Offline vs Online Remapping Workflows

A Comparison of Spatial Remapping Implementations in E3SM Climate Model

Vijay Mahadevan[1] Iulian Grindeanu[1] Jason Sarich[1] and Robert Jacob[1]

[1]Argonne National Laboratory

5th Workshop on Coupling Technologies for Earth System Models

Date: September 21, 2020

Acknowledgements

Sponsors

This work is supported by the DOE-BER E3SM climate project.

E3SM Energy Exascale Earth System Model

https://e3sm.org/

Overview

Challenges in Remapping for Climate Simulations

Solution remapping on unstructured meshes is a complex process

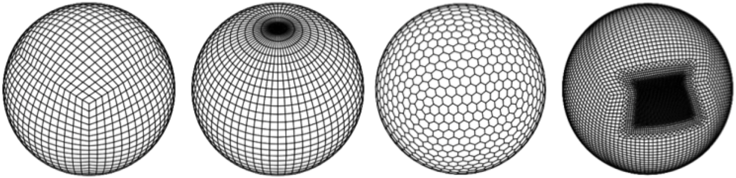
Sensitivity: Algorithmic invariance to underlying mesh topology

Conservation: Ensure global, (and) local conservation of critical quantities

Consistency: Retain discretization order and accuracy in a given norm

Monotonicity: Preserve global solution bounds during remap

Scalability: Dynamic load balancing for optimal performance



Current state of art: Offline-Online model in E3SM

Imposes severe bottlenecks in the overall workflow

Software tools

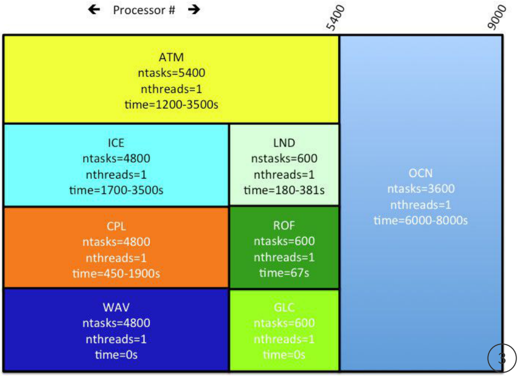
- **ESMF-RegridWeightGen:** Bilinear, higher-order patch, first/second-order conservative, and two types of nearest neighbor interpolations.
- **TempestRemap:** Arbitrary order conservative and consistent remapping with monotonicity, between FEM/FDM/FVM discretizations
- **ncremap:** Optionally re-use ESMF or TempestRemap with specific support for climate NetCDF files, and parallelization with OpenMP
- **Alternatives:** SCRIP, Common Remapping Software (CoR), YAC interpolator

MOAB: A flexible infrastructure for grid aware coupling

- Full online computation of conservative and monotone remaps
- No approximations to mesh topology or field discretization description
- Optimal parallel performance characteristics with arbitrary PE layouts

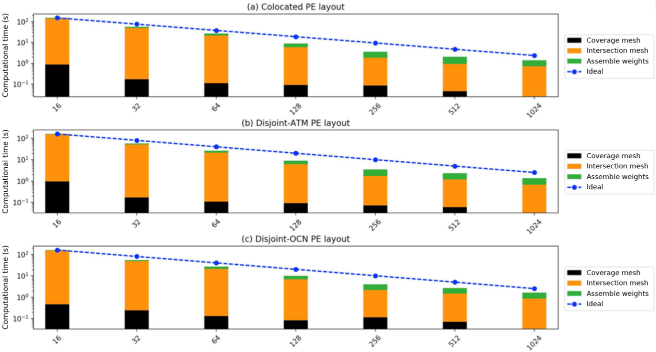
Workflow

- Offline: `mbtempest` tool
  Conservative map generation and I/O on colocated PEs
- Online: `iMOAB` API
  Scalable linear map generation on arbitrary component and coupler PE layouts



Spectral projection with Zoltan repartitioner on Cori

- Higher resolution model provides excellent scalability upto 1024 processes
- The coverage and intersection mesh computation time are bounded
- More optimizations for the bandwidth-limited weight assembly possible
- Switch from offline-online to fully-online seamlessly based on case demands



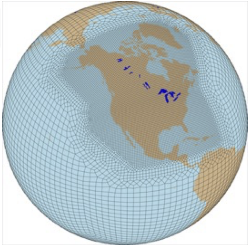
NE120 test case with different PE layouts on Cori (KNL)

Mahadevan V. S. et al., (2020), Geoscientific Model Development

E3SM v2+ Changes for Coupling

Tri-grid configuration will be default

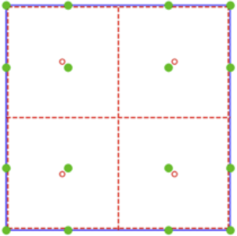
- What is Tri-grid ? Atmosphere, Land/River, Ocean/Sea-ice.
- New ATM-LND/RIV maps; Enables closer coupling between land and river models: water management, irrigation, inundation, . . .



New RRM Grid

Atmospheric physics now uses a FV "Physgrid"

- Compute physics on subcells of spectral element (SE) instead of GLL point locations; E3SM v2 will use 2x2 subcells "pg2" per SE
- Physics columns reduced by factor=4/9 compared to SE; Speedup > 100%
- Transformation between SE and PG done internal to EAM with mass conserving algorithm; supports new Regionally Refined Meshes



SE4-PG2 Mapping

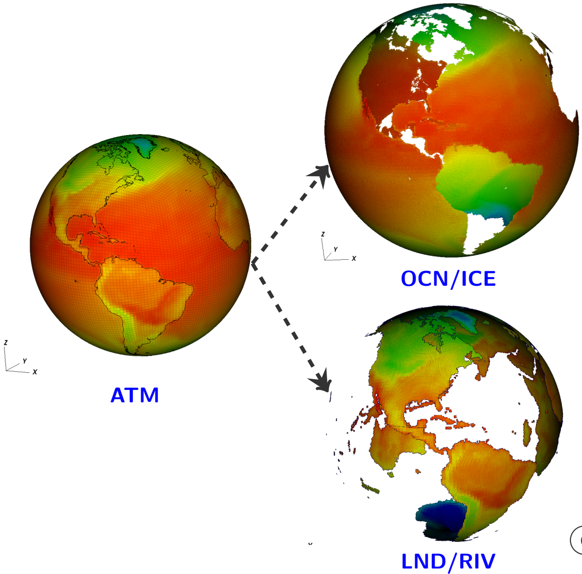
E3SM v2.0 scheduled release: Sept, 2021

MBTR Tri-Grid Configuration Support

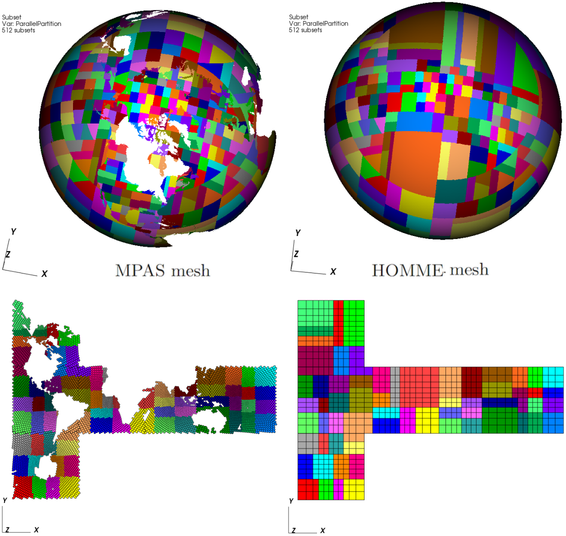
MOAB-TempestRemap Tri-Grid online remap computations for E3SM

ATM→OCN
(Coupling with MPAS is FV–FV instead of SE–FV)

ATM→ $\frac{1}{2}^o$ LND grid
(LND and RIV components now on separate common $\frac{1}{2}^o$ grid)



Coupler Grid Decomposition Optimization



"Inferred" Partitions on 3-D grid (top) and on its gnomonic projection (bottom)

Topological holes increase computation cost of $\Omega_{S\cap T}$ for $OCN \rightarrow ATM$ vs $ATM \rightarrow OCN$

Load-Balance Analysis: 1024 cores

- RCB: Independent parts
- iRCB: Inferred target partition in 3-D
- iRCB-G: Inferred target in 2-D Gnomonic plane

NE256-PG2

OCN(3.7M) $\rightarrow$ ATM(1.57M)

Strategy	$\Omega_{S,C}$	$\Omega_{S\cap T}$
RCB	1.81	20.49
iRCB	0.66	7.72
iRCB-G	0.43	3.08

E3SM Software Updates

MCT version: 2.11

Minor changes:

- Fix bug in Rearranger when using alltoall
- impi optimization for large task counts
- Support for new ifx Intel compiler

To be released next month!

MOAB-TempestRemap

New Release:

- `MOAB` version 5.2.1 released on August 22, 2020
- `TempestRemap` version 2.0.5 released on June 12, 2020
- New Conda package available for MBTR stack usage

Summary

- With **MOAB-TempestRemap** (MBTR) integrated in E3SM, a path to achieve scalable performance, without sacrificing coupling accuracy
- MOAB-E3SM interface implemented for ATM, OCN/ICE and LND/RIV components with remap of two-way coupled data
- Performance comparison of MBTR vs MCT coupler in E3SM underway

Miles to go before we sleep …

- Property preservation in MBTR with **ClipAndAssuredSum (CAAS)**
- Utilize advanced **multi-grid inferred partitions** in the online remapping case for synchronized grid and coupled data transfers
- Distributed **component-to-component** direct transfers (same API)
- Support **meshless** higher order Weighted Least-Squares ENO and GMLS remap scheme through MOAB for scalar data

EXTRAS

EXTRA SLIDES IF NEEDED FOR DISCUSSION

Regridding with MOAB-TempestRemap

Algorithm 1 Compute Intersection mesh from Ω_S and Ω_T on N_x

procedure COMPUTE SOURCE COVERAGE AND INTERSECTION MESH

Input: *Partitioned Ω_S and Ω_T on coupler PEs.*

Output: Intersection mesh $\Omega_{S \cap T} = \Omega_S \bigcup \Omega_T$.

- *Migrate* component mesh from component to coupler PEs with dynamic repartitioning with Zoltan2
- Accumulate *source coverage* mesh to localize work
- Compute *source-target mesh intersection* on coupler PEs
  - a: Advancing front intersection based on adjacency graph
  - b: Kd-tree based search and locate point
- Store *communication graph* for point-to-point **c2x/x2c** transfers
- Utilize `TempestRemap` to *compute remapping weights* operator ($A_{S \to T}$)

end procedure

Regridding with MOAB-TempestRemap

Algorithm 2 Compute solution remap from Ω_S and Ω_T

procedure COMPUTE SOLUTION PROJECTION WITH REMAPPING WEIGHTS

Input: *Coupled solution* $X_S \in \Omega_S$ *on source component PEs.*

Output: *Coupled solution* $X_T \in \Omega_T$ *on target component PEs.*

  Communicate X_S to coupler PEs using *communication graph* for **c2x**

  Apply remap weights operator: $X_T = A_{S \to T} X_S$ using SpMV algorithm

  Reduce X_T in coupler PEs for shared DoFs

  Communicate X_T on coupler PEs to target component using *communication graph* for **x2c**

end procedure