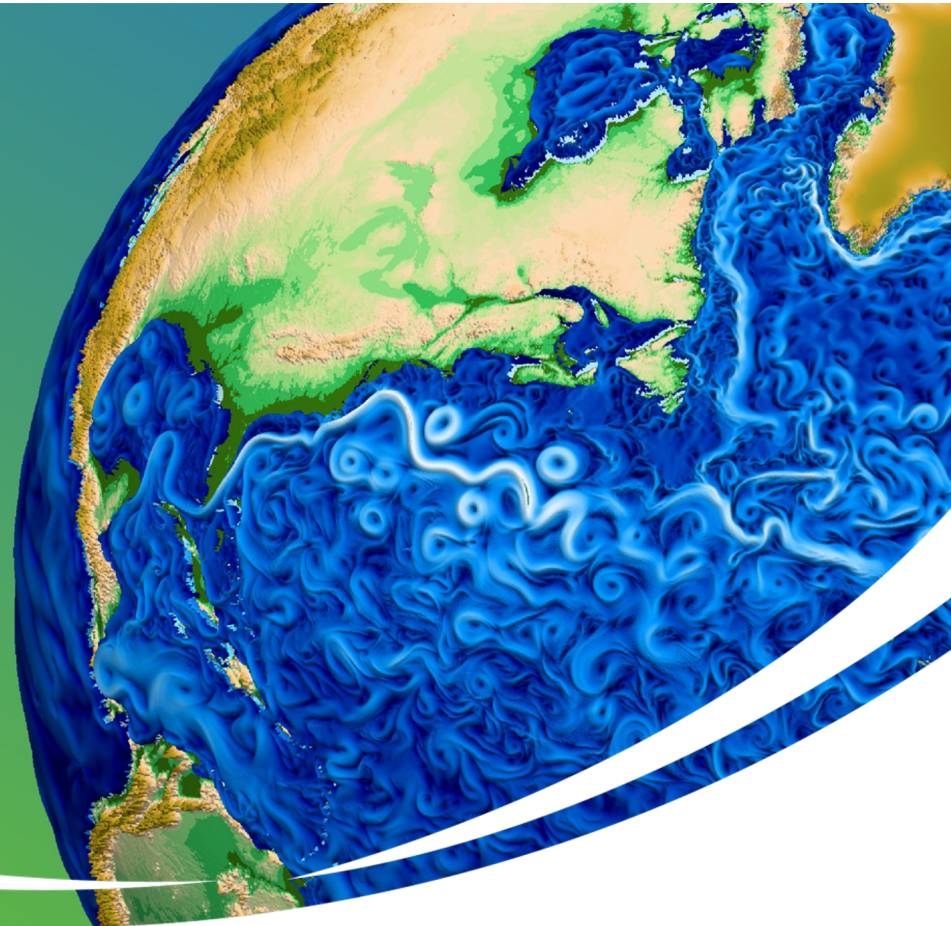


# MCT and MOAB coupling in E3SM

Rob Jacob, Iulian Grindeanu,  
Vijay Mahadevan, Jason Sarich

6<sup>th</sup> Workshop on Coupling Technologies for  
Earth System Models

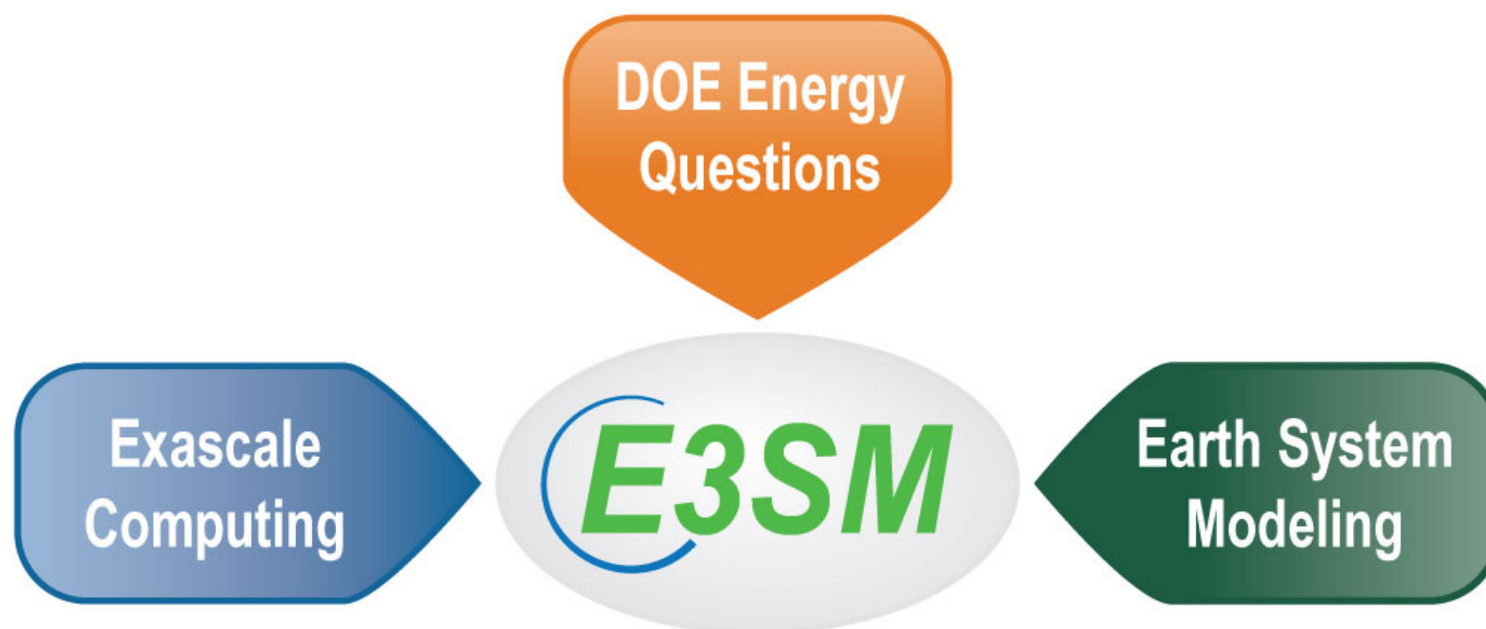
Toulouse, France, Jan 20, 2023



# The U.S. National Lab System



## Energy Exascale Earth System Model



*The E3SM Mission: Use exascale computing to carry out high-resolution Earth system modeling of natural, managed and man-made systems, to answer pressing problems for the U.S. DOE.*

3 more years of funding approved! Jan 2023-2026



# E3SM hardware landscape

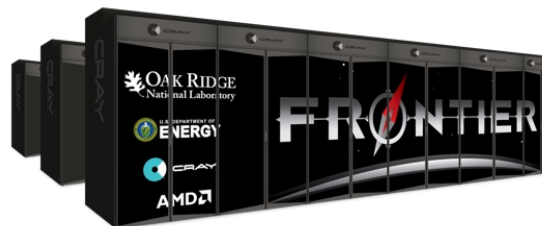
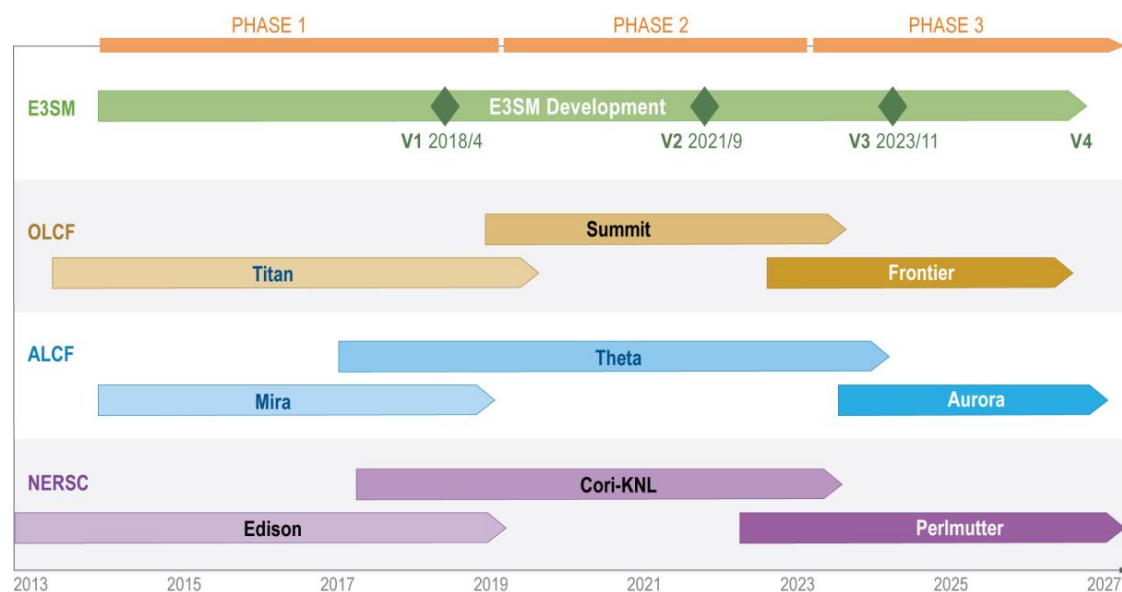
## Upcoming DOE machines are all GPU based:

- 2021: NERSC Perlmutter 8 MW
  - 6000 **NVIDIA GPUs**.
  - 3000 CPU-only nodes (AMD)
- 2022: OLCF Frontier 30 MW (#1 on top500)
  - 9400 nodes
  - 1 64-core AMD EPYC CPU + 4 **AMD MI250x GPUs** per node.
- 2023: ALCF Aurora ~40 MW
  - 9000 nodes
  - 2 Intel Sapphire Rapids CPUs and 6 **Intel Ponte Vecchio GPUs**

## E3SM dedicated CPU resources:

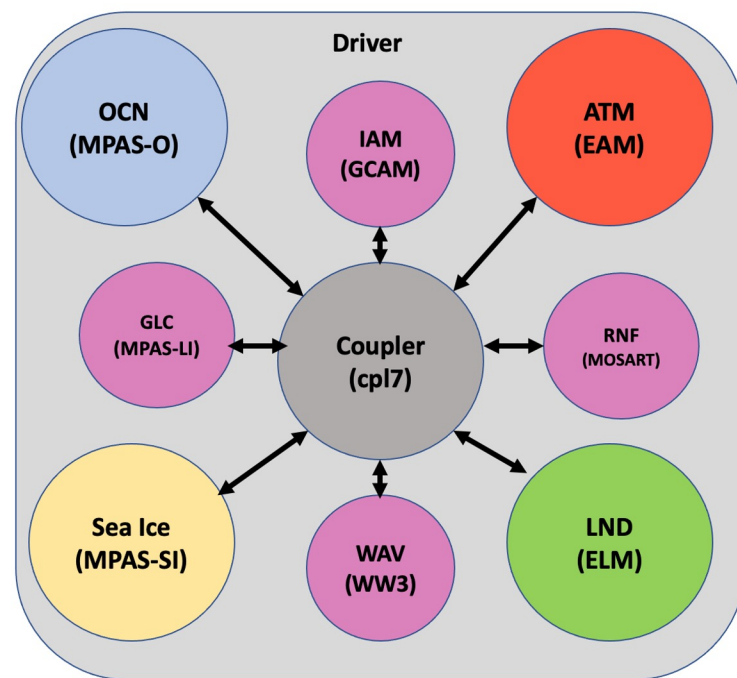
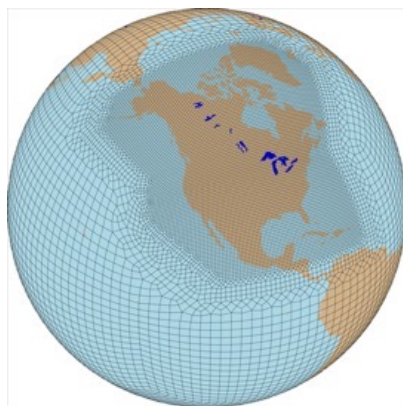
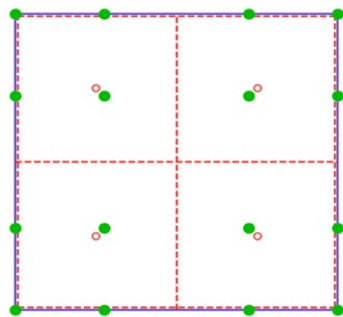
- Anvil, Compy, Chrysalis
- 1200 CPU-only nodes, ~0.6 MW

(slide from Mark Taylor)



## E3SM (coupler-related) developments since CW2020

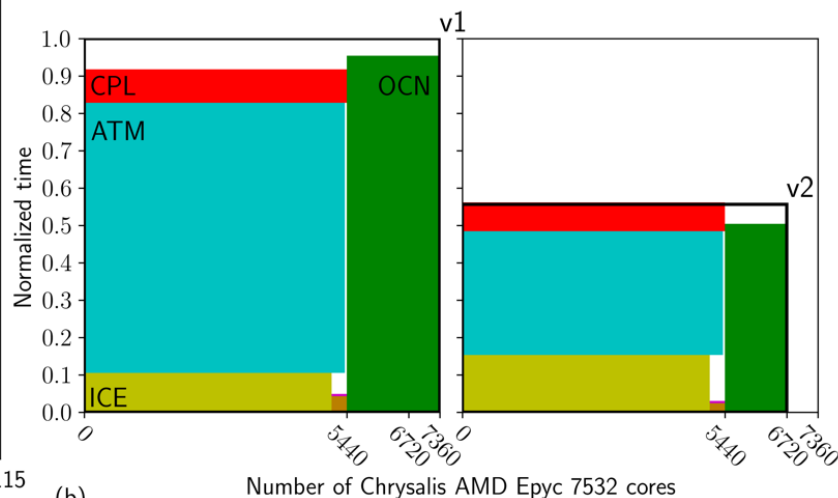
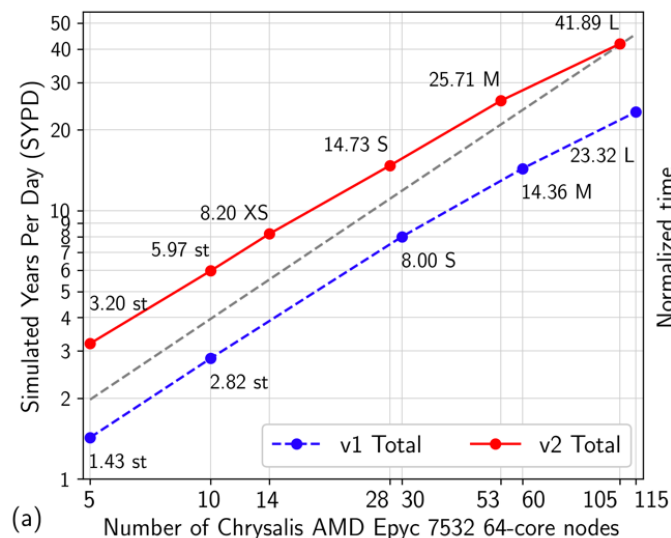
- Sept, 2021: v2.0 released
  - Introduced “physics grid” in atmosphere.
  - Regionally refined meshes possible in atm, ocn, lnd, sea-ice.
  - Coupler: cpl7/MCT
- Jan, 2023: V2.1 released
  - New parameterizations in ocean improved AMOC.
  - Additional components (experimental)
  - Coupler: cpl7/MCT



[github.com/E3SM-Project/E3SM](https://github.com/E3SM-Project/E3SM)

# E3SM v1 and v2 performance

Performance of maint-1.0 A\_WCYCL1850S\_CMIP6.ne30\_oECv3 and v2.0.0 WCYCL1850.ne30pg2\_EC30to60E2r2



- V3: Still mostly Fortran/CPU based. June 2023
- V4: Exascale-capable. C++ components. June 2026



## What exactly is cpl7/MCT?



MCT just provides “plumbing and wiring” for a coupler.

Cpl7 is the driver and additional data types (built on MCT) for a coupled system

- [seq\\_comm\\_mct.F90](#) - lay out models on mpi tasks and build MPI communicators according to namelist input.
  - Initialize `MCT_World`
- [component\\_type\\_mod.F90](#) – one instance per model which has most of the MCT data for the model itself and its representation in the coupler.
  - Model: `MCT_GSMap`, `MCT_AttributeVector` for send, AV for receive. `MCT_GeneralGrid` for lats,lons, GlobalID
  - Model-in-coupler: `MCT_GSMap`, `MCT_AttributeVector` for send, AV for receive, `MCT_GeneralGrid`.
  - Model sets its parts of the component, coupler sets its parts.
- [seq\\_map\\_mod.F90](#) - one method to do either a copy of 2 AVs, a rearrange (with `MCT_Rearrange`) or an interpolation (with `MCT_SparseMatrixPlus` and `MCT_MatAttrVectMul`)
  - Plus a method to read in mapping weights from a file and load them in the `SparseMatrix`.





## What exactly is cpl7/MCT?



- [seq\\_fld\\_mod.F90](#) – Describe all the fields going between components and coupler.
  - `seq_flds_a2x_states=`  
`Sa_z:Sa_topo:Sa_u:Sa_v:Sa_tbot:Sa_ptem:Sa_shum:Sa_pbot:Sa_dens:Sa_uovern:Sa_pslv:Sa_co2prog:Sa_co2diag`
  - Colon-delimited strings used to define Attributes in MCT.
- [cime\\_comp\\_mod.F90](#), [cime\\_driver.F90](#) – The top-level “main” and init, run and finalize of the coupler.
- Cpl7/MCT is the coupler in CCSM4, CESM1, CESM2.0-2.2, E3SM1, E3SM2

Craig, Vertenstein, Jacob, 2012, “A new flexible coupler for earth system modeling developed for CCSM4 and CESM1”, *Int. J. High Perf. Comp. App.*, 26(1), 31-42

(cpl6/MCT was the coupler in CCSM3 (circ. 2005).

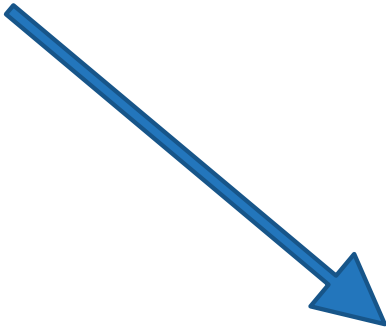


# Nov, 2022 marked 20 years of MCT!



Change Log at  
<https://www.mcs.anl.gov/research/projects/mct/changes.html>  
OR [shorturl.at/bhiR8](http://shorturl.at/bhiR8)

## Versions and Release Dates



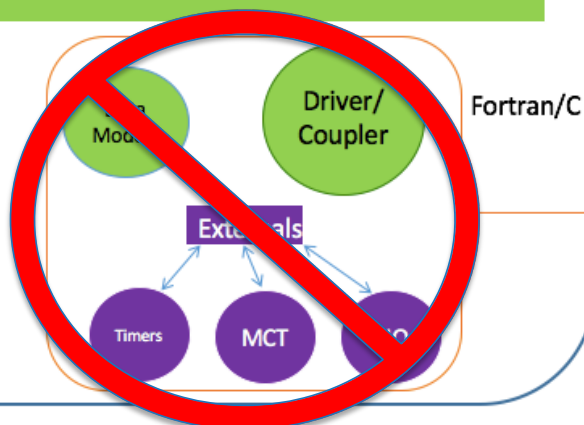
10/18/00: Initial prototype  
02/09/01: working MxN transfer  
04/27/01: parallel SparseMatrix multiply  
03/29/02: Rearranger for transposes  
11/14/02: Version **1.0.0** released.  
04/23/04: Version **2.0.0** released.  
05/24/04: Version **2.0.1** released.  
07/11/04: Version **2.0.2** released.  
02/11/05: Version **2.1.0** released. (Also part of CCSM3) Released June 16, 2004  
12/01/05: Version **2.2.0** released.

# Wasn't cpl7/MCT part of "Common Infrastructure for Modeling the Earth (CIME)"?

## CIME

**Case Control System:** python scripts to configure, build, and run a CIME-driven climate model  
`create_newcase`, `case.setup`, `case.build`, `case.submit`  
 Written in Python. XML configuration files.

**Misc tools:** weight generation, offline load balancing, `netcdf` compare, statistical tests



Core Functionality

Modeling support tools

External libraries  
(provided for convenience)

Compiled in with  
E3SM,  
CESM, ...

Cpl7/MCT was part of CIME developed jointly by CESM/E3SM

We realized different science/computation goals made it better to each have our own copy of this code.

E3SM/driver-mct (cpl7/mct)  
 E3SM/components/data\_comps  
 E3SM/externals/mct

E3SM/CESM are still collaborating on the CIME Case Control System for configure, build, run, test.

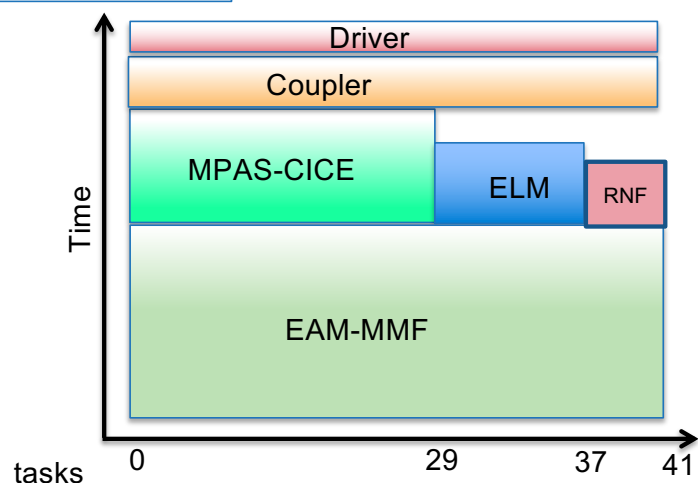


## Some E3SM developments in cpl7/MCT since CW2020

- Additional land-river coupling (irrigation, flooding)
- Allow biogeochemical coupling from river to ocean.
- Add "exclusive stride" option for GPU-exclusive configurations.
- Fixes for tri-grid merging (atm, lnd, ocn on 3 different grids)
- 2-way river-ocean coupling (SSH affects dynamic water stage boundary)
- Carbon budget calculation (with optional BGC fields)
- NOAA's WaveWatchIII added as a component
  - 2-way coupling with ocean, 1-way from atm, sea-ice
- GCAM integrated assessment model added as a component (still on a branch)
- MCT 2.11.0 (Released, Feb 2021. Fix occasional hang in Rearranger; autoconf update; ifx, gnu10 support (thanks to Andrea Piacentini) )

# Cpl7/MCT: mods to allow exclusive GPU access (1 of 2)

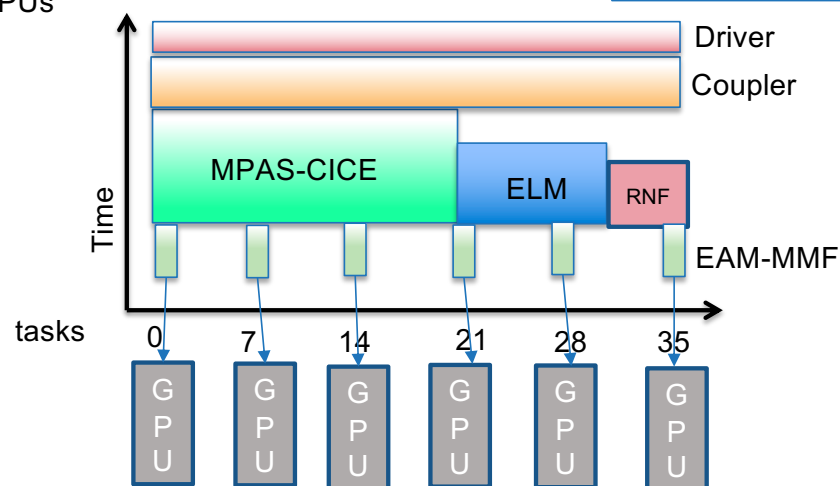
No GPU use



Comp	NTASKS	NTHRDS	ROOTPE	PSTRIDE
CPL :	42	1	0	1
ATM :	42	1	0	1
ICE :	30	1	0	1
LND :	8	1	30	1
ROF :	4	1	38	1

Summit Node:  
2 Power 9 (42 cores total)  
6 NVIDIA V100 GPUs

ATM using all GPUs



Comp	NTASKS	NTHRDS	ROOTPE	PSTRIDE
CPL :	42	1	0	1
ATM :	6	1	0	7
ICE :	30	1	0	1
LND :	8	1	30	1
ROF :	4	1	38	1



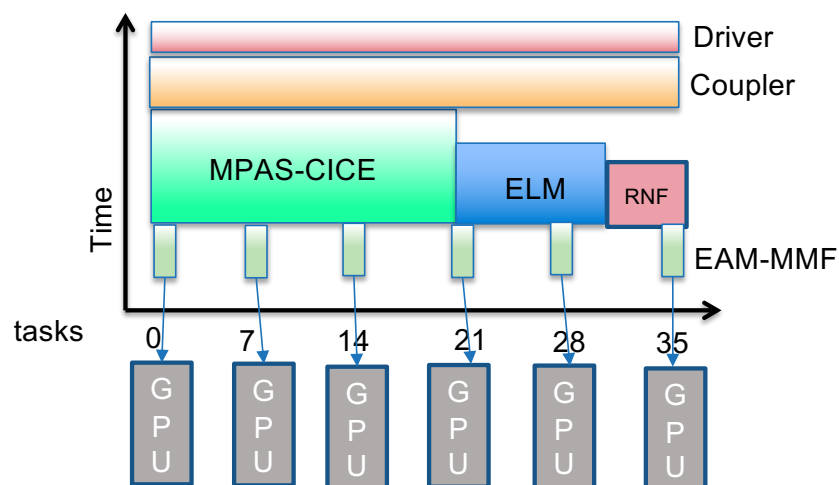
# Cpl7/MCT: mods to allow exclusive GPU access (2 of 2)

Summit Node:

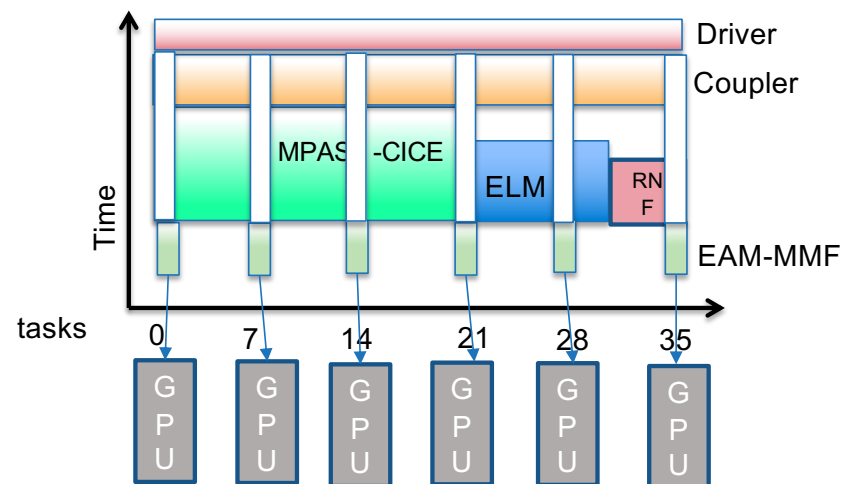
2 Power 9 (42 cores total)

6 NVIDIA V100 GPUs

“Exclusive stride” specified in driver namelist, used in seq\_comm\_mod to adjust communicators. Exclusive access **reduces total memory** on atm tasks.



Comp	NTASKS	NTHRDS	ROOTPE	PSTRIDE
CPL :	42	1	0	1
ATM :	6	1	0	7
ICE :	30	1	0	1
LND :	8	1	30	1
ROF :	4	1	38	1

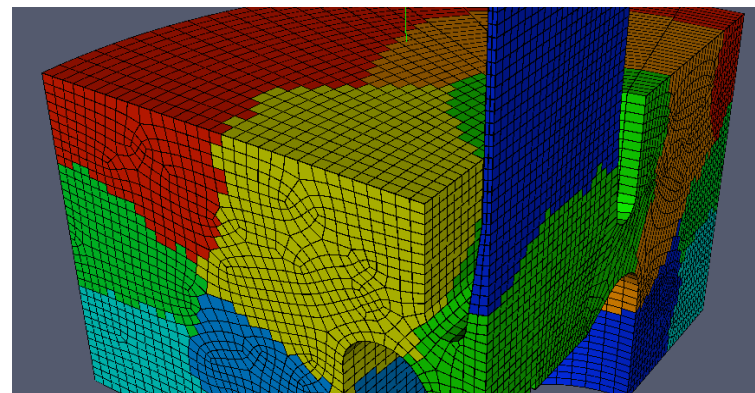


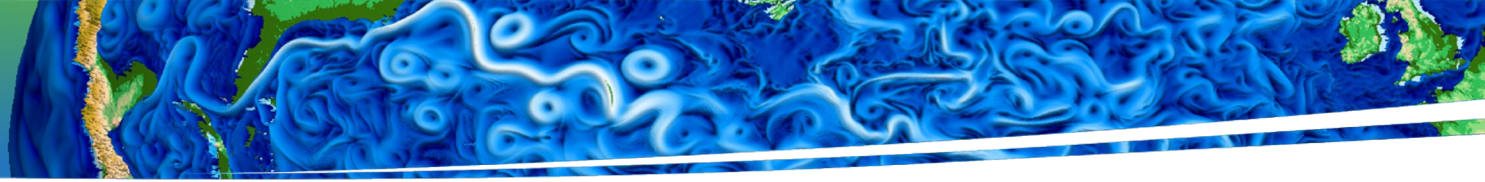
Comp	NTASKS	NTHRDS	ROOTPE	PSTRIDE	EXCL_STRIDE
CPL :	42	1	0	1	0
ATM :	6	1	0	7	7
ICE :	30	1	0	1	0
LND :	8	1	30	1	0
ROF :	4	1	38	1	0



## Motivations for MOAB-based coupler

- A complete mesh representation for:
  - Online mapping weight computation on arbitrary PE layouts
  - Smarter decompositions targeting better parallel performance of the coupler
  - Scalable topology and field data migration strategies that minimize communication bottlenecks
- Faster, less memory (array-based representations)
  - Mesh is distributed efficiently, and there are no global data structures
  - Eliminate the need for GSMap, which is replicated on each PE. In a worst case, GSMap can be  $3 \times \text{sizeof}(\text{int}) \times \text{Total number of grid points}$
- Correct mapping of high-order SE-FV (without a need for dual meshes)
- Ongoing developments to support GPU computations for map generation and field projection using C++ performance portable frameworks (e.g. Kokkos)



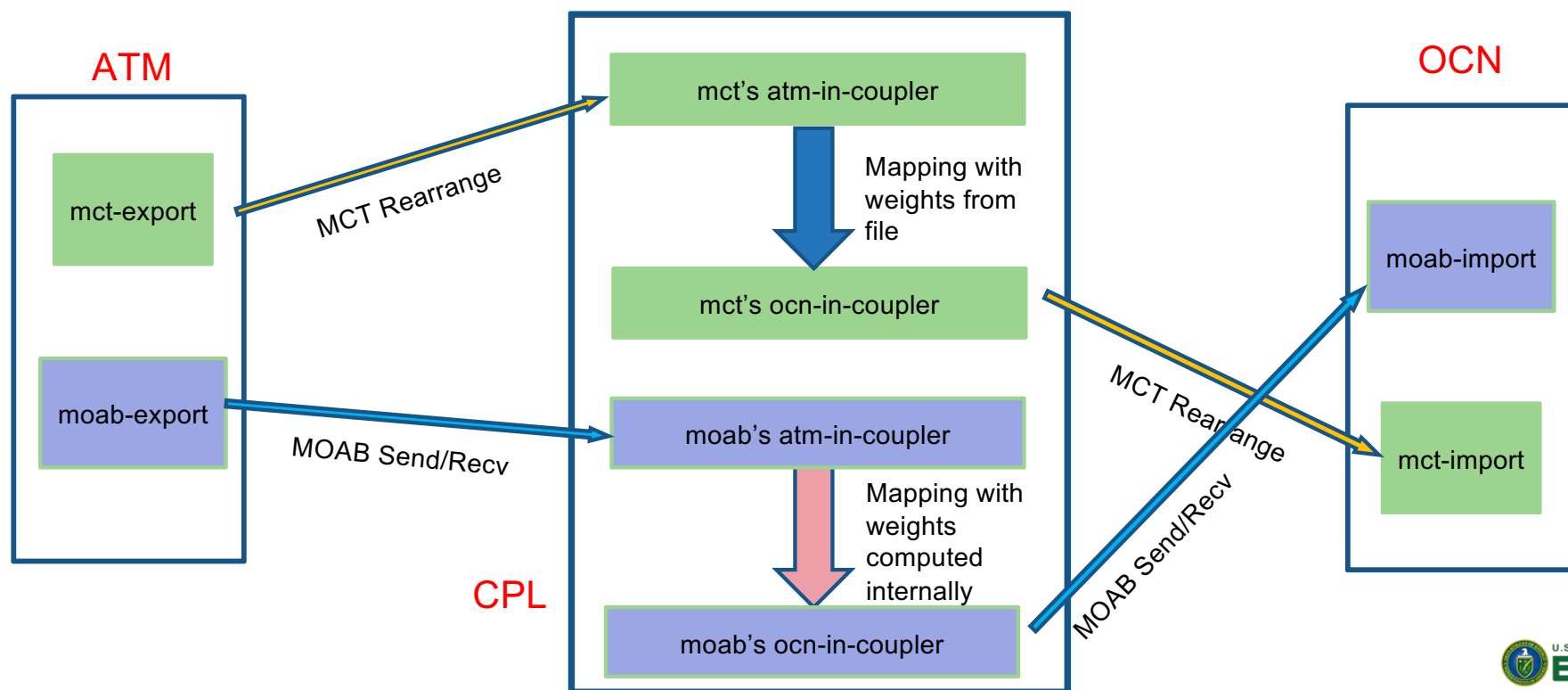


## Changes needed to MOAB

- Support for non-collocated applications.
  - MOAB had assumed all coupled apps shared the same processors. No longer.
  - Mesh Migration: Method to send MOAB's complete mesh description from one set of processors to another.
- Expose more functions in iMOAB: a convenient subset of MOAB functions callable from Fortran/C/C++.
- Augment existing `ParallelComm` functions with `ParCommGraph`, roughly equivalent to MCT Router.
- Change string separator from semi-colon to colon (to match MCT)
- Link to `TempestRemap` to calculate mapping weights on sphere.

# Converting cpl7/MCT to cpl7/MOAB

Introduce MOAB alongside existing MCT routines in cpl7







## Relationship between MOAB and MCT functions/concepts

### MCT

- Data accessed by strings called “attributes”
- A group of attributes is stored in an Attribute Vector ordered (varid, gridid)
- The grid is just another Av with extra Lists the keep the coordinate attribute names. Does not understand connectivity.
- A “stateless” library.
- “transparent” data types allow user to directly access values (eg. `Av%rAttr(n,m)` )
- Relationship between data in Av and grid it corresponds to is implicit.
- Communicate between components with Avs

### MOAB

- Data accessed by strings called “tags” (can use same strings: `Sa_tbot`)
- A group of tags is returned in an array ordered (gridid, varid).
- The mesh is a first-class object with full connectivity information.
- Keeps state of mesh and tag values internally.
- Opaque data types requires methods to get/set values.
- Tags are always associated with a specific mesh.
- Communicate between applications with groups of tags

# Examples of MCT and MOAB functions

```
MCTWorld_init(ncomps, global_comm, my_comm, myid)
```

```
    iMOAB_RegisterApplication('name', my_comm, myid, moabid)
```

```
MCTAv_init(Av, 'tag1:tag2', size)
```

```
    iMOAB_DefineTagStorage(moabid, 'tag1:tag2', type, dim, tagindex)
```

```
MCTImport_rAttr(Av, 'att', data(:))  (most people do Av%rAttr(attid,:) = data(:) )
```

```
    iMOAB_SetDoubleTagStorage(moabid, 'tag', datasize, ent_type ,data(:) )
```

```
MCTExport_rAttr(Av, 'att', data(:))  (most people do data(:) = Av%rAttr(attid,:) = )
```

```
    iMOAB_GetDoubleTagStorage(moabid, 'tag', datasize, ent_type ,data(:) )
```

```
    (can also get/set multiple values at once)
```

```
    iMOAB_GetDoubleTagStorage(moabid, 'tag1:tag2', datasize, ent_type ,data(:, :) )
```



# Examples of MCT and MOAB functions

MCTRearranger(SrcAV, TargetAV, Rearranger)

```
    iMOAB_SendElementTag (sendmoabid, 'tag1:tag2', comm, context)
    iMOAB_RecvElementTag (recvmoabid, 'tag1:tag2', comm, context)
    iMOAB_FreeSendBuffers(sendmoabid, context)
```

MCTsMatAvMult(SrcAV, SparseMatrix, TrgAV)

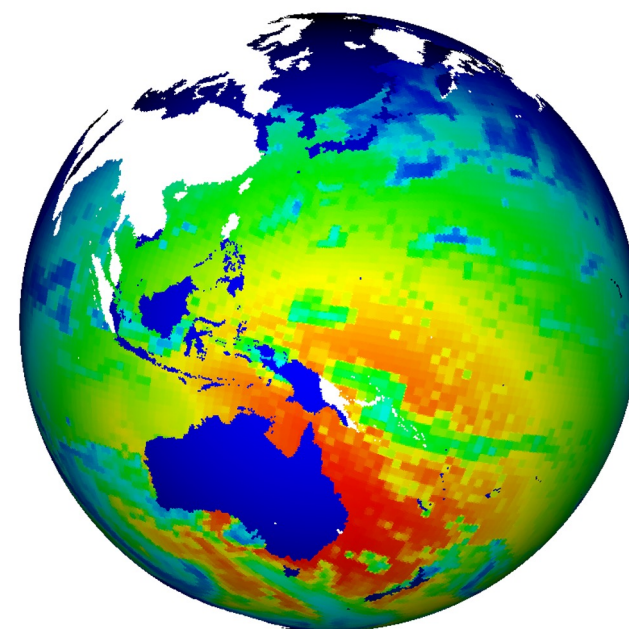
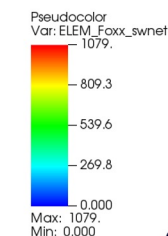
```
    iMOAB_ApplyScalarProjectionWeights(mbintxid, wtype, Srclist, Trglist)
```

Not in MCT

```
iMOAB_LoadMappingWeightsFromFile - read mapping weights in parallel
iMOAB_GetMeshInfo - return number of vertices, elements, other info
iMOAB_WriteMesh(moabid, filename, wopts) - write out mesh and all tags in h5m
                                         (HDF5) format in parallel
iMOAB_ComputeMeshIntersectionOnSphere(sourceid, targetid, intxid)
iMOAB_ComputeScalarProjectionWeights(intxid, . . . )
```

## Latest Status

- Basic coupled model (watercycle case) works!
  - Models and meshes:
    - EAM, ELM – on spectral mesh
    - MPAS-Seaice, MPAS-Ocean – on MPAS mesh
    - MOSART – on RLL mesh
    - Stub models for wave, land ice, iac
  - Atm-ocean, land-river, atm-river weights calculated online
  - River-ocean weights read from file.



Snapshot of merged net  
shortwave down

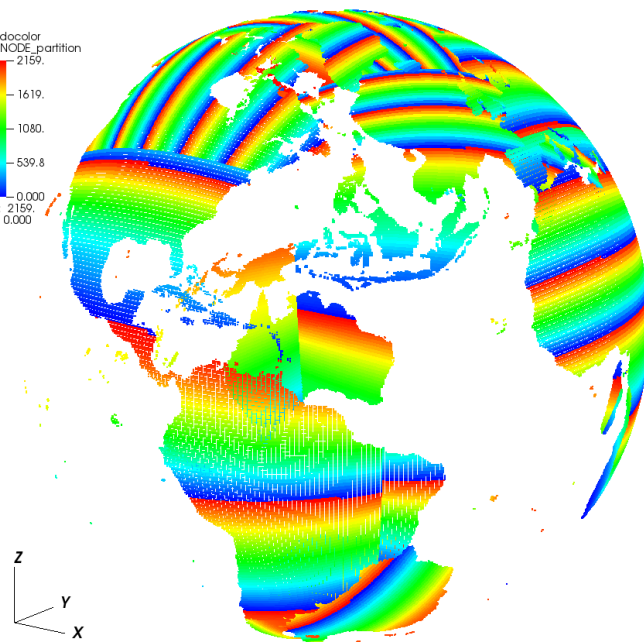


## Near future plans for cpl7/MOAB

- Hook up remaining models to MOAB coupler:
  - Data models, MPAS-land-ice, WW3.
- Additional online mapping options
  - (TempestRemap bilinear, others from SciDAC-CANGA)
- Remove MCT “scaffolding” from cpl7/MOAB
- Improve documentation
- Performance tuning
- Release with E3SM version 3 as an option.
- driver-mct will remain supported for bug fixes, porting

DB: wholeLnd.h5m

Pseudocolor  
Var: NODE\_partition  
-2159.  
-1619.  
-1080.  
-539.8  
Max: 0.000  
Min: 0.000



ELM decomposition from ELM MOAB instance