# Update on the Earth System Modeling Framework (ESMF)

D. Rosen, G. Theurich, R. Oehmke, U. Turuncoglu, B. Sacks

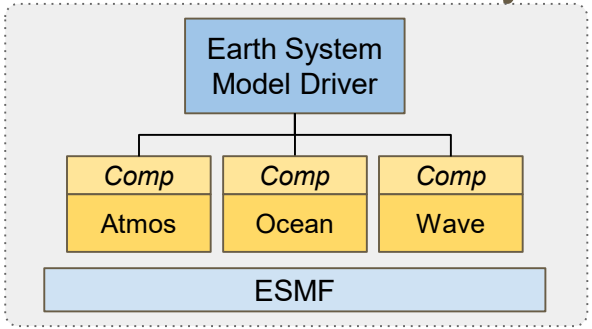
(2023) *6th Workshop on Coupling Technologies for Earth System Models*

NCAR CGD

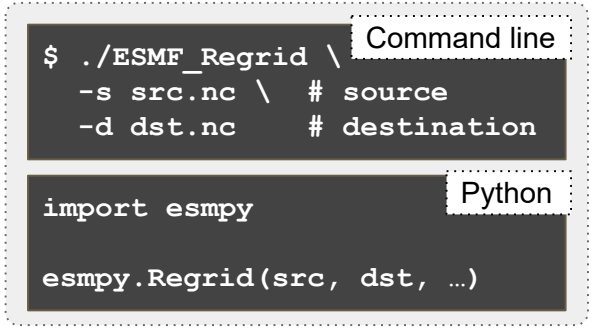
# Overview of ESMF/NUOPC

The **E**arth **S**ystem **M**odeling **F**ramework (**ESMF**) is a **parallel high - performance software infrastructure** used in coupled Earth science applications.

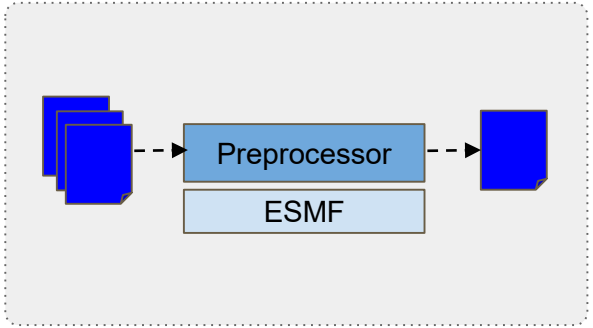
## There are different ways to use ESMF:




***Coupling infrastructure*** *in a modeling system (includes the NUOPC Layer)*




***Offline tool*** *for* ***grid remapping*** *and* ***interpolation weight generation*** *(command line and Python)*




***Library*** *used to construct* ***custom tools****, such as preprocessor or postprocessor*

# Modeling Systems using ESMF/NUOPC

NCAR | COMMUNITY EARTH SYSTEM MODEL CESM

MOSSCO
Helmholtz-Zentrum Geesthacht
Zentrum für Material- und Küstenforschung

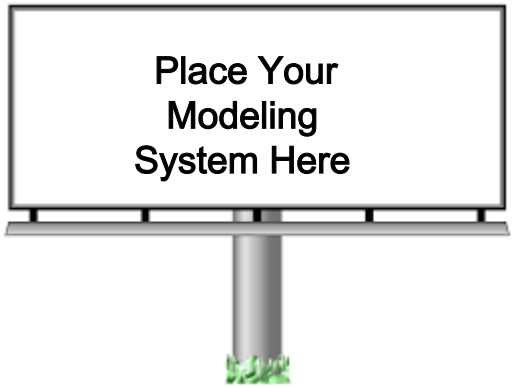
UFS

COAMPS™

ICON
GETM

NASA
GEOS

ESPC



# ESMF Software Process

**ESMF Change Review Board** 


- multi -agency board - suggests, reviews, and accepts new developments

**ESMF Support** 

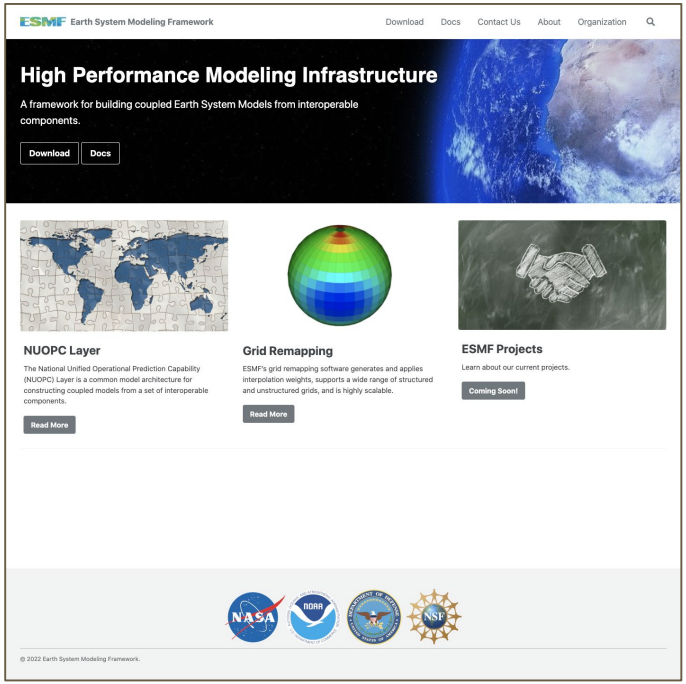
- community driven product feedback (bugs, new features, performance)

**Regular Releases (Most Current Release 8.4.0)** 

- semi-annual feature releases
- patches released as needed

**Open Source on GitHub** 

- https://github.com/esmf-org/esmf




https://earthsystemmodeling.org/

# Updates

- Performance & Scalability Improvements
- Enhanced Usability
- Framework Improvements
- I/O Capabilities
- Portability

# Earth System Modeling eXecutable (ESMX

Brand New Component **Coupling Application Layer**

- provides an executable
- provides a NUOPC-based coupled system driver
- uses CMake to embed components into a system

Motivations:

- **Accelerate development** of new NUOPC-based systems.
- Introduce mechanism for **testing** model components and coupling systems.
- **Reduce maintenance cost** for established NUOPC-based systems.
- **Standardize processes** for NUOPC-based systems. (configuration files, build procedures, etc.)
- **Accelerate new feature roll-out** for NUOPC/ESMF.

# Configuration Files

Build configuration (YAML)

```
components:
  tawas:
    cmake_config: TaWaS/tawas.cmake
    fort_module: TaWaS/tawas.mod
  lumo:
    cmake_config: Lumo/lumo.cmake
```

with new build configuration options in development

Run configuration (ESMF Config)

```
logKindFlag:          ESMF_LOGKIND_MULTI
ESMX_component_list:  ATM OCN
ATM_model:            tawas
ATM_omp_num_threads:  4
OCN_model:            lumo
OCN_petlist:          1 3

startTime:  2012      - 10- 24T18:00:00
stopTime:   2012      - 10- 24T19:00:00
runSeq::
   @900
      ATM - > OCN
      OCN - > ATM
      ATM
      OCN
   @
::
```

YAML run configuration coming soon

# Build System

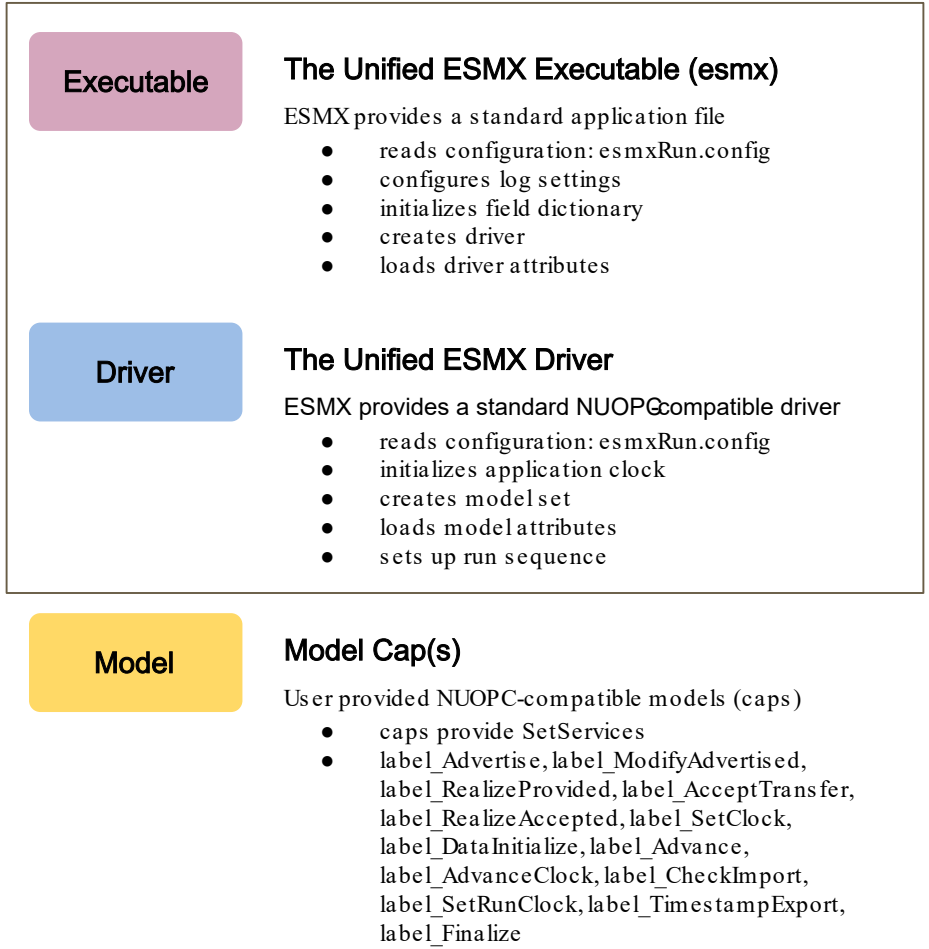
ESMX is CMake based

```
cmake -S$ESMF_ESMXDIR -Bbuild
cmake --build build
```

ESMX uses <Model>.cmake file to link model libraries for each model

ESMX Driver creates model components and registers the SetServices routine from each model cap

Future release will build model components from source with or without <Model>.cmake



# NUOPC Resource Control

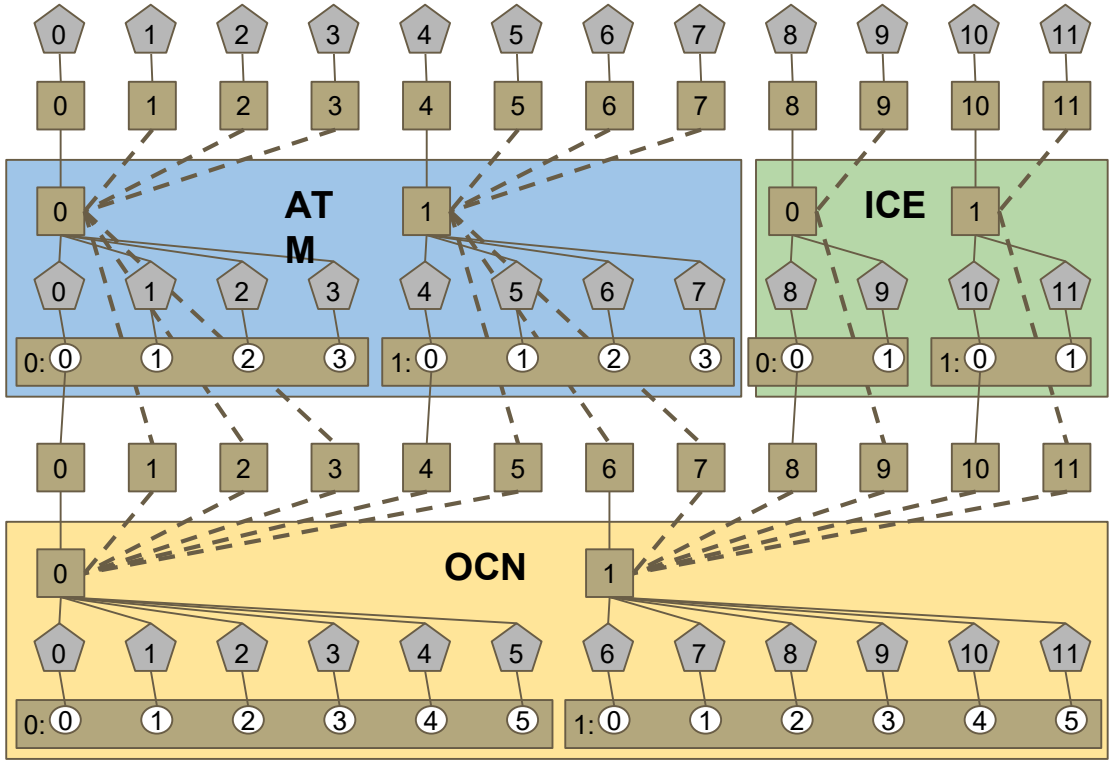
Motivation

- MPI scaling limit reached at different number of MPI tasks.
- Many components implement hybrid parallelism via MPI+OpenMP to extend their scalability.
- The optimal mix of OpenMP threads per MPI task is model-, problem-, and hardware- dependent.

Resolution

- Implement per component resource control through NUOPC interface.

# Resource Control for Coupled Hybrid Parallelism




ATM and ICE run concurrently on different PEs with different threading levels.

OCN runs sequentially on the same PEs as ATM and ICE, but with different threading level.

# NUOPC Resource Control Configuration

driver configuration

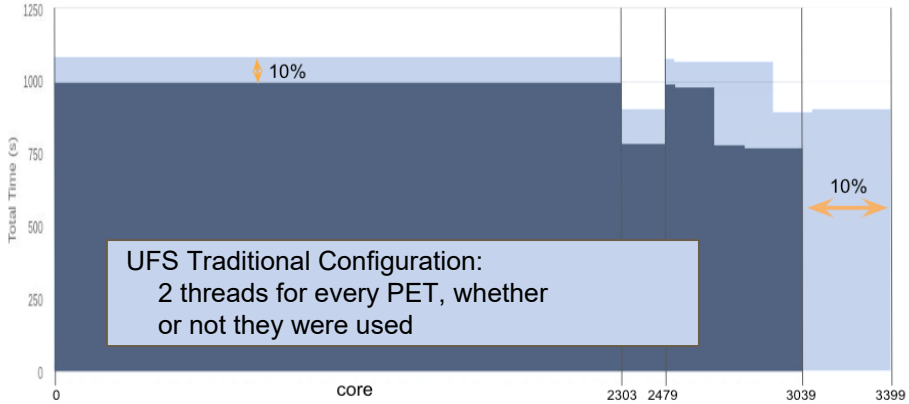
```
MED_petlist_bounds:          0 2303
MED_omp_num_threads:                 8
ATM_petlist_bounds:          0 2479
ATM_omp_num_threads:                 2
OCN_petlist_bounds:          2480 2679
OCN_omp_num_threads:                 1
ICE_petlist_bounds:          2680 2799
ICE_omp_num_threads:                 1
WAV_petlist_bounds:          2800 3039
WAV_omp_num_threads:                 2
```

- 288 tasks on 2304 cores, overlapping with FCST
- 2304 FCST threads + 176 WRT threads on **2480** cores
- 200 OCN tasks on **200** cores
- 120 ICE tasks on **120** cores
- 240 WAV threads on **240** cores

job card

```
#SBATCH -- nodes=76
#SBATCH -- ntasks  - per - node= 40
```

Total Run Cycle Time: Traditional vs ESMF-managed



# XGrid Introduction

Exchange grids were originally developed at GFDL [1]

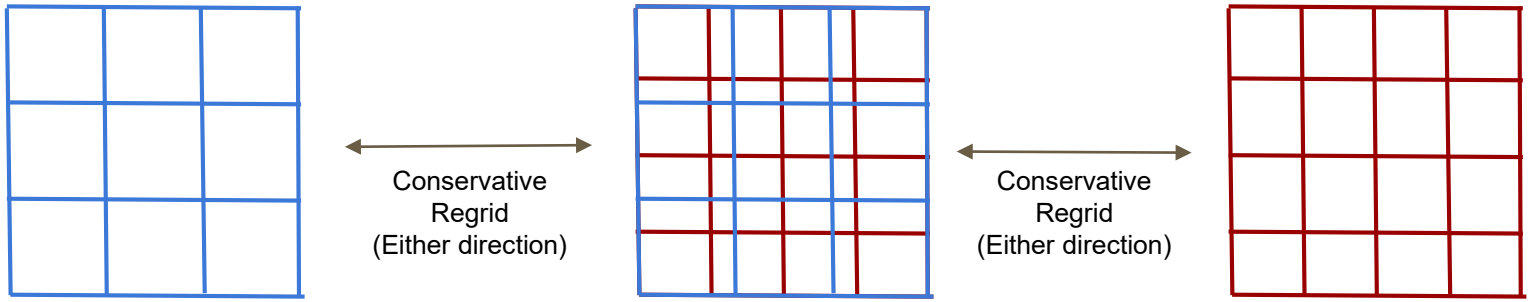
The ESMF XGrid was introduced into ESMF many releases ago

In recent years ESMF XGrid usage has increased

- CESM uses XGrids in their mediator for flux conservation between atmosphere and ocean by default.
- UFS is running experiments using XGrids in their applications.
- ESMF core team is working with NASA to try using ESMF XGrids in GEOS.

[1]Balaji, V., Anderson, J., Held, I., Winton, M., Durachta, J., Malyshev, S., & Stouffer, R. J. (2006). The exchange grid. A mechanism for data exchange between earth system components on independent grids. In *Parallel Computational Fluid Dynamics 2005* (pp. 179-186)

# eXchange Grid Example




**XGrid:**

- Intersection of side A and side B
- Each cell corresponds to exactly one* cell in one grid on side A and one cell in one grid on side B
- Can build ESMF Field/store data/do calculations on
- Can move data conservatively to/from each grid on each side to XGrid
- Can have any number of Grids and Meshes (or a mix) on each side

* However, it may correspond to only part of that cell.

# XGrid Recent Developments

- ESMF XGrid support for meshes containing elements with **>4 sides** .
- XGrid creation has been improved to be **more accurate and quicker** .
- Fields created on XGrids can now be used as **source, destination, or both** in regridding enabling the use of any regrid method or option on them
- ESMF can now return the **cell areas** for an XGrid in *ESMF_FieldRegridGetArea()*
- Bug and interface consistency fixes

# Further Updates Since CW2020

## Performance & Scalability Improvements

- ESMF_Info replaced ESMF_Attribute
- VMEpoch improvements
- *NUOPC Resource Control*
- DE-sharing between PETs
- Further MOAB adoption for the internal ESMF_Mesh representation
- Some 32-bit limits addressed

## Enhanced Usability

- *Earth System Modeling eXecutable*
- NUOPC Semantic Specialzation labels
- NUOPC RunSequence Alarm syntax
- ESMF Profiling available under NUOPC

## Framework Improvements

- *Improved XGrid support*
- Improved LocStream support
- Named Alias feature
- Optional auto calculation of Mesh node owners
- ESMF_GridRedist() API for moving grid support

## I/O Capabilities

- Multi-tile I/O capabilities
- More scalable MeshCreate() from file
- Switched to using PIO2 internally

## Portability

- Darwin (Mac OS) support for Gfortran/Clang combination

**Detailed Release Notes on GitHub:** https://github.com/esmf-org/esmf/releases

# Future Work

- Improvements to ESMX:  Extend the CMake based build mechanism to allow seamless integration with multi -component application builds. Move to a YAML based run configuration.
- Improved multi -tile I/O support:  Allow ungridded dimensions and cases other than 1 DE/PET. Support meta -data.
- Mesh write capability. Field on Mesh read/write capability.
- Spherical vector regridding support by regridding system via 3D Cartesian space.
- Monotonic 2nd order conservative regridding.
- Complete the MOAB integration effort.
- Support of dynamically changing grids.
- Explore approaches to mitigate the impact of models residing in GPU memory and other non -uniform memory architectures on coupling performance.
- NUOPC run sequence features: restart at any point of nested, resource re -balancing
- Time Manager Alarm extensions: irregular alarm time, external trigger signals.
- Field operations: simple arithmetic, merging with blending.