OR--

Install NCO independently (e.g., for newer features)

> conda install

Climatology Generation with Regridding

 # Idealized command for EAM

```
> ncclimo -P eam --caseid=genz --var=T,Q --yr_srt=1997 \
--yr_end=2012 --drc_in=raw --drc_out=clm --map=map.nc
```

--caseid = Name of simulation
--drc_in = Input directory
--drc_out = Output directory for timeseries/climo
--map = Regridding weights from native to analysis grid
--prc_typ = -P = Processing type (model name)
--var = Variable(s) to extract
--yr_srt = Start year of timeseries/climo
--yr_end = End year of timeseries/climo

Timeseries Extraction with Regridding

 # Idealized command

```
> ls ${caseid}*.h0.nc | ncclimo --var=T --ypf=1 --yr_srt=56 \
--yr_end=76 --drc_out=clm --map=map.nc
```

--ypf = Years-per-file in output timeseries

Timeseries with Regridding: Atmosphere/EAM

 # 500-yr Atmosphere dataset:

```
> cmip6_opt='-7 --dfl_lvl=1 --no_cll_msr --no_frm_trm --no_stg_grd'
> spl_opt='--yr_srt=1 --yr_end=500 --ypf=500' # 2D
> cd ${drc_in};ls *.eam.h0.0???-*.nc | \
ncclimo --var=${vars} ${cmip6_opt} ${spl_opt} \
--map=map_ne30pg2_to_cmip6_180x360_nco.20200901.nc \
--drc_out=clm --drc_rgr=rgr
```

Timeseries with Regridding: Ocean/MPAS-O

 # 500-yr Ocean dataset:

```
> cmip6_opt='-7 --dfl_lvl=1 --no_cll_msr --no_frm_trm --no_stg_grd'
> spl_opt='--yr_srt=1 --yr_end=500 --ypf=25' # 3D
> mpas_opt='-P mpaso --d2f'
> cd ${drc_in};ls mpaso.hist.am.timeSeriesStatsMonthly.0???-*.nc | \
ncclimo --var=${vars} ${cmip6_opt} ${spl_opt} ${mpas_opt} \
--map=map_EC30to60E2r2_to_cmip6_180x360_nco.20201001.nc \
--drc_out=clm --drc_rgr=rgr
```

Timeseries with Regridding: Land/ELM

 # 500-yr Land dataset:

```
> cmip6_opt='-7 --dfl_lvl=1 --no_cll_msr --no_frm_trm --no_stg_grd'
> spl_opt='--yr_srt=1 --yr_end=500 --ypf=500'
> elm_opt='-P elm' # Applies --sgs_frc=landfrac
#elm_opt='--sgs_frc=${DATA}/grids/elm_landfrac_ne30.nc/landfrac' # Alt.
> vars='FSDS,TBOT,H2OSOI' # 2D+3D
> cd ${drc_in}; /bin/ls *.elm.h0.0???-*.nc | \
ncclimo --var=${vars} ${cmip6_opt} ${spl_opt} ${elm_opt} \
--map=map_r05_to_cmip6_180x360_nco.20200901.nc \
--drc_out=clm --drc_rgr=rgr
```

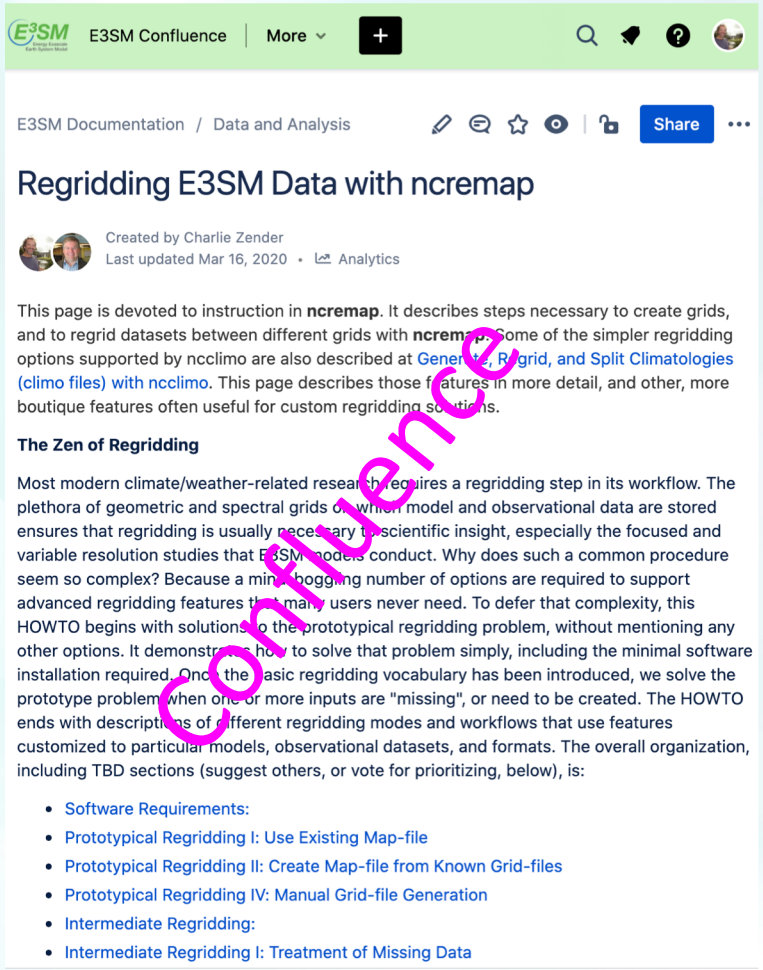
Timeseries with Regridding: Sea Ice/MPAS-SI

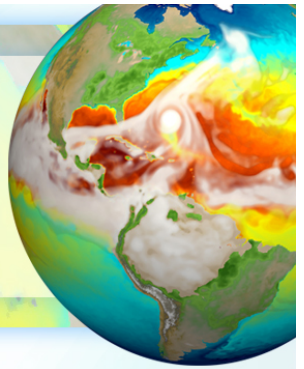
 # 165-yr Sea Ice dataset:

```
> cmip6_opt='-7 --dfl_lvl=1 --no_cll_msr --no_frm_trm --no_stg_grd'
> spl_opt='--yr_srt=1 --yr_end=165 --ypf=165' # 2D
> mpas_opt='-P mpassi --d2f' # Applies --
sgs_frc=timeMonthly_avg_iceAreaCell
> vars='timeMonthly_avg_surfaceTCell,timeMonthly_avg_iceVolumeCell'
> cd ${drc_in};ls mpascice.hist.am.timeSeriesStatsMonthly.0???-??.nc | \
ncremap --var=${vars} ${cmip6_opt} ${mpas_opt} \
--map=map_EC30to60E2r2_to_cmip6_180x360_nco.20201001.nc \
--drc_out=tmp
> ncclimo ${cmip6_opt} ${spl_opt} --drc_in=tmp --drc_out=rgr
```

Where to Find Documentation/Help

- Commands `ncremap` and `ncclimo` print their own options with examples
- Search Confluence and online manual
- Contact me





E3SM Diagnostics Package (e3sm_diags v2)

Core Development Team: Jill Chengzhu Zhang, Ryan Forsyth, Chris Golaz and Zeshawn Shaheen

Lawrence Livermore National Lab

Contributors: Xylar Asay-Davis, Charlie Zender, Sterling Baldwin

and many members from E3SM

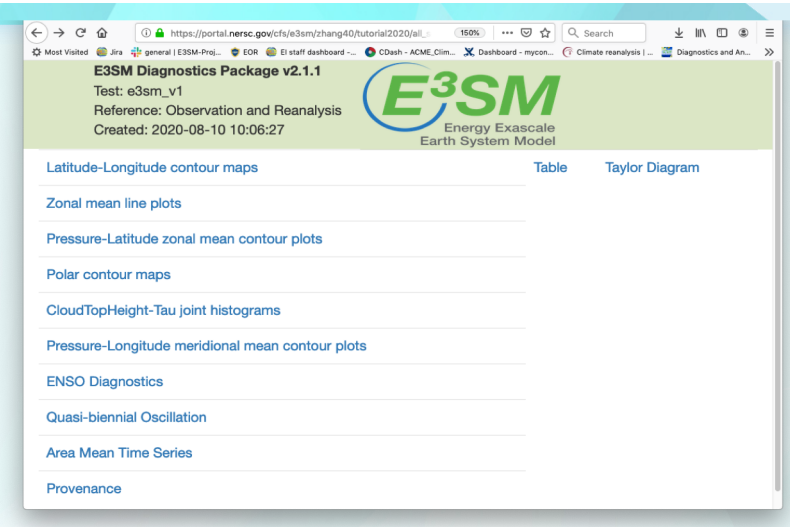
OCT 2020

E3SM Energy Exascale Earth System Model

U.S. DEPARTMENT OF ENERGY

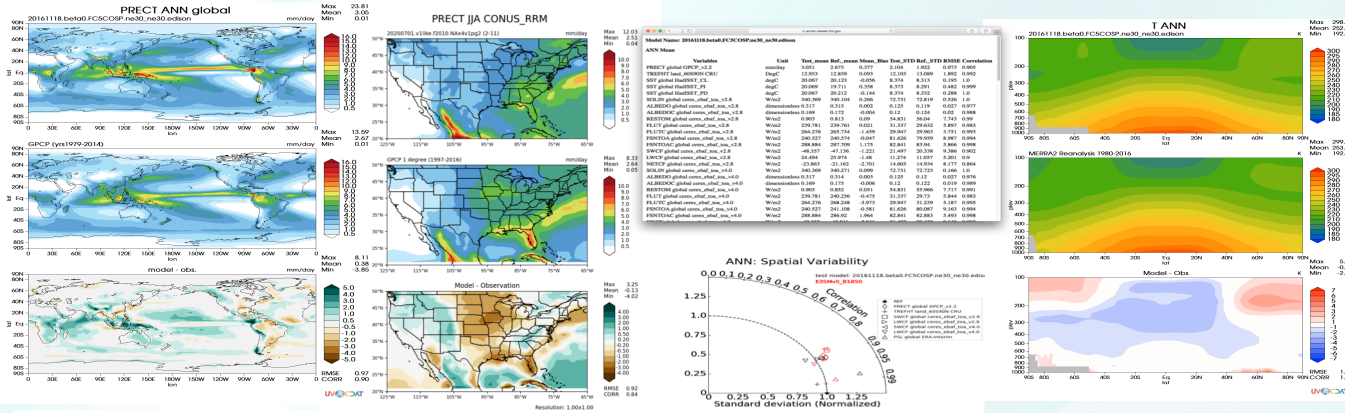
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL PRES-813617

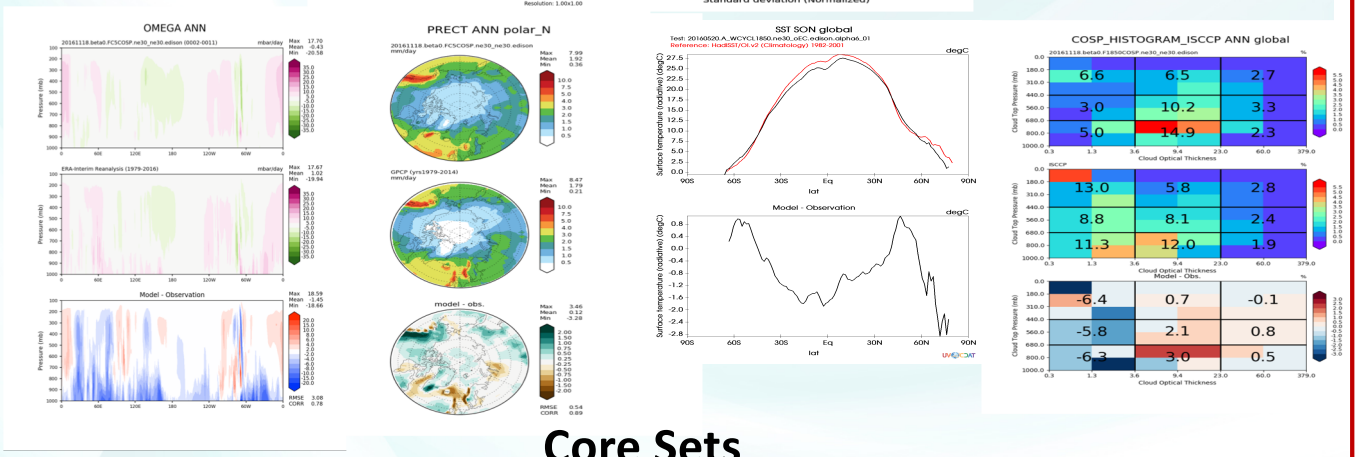
e3sm_diags



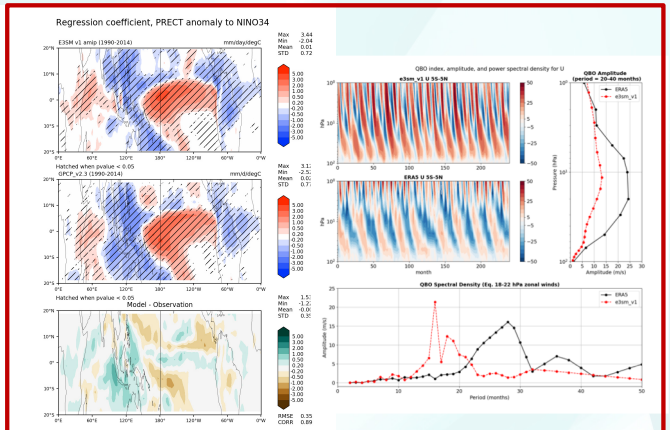
- A **modern, Python-based** diagnostics package developed for supporting E3SM model development.
- Modeled after NCAR's atmosphere diagnostics package with key sets implemented.
- Focuses on atmospheric variables. Support for land/river variables is ongoing.
- Features:
  - ✓ Flexible to add new observational datasets/diagnostics, modify figures.
  - ✓ Easy installation, configuration, and execution.
  - ✓ Runs fast using multi-processing.
  - ✓ Provenance saved for reproducing diags figures.
- Maintain an **updated** observational data repository.
- A **community tool** that accommodates CMIP convention.

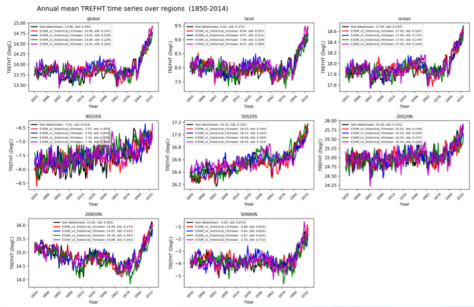
Current Available Sets





Core Sets





ENSO diags and QBO diags

Annual mean time series

Input Data Requirement

- Support data on regular latitude-longitude grids (not support raw EAM output)
  - Preprocessing through NCO to generate regrided climo and time series files

- Use seasonal climatology data as input for core set
  - ncclimo -s start_yr -e end_yr -c run_id -i drc_in -o drc_out -r map_fl **-O drc_rgr**
  - Filename: 20180215.DECKv1b_H1.ne30_oEC.edison_ANN_200001_200112_climo.nc

- Use monthly time series data as input for both core and new sets
  - # Pipe list to stdin
  - cd $drc_in;ls *cam*200[1-9]*.nc | ncclimo -v TREFHT -s 1 -e 9 -o drc_out -r map_fl **-O drc_rgr**
  - Filename: TREFHT_185001_201312.nc

  (Also support CMIP-like files: tas _185001_201312.nc or tas _185001_201312.xml)

- Example data

Installation

- On E3SM supported machines (Cori, Compy, Anvil, Cooley, Rhea)
  - **source <activation_path>/load_latest_e3sm_unified.sh**
  - Observation data and example data for testing are available on these machines

- Run on Linux or MacOS machines/ or use the latest version
  - Conda install
  - Download obs and sample model data for testing available from E3SM data server

    Obs: climatology and time-series,

    Example testing data

Configuration and Run: Core Sets

- **Run: Python `tutorial_2020_climo_sets.py`**
-

```
import os
from acme_diags.run import runner
from acme_diags.parameter.core_parameter import CoreParameter

param = CoreParameter()

param.reference_data_path = '/global/cfs/cdirs/e3sm/acme_diags/obs_for_e3sm_diags/climatology'
param.test_data_path = '/global/cfs/cdirs/e3sm/acme_diags/test_model_data_for_acme_diags/climatology/'
param.test_name = '20161118.beta0.FC5COSP.ne30_ne30.edison'
param.seasons = ["ANN","JJA"]
prefix = '/global/cfs/cdirs/e3sm/www/zhang40/tutorial2020'
param.results_dir = os.path.join(prefix, 'climo_sets')
param.multiprocessing = True
param.num_workers = 30

#Additional parameters:
#param.short_test_name = 'e3sm_v1'
#param.run_type = 'model_vs_model'
#param.diff_title = 'Difference'
#param.output_format = ['png']
#param.output_format_subplot = ['pdf']
#param.save_netcdf = True

runner.sets_to_run = ['lat_lon','zonal_mean_xy', 'zonal_mean_2d', 'polar', 'cosp_histogram', 'meridional_mean_2d']
runner.run_diags([param])
```

Run script for all sets

All available parameters

See output results

Quick Guide on Cori NERSC

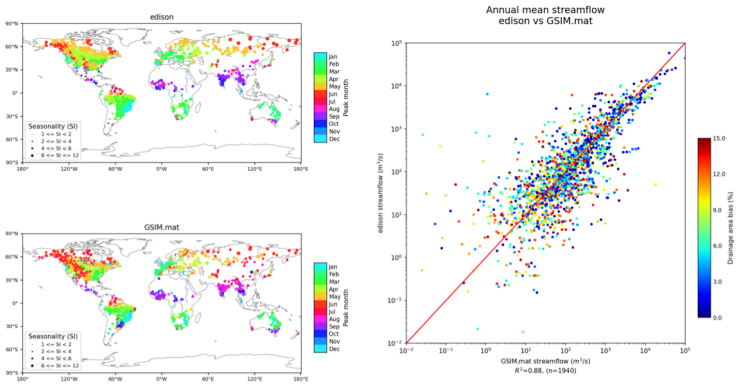
- SSH to cori
- Download tutorial examples: wget https://raw.githubusercontent.com/E3SM-Project/e3sm_diags/master/examples/tutorials/tutorial_2020_climo_sets.py
- Edit script: tutorial_2020_climo_sets.py
  - Change results_dir
- salloc --nodes=1 --partition=debug --time=00:30:00 -C haswell
- source /global/cfs/cdirs/e3sm/software/anaconda_envs/load_latest_e3sm_unified.sh

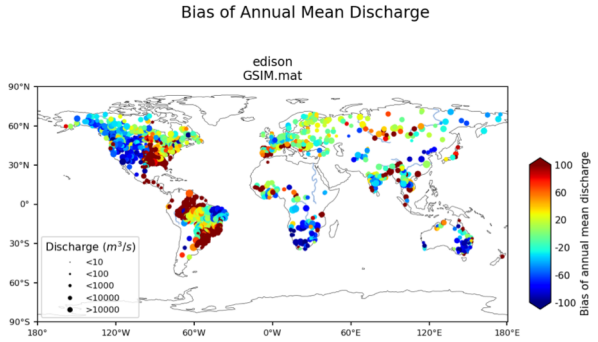
(Alternatively, conda activate e3sm_diags_env, if e3sm_diags is installed)

- **python tutorial_2020_climo_sets.py**
- View output at the web address specified

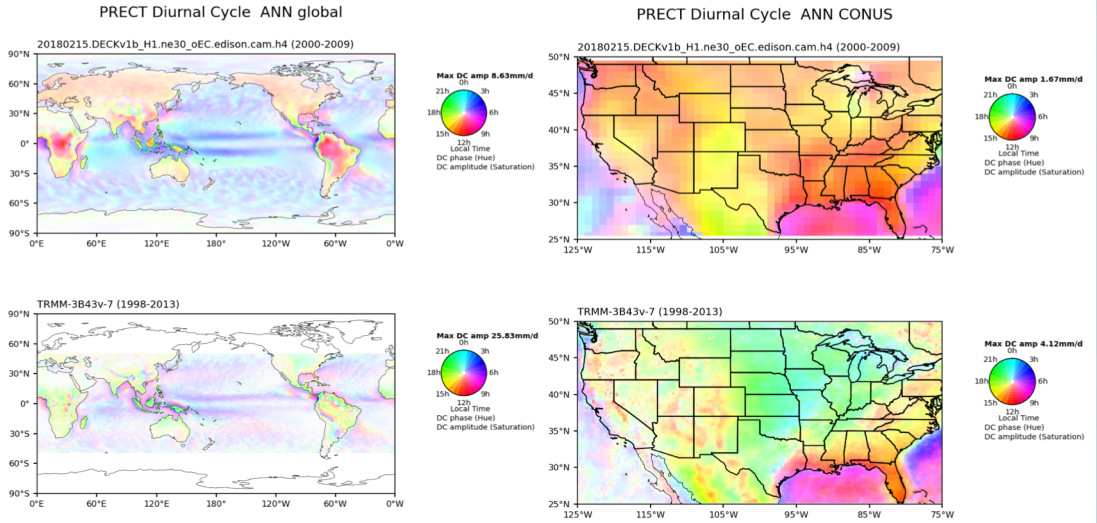
E3SM-diags new sets in next release

Streamflow diagnostics
based on GSIM gauge data





Diurnal cycle of precipitation using TRMM [0.25 deg]



<u>Ongoing and Planed</u>: ARM data-oriented diagnostics,
TC analysis, Stratospheric ozone diagnostics,
Dust aerosol, Precipitation intensity, Atmospheric CO2
diagnostics/metrics, Key land variables

How to Contribute

- Feature requests via Confluence or GitHub.
- Share the data sets and Python-based script (including instructions on data pre-processing)
- Developer's guide on how to add new diagnostics set.
- We will help with providing skeleton codes and provide infrastructure help.
- Final touch-up: linking viewers, code structure re-org, testing etc.

Thank you!

Please try it out and give us your feedback ☺
GitHub: https://github.com/E3SM-Project/e3sm_diags
Documentation on quick guide and more examples:
https://e3sm-project.github.io/e3sm_diags/docs/html/index.html

An Introduction to MPAS-Analysis

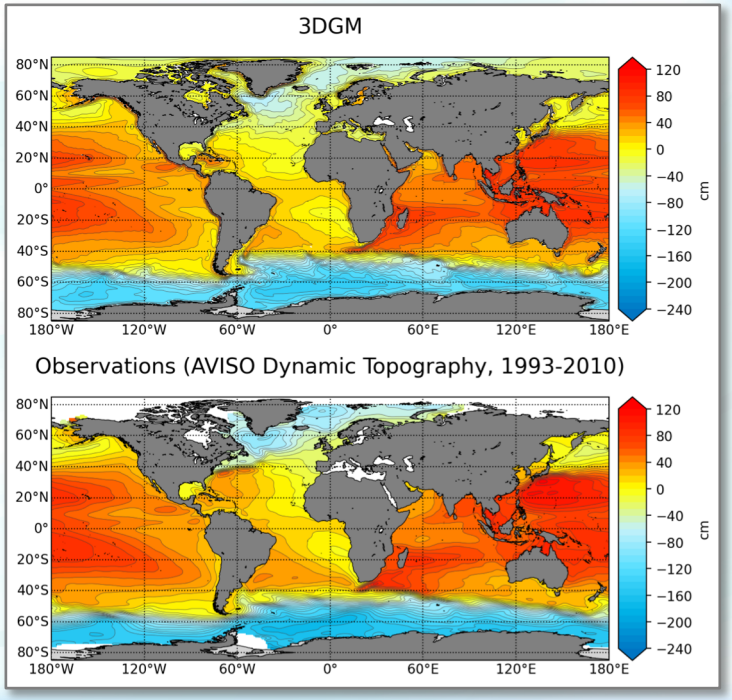
Main developers:

Xylar Asay-Davis

Milena Veneziani

Contributors:

Sterling Baldwin
Riley X. Brady
Darin Comeau
Charles Doutriaux
Jeremy Fyke
Matthew Hoffman
Joseph Kennedy
Mark Petersen
Stephen Price
Kevin Rosa
Greg Streletz
Adrian Turner
Luke Van Roekel
Phillip J. Wolfram



Sea surface height

E3SM Energy Exascale Earth System Model

U.S. DEPARTMENT OF ENERGY

Installing MPAS-Analysis

- Details:
  - MPAS-Analysis Documentation
  - MPAS-Analysis Tutorial
- The gist:
  - Install Miniconda3

```
conda config --add channels conda-forge
conda create -n mpas-analysis python=3.8 mpas-analysis
conda activate mpas-analysis
```

  - Download
    - observations
    - mapping and mask files for standard meshes

```
download_analysis_data -o /path/to/mpas_analysis/diagnostics
```

  - Already installed on supported machines through e3sm_unified.

E3SM Results for Input

- E3SM simulation directory:

```
baseDirectory = /global/cscratch1/sd/sprice/e3sm_scratch/cori-knl/20200610.A_WCYCL1850-DIB-ISMF_CMIP6.ne30_ECwISC30to60E1r2.cori-knl.maint1p2-3DGM
```

- A subdirectory with an MPAS-Ocean restart file:

```
runSubdirectory = run
```

- Subdirectories with ocean and sea-ice monthly averaged data:

```
oceanHistorySubdirectory = archive/ocn/hist
seaIceHistorySubdirectory = archive/ice/hist
```

- And namelists and “streams” files describing MPAS parameters and output:

```
oceanNamelistFileName = run/mpaso_in
oceanStreamsFileName = run/streams.ocean
seaIceNamelistFileName = run/mpassi_in
seaIceStreamsFileName = run/streams.seaice
```

Configuring MPAS-Analysis

- Configuration is with Python cfg (also called ini) files:

```
[runs]
# mainRunName is a name that identifies the simulation being analyzed.
mainRunName = runName

[execute]
# the number of parallel tasks (1 means tasks run in serial, the default)
parallelTaskCount = 1

# the parallelism mode in ncclimo ("serial" or "bck")
ncclimoParallelMode = serial
```

- The default config file contains over 1,000 config options
  - Lots of flexibility
  - A bit overwhelming
- Override defaults with one or more custom config files
  - We'll go over some common config options later in the presentation

Run the code

- Run:

```
$ mpas_analysis 20200610.A_WCYCL1850.ne30_ECwISC30to60E1r2.cori-knl.cfg
```

- (Better yet, run a batch job)
- Typical output:

```
Running tasks:   8% |###                                          | ETA:   0:02:34

Log files for executed tasks can be found in
/media/xylar/bbyates/analysis/output/GMPAS-QU240wLI/native_transects_mpas_tools/logs
Total setup time: 0:00:07.21
Total run time: 0:01:19.78
Generating webpage for viewing results...
Done.
```

- Copy and/or chmod the resulting web output so you can view it in a web portal

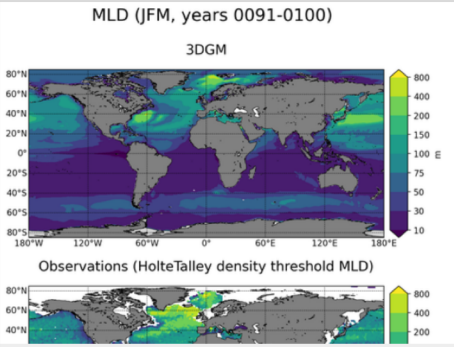
Web Interface

MPAS-Analysis Diagnostics

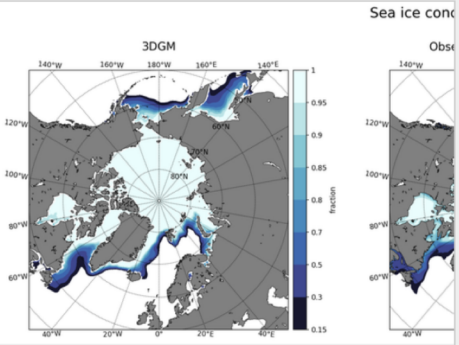
Run: 3DGM

MPAS
Model for Prediction Across Scales

Components



Ocean



Sea Ice

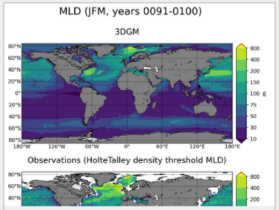
Web Interface: Ocean

MPAS-Analysis Diagnostics: Ocean

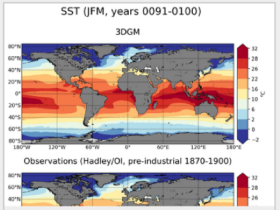
Run: 3DGM

MPAS
Model for Prediction Across Scales

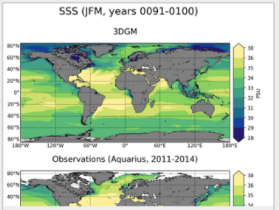
Quick Links



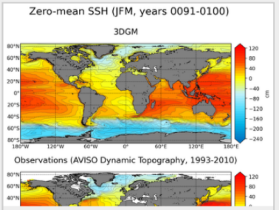
Global Mixed-Layer Depth



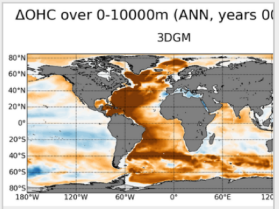
Global Sea Surface Temperature



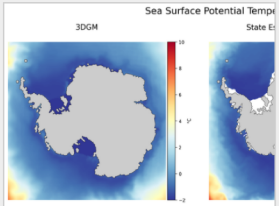
Global Sea Surface Salinity



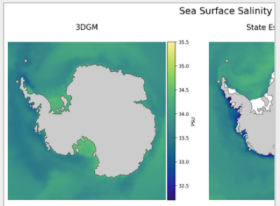
Global Sea Surface Height



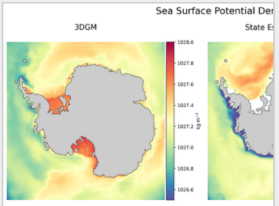
Global OHC Anomaly



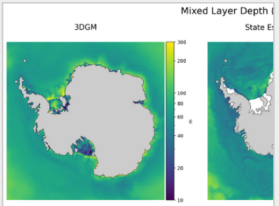
Antarctic Potential Temperature



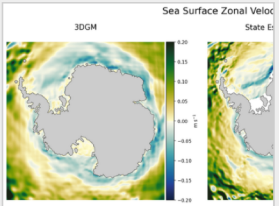
Antarctic Salinity



Antarctic Potential Density



Antarctic Mixed Layer Depth



Antarctic Zonal Velocity

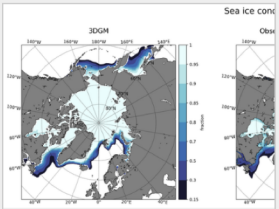
Web Interface: Sea Ice

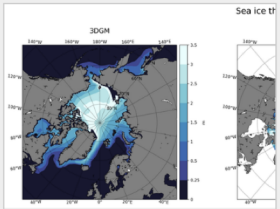
MPAS-Analysis Diagnostics: Sea Ice

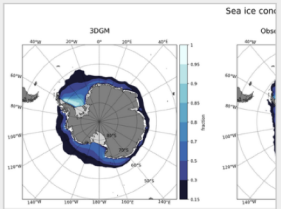
Run: 3DGM

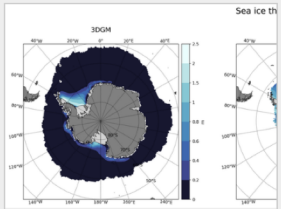
MPAS
Model for Prediction Across Scales

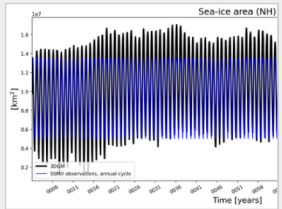
Quick Links


Northern-Hemisphere Sea-Ice Concentration


Northern-Hemisphere Sea-Ice Thickness


Southern-Hemisphere Sea-Ice Concentration

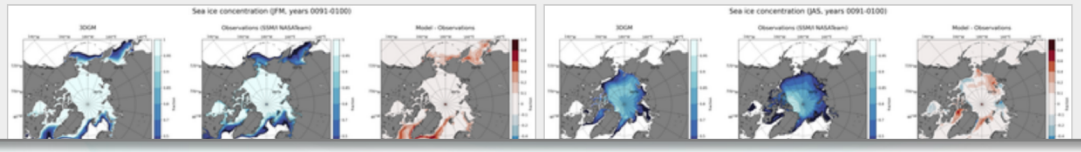

Southern-Hemisphere Sea-Ice Thickness



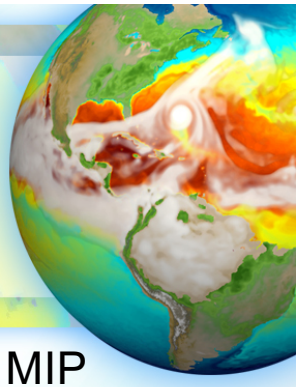
Time Series

Northern-Hemisphere Sea-Ice Concentration

Observations: SSM/I NASATeam



e3sm_to_cmip



An extensible and portable python package for conversion of E3SM data to the MIP format. Able to take atmosphere, land, sea-ice, and ocean variables as input, and converts them into CMIP compliant datasets. The package has been used to produce for more then 1800 datasets published to CMIP6.

Currently supports:

- 54 Atmosphere variables
- 24 Land variables
- 27 Ocean variables
- 9 Sea-ice variables

Atmosphere and land variables must first be regridded and extracted as time-series. Ocean and sea-ice can be converted from raw MPAS output.

New variables can be added by simply adding an entry to a YAML resource file.

E3SM Energy Exascale Earth System Model

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC. LLNL-POST-750734 Release #LLNL-PRES-815934

U.S. DEPARTMENT OF ENERGY

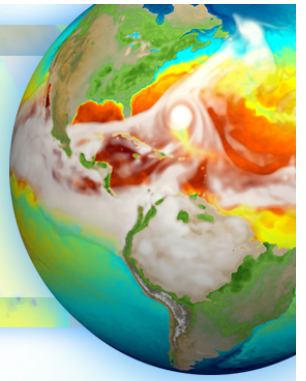
Usage and Future Work

- e3sm_to_cmip can run in both “production” and “simple” modes
  - “Production” mode is used to produce official CMIP6 datasets ready for publication and requires additional metadata.
  - “Simple” mode can be used with E3SM output from any debug run with no extra metadata needed.
- A suite of Common Workflow Language tools are provided to do the complete end-to-end conversion process, including the required horizontal regridding and vertical regridding for 3D atmosphere variables.
- Planned features:
  - Support for high-frequency output conversion
  - Simplification of variable converters

Links & resources

- Repository: https://github.com/E3SM-Project/e3sm_to_cmip/
- Getting started with CWL: https://github.com/E3SM-Project/e3sm_to_cmip/blob/master/docs/workflow.md
- Example metadata: https://raw.githubusercontent.com/E3SM-Project/e3sm_to_cmip/master/e3sm_user_config_picontrol.json

zstash v0.4.2: HPSS long-term archiving tool



Core Development Team: Ryan Forsyth, Chris Golaz, and Zeshawn Shaheen

Lawrence Livermore National Lab

October 2020

E3SM Energy Exascale Earth System Model

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. It is supported by the Energy Exascale Earth System Model (E3SM) project, funded by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research.

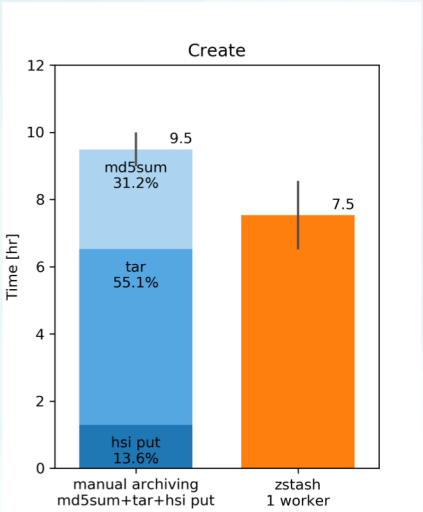
U.S. DEPARTMENT OF ENERGY

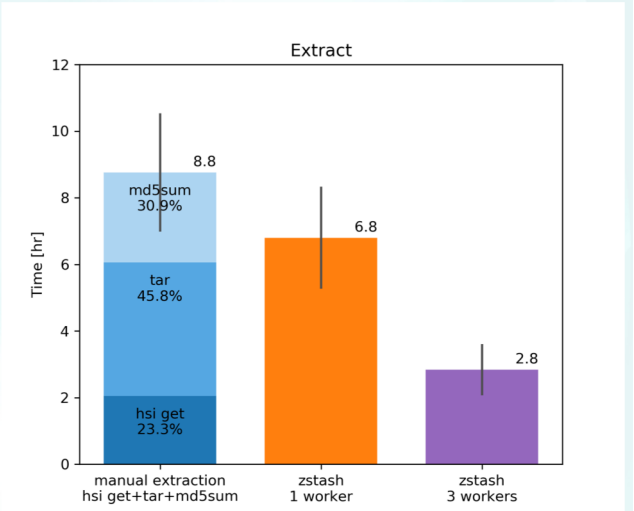
zstash: HPSS long-term archiving tool

- A Python tool for long-term HPSS archiving of E3SM simulations with features:
  - **Standard tar files** generated locally before transfer to HPSS
  - **Checksums (md5)** are computed *on-the-fly* during archiving and **verified** on extraction
  - Metadata stored in a sqlite3 index **database,** enabling **faster retrieval** for target files
  - **Parallel extraction and verification** for increased performance

  - **All E3SM v1 production simulations have been archived on NERSC HPSS using zstash**

Performance data for a 4 TB archive consisting of more than 13,000 files. Mean and range of three realizations on NERSC's Data Transfer Nodes (dtn).





Example 1: extract simulation data

- https://e3sm-project.github.io/zstash/docs/html/best_practices.html#nersc
- Say you're on NERSC, just completed a run with directory name `e3sm_output`
- We could archive that data with:
  - $ ssh dtn01.nersc.gov
  - $ screen
  - $ bash
  - $ source /global/cfs/cdirs/e3sm/software/anaconda_envs/load_latest_e3sm_unified.sh
  - $ zstash create --hpss=path/to/zstash_archive e3sm_output
- Someone else could then extract the data from the HPSS archive with:
  - $ zstash extract --hpss=path/to/zstash_archive

Example 2:archive and transfer to a machine that has HPSS

- https://e3sm-project.github.io/zstash/docs/html/best_practices.html#compy-anvil
- Compy and Anvil do not have HPSS. How can we get that data archived on NERSC HPSS?

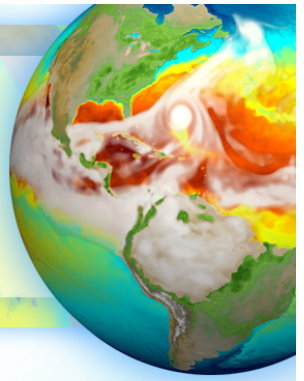
Steps:

1. Run '**zstash create**' to archive to disk on compy/anvil.
2. Using **Globus**, transfer zstash archive files (everything under the zstash/ subdirectory) to NERSC HPSS. Select the option to preserve original files modification date.
3. Run '**zstash check**' to verify integrity of zstash archive (and their transfer).

Getting Started

- On NERSC, load the E3SM unified environment which includes zstash: $ source /global/cfs/cdirs/e3sm/software/anaconda_envs/load_latest_e3sm_unified.sh
- More details: https://e3sm-project.github.io/zstash/docs/html-v0-4-2/getting_started.html
- Documentation (for `master` branch): https://e3sm-project.github.io/zstash/docs/
- Tutorial: https://e3sm-project.github.io/zstash/docs/html-v0-4-2/tutorial.html
- Source code: https://github.com/E3SM-Project/zstash
- Contact: Ryan Forsyth (forsyth2@llnl.gov), Chris Golaz (golaz1@llnl.gov)

Performance Analytics for Computational Experiments



Sarat Sreepathi

Oak Ridge National Laboratory

ESMD/E3SM Meeting

October 29th, 2020

In collaboration with

Youngsung Kim, ORNL

Special thanks to Patrick H. Worley

Past Students:

Zachary Mitchell,
Pellissippi State Community College

Gaurab KC,
University of Tennessee Knoxville

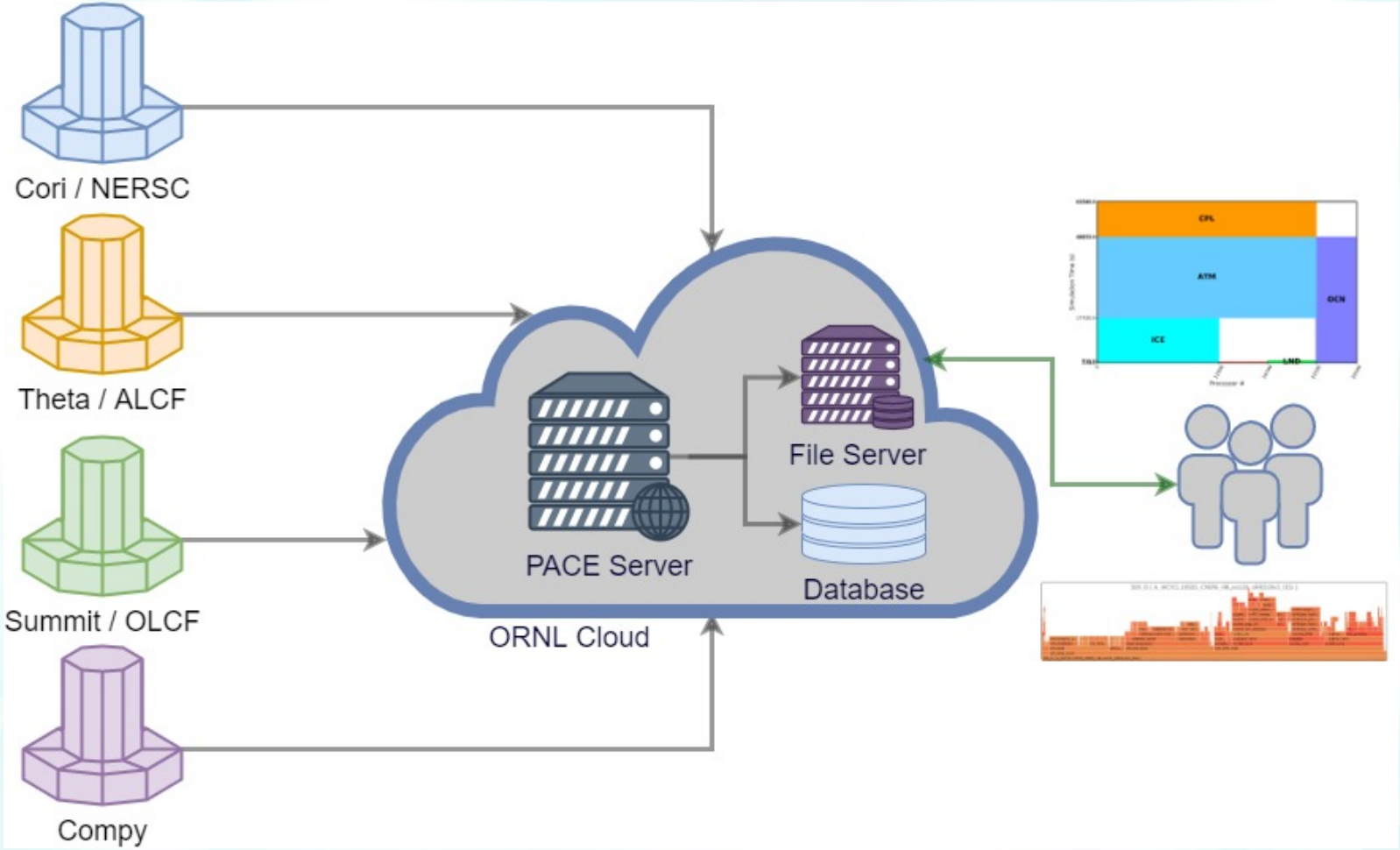
Goals

Provide executive summary of E3SM performance to stakeholders

Facilitate:

- Interactive analyses and deep-dives into experiments and application sub-regions, as desired,
- Tracking performance benchmarks and simulation campaigns of interest,
- Facilitating performance research on load balancing and process layouts,
- Identification of bottlenecks to inform targeted optimization efforts.

PACE Architecture



PACE Statistics

- 178 users
- 10 platforms
- Performance Benchmarks
  - High-res Atmosphere
  - High-res Ocean
- Simulation Campaigns
  - DECK v1
  - High-res Water Cycle
  - Cloud-resolving model (MMF) Early Science
  - Cryosphere
  - BGC campaigns

- **37,332** experiments
- **1 million+** model input files (XML, Namelist, RC)

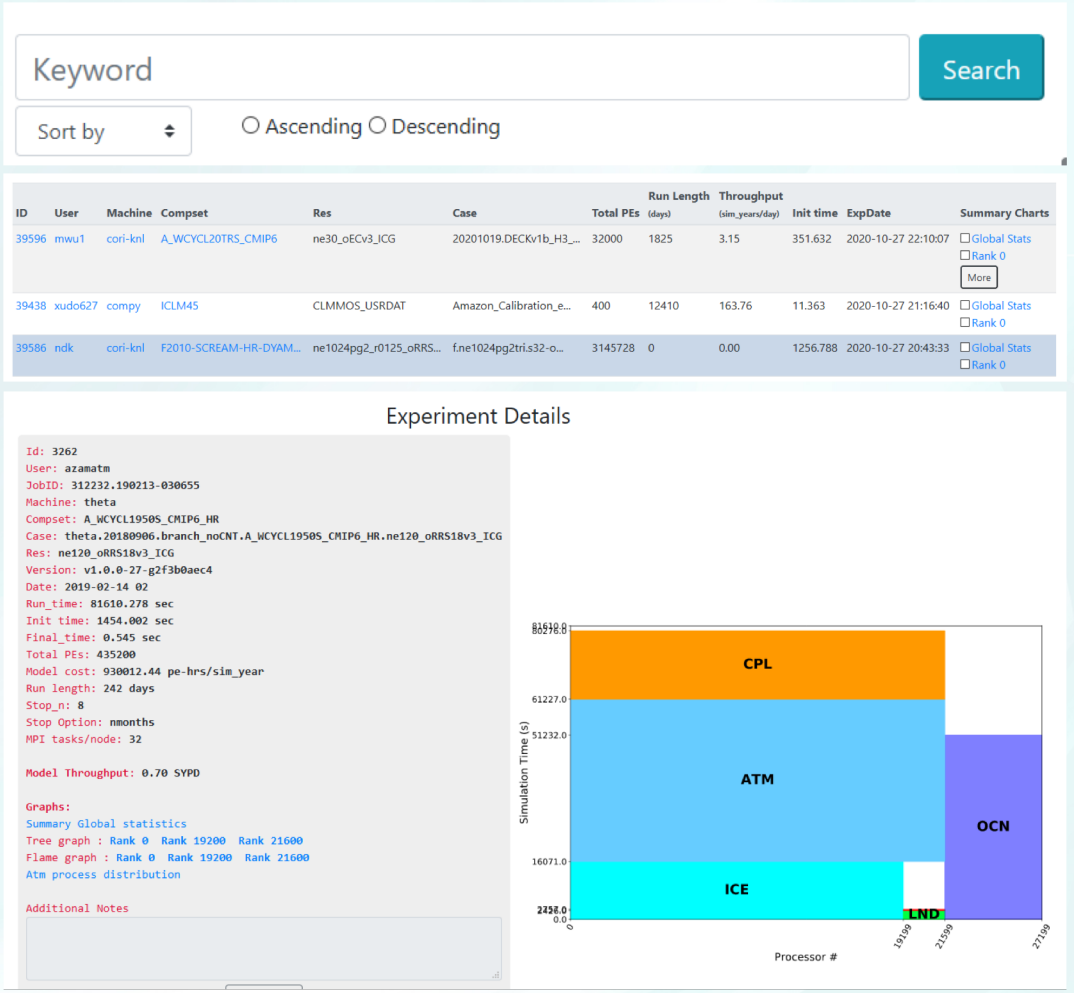
Aggregate statistics and reports can inform:

- DOE INCITE and other compute allocation proposals
- Computing procurements

Usage

https://pace.ornl.gov

- Search for existing experiment using case, compset, grid, user etc. (Autocomplete supported)
  - Sort by desired criterion
- Click on a row from search results to dive into specific experiment
- Experiment details page contains
  - Metadata: user, machine, date etc.
  - Provenance: Browse model inputs
  - Performance overview
    - Model, Component throughputs
  - Process layout diagram
  - Links to detailed performance graphs



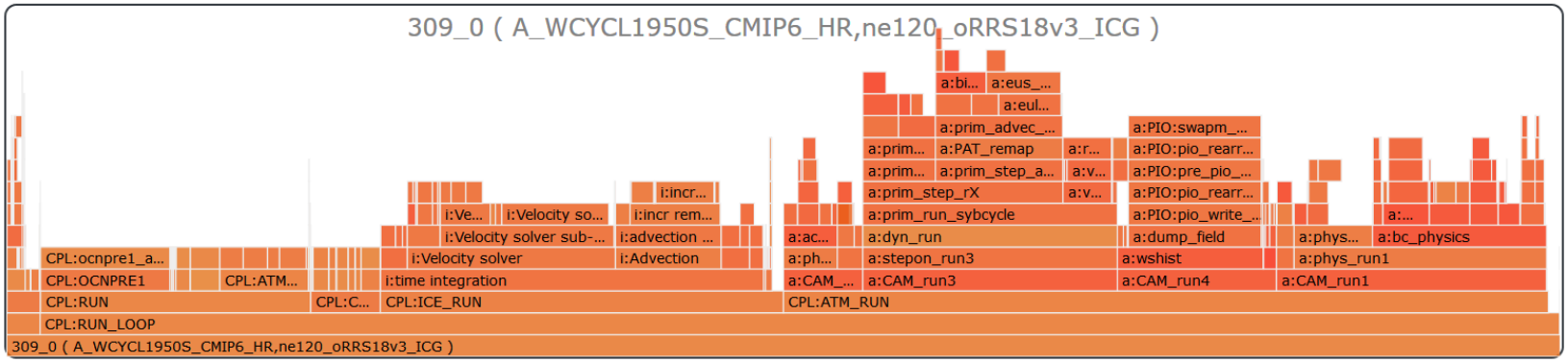
Tree Graph

Summarize time taken by model components

Recursively explore time taken by model sub-regions

Flame Graph

High-level overview of a parallel process execution time



Ongoing and Future Work

Assistant

- Simulation planning
- Process layouts
- Data analytics
- Anomaly detection
- Allocation reports
- I/O Performance

Steve The Minion – from Pixabay
https://pixabay.com/photos/minions-banana-steve-the-minion-2552584/

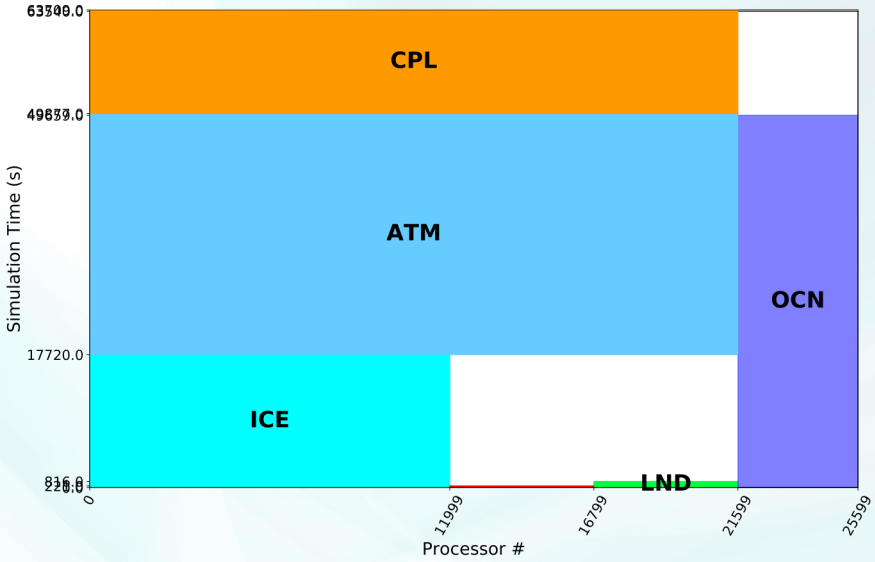
Wizard

- Recommend optimizations
- Optimal resource allocations
- Machine Learning
- Communication optimization
- Active monitoring and reporting

Dennis Jarvis from Halifax, Canada / CC BY-SA (https://creativecommons.org/licenses/by-sa/2.0)

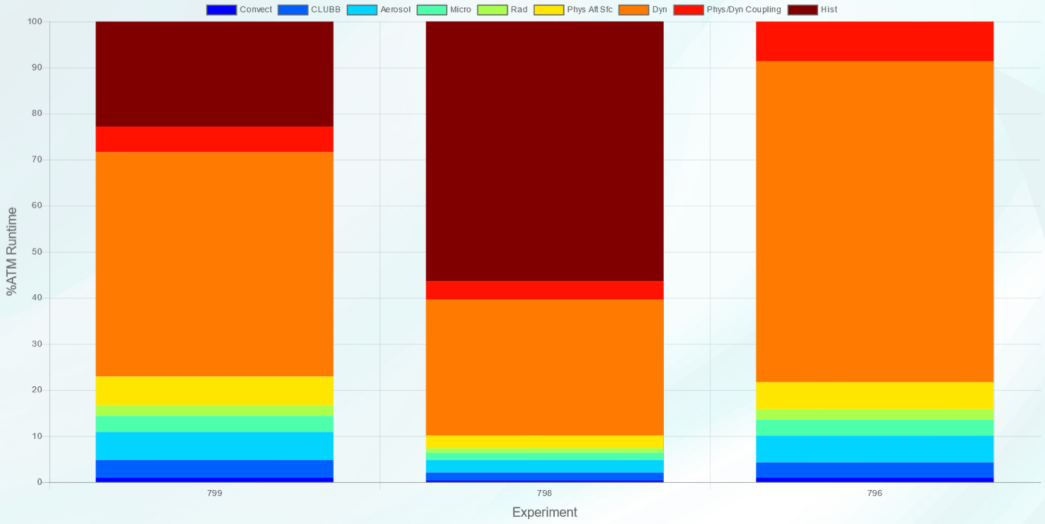
Performance Research Directions

Resource Allocation and Load Balancing



MPI Task Mapping

Targeted Optimization



Atmosphere model time distribution

Links

- PACE Portal
- DOE BER Highlight
- Reference Page on E3SM.org
- Video - Web Portal Features
- Video - How to Upload Data

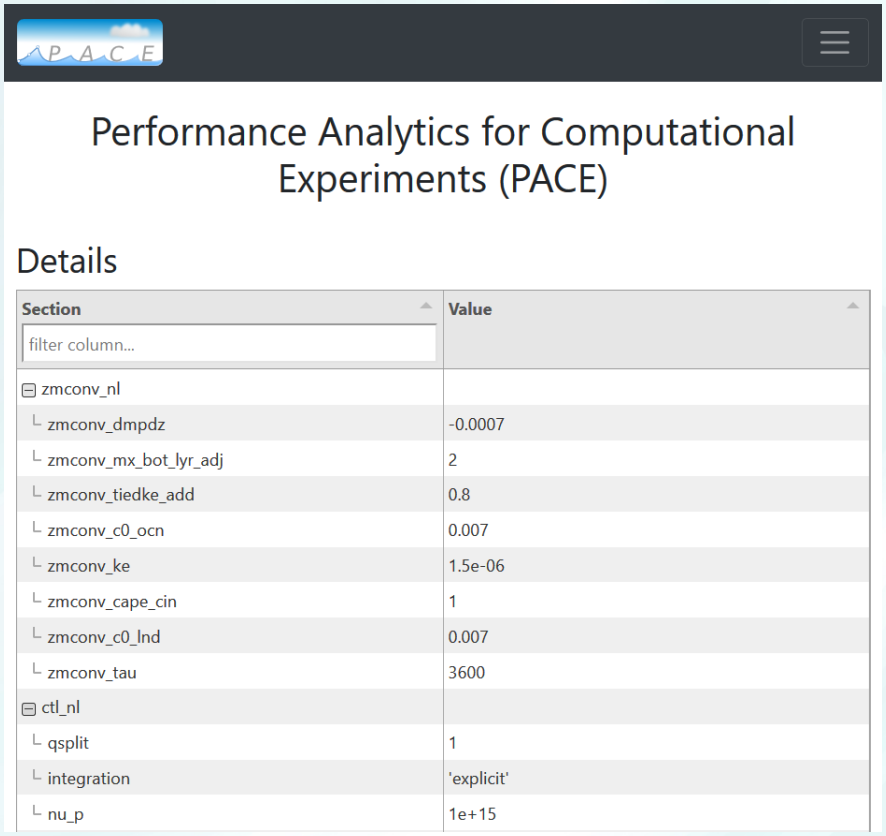
https://pace.ornl.gov

Thanks!

Contact:
sarat@ornl.gov

Recent Capabilities

- Parse Model Inputs
  - XMLs: 261,823
  - Namelists: 595,809
  - RCs: 33,850
  - Raw data (.zip): 34,029
  - Total experiments: 34,029
- Automated nightly uploads
  - Using Jenkins
  - Deployed on E3SM supported machines



View and filter namelists

Questions or further discussion

- We will answer (in text) any Q's on the zoom Q&A or Chat from the talk
- For more Q's or discussion, see #day4 channel on Slack