DART-X: Software Infrastructure for Prototyping in-memory Data Transfer between Ensemble Data Assimilation and Coupled Earth Systems Models

Anh Pham[1,2], Suman Shekhar[1,3], Helen Kershaw[1], Dan Amrhein[1], Ufuk Turuncoglu[1]

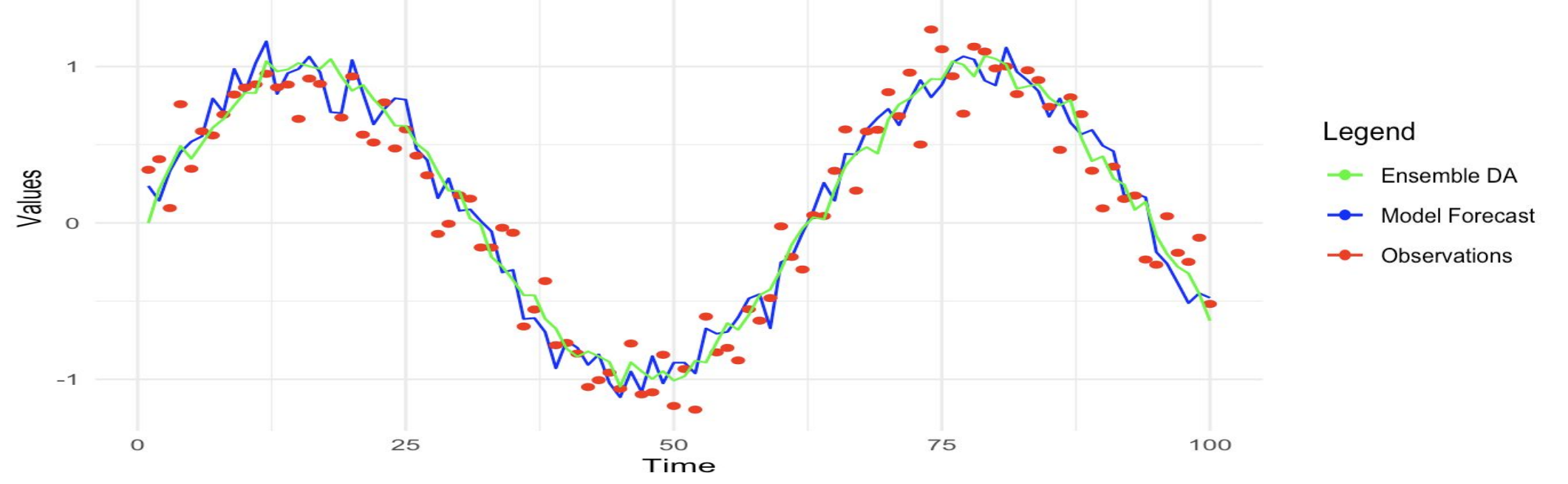
[1]NSF National Center for Atmospheric Research, [2]Mount Holyoke College, MA, [3]Rutgers University, NJ,
pham36a@mtholyoke.edu

1. BACKGROUND

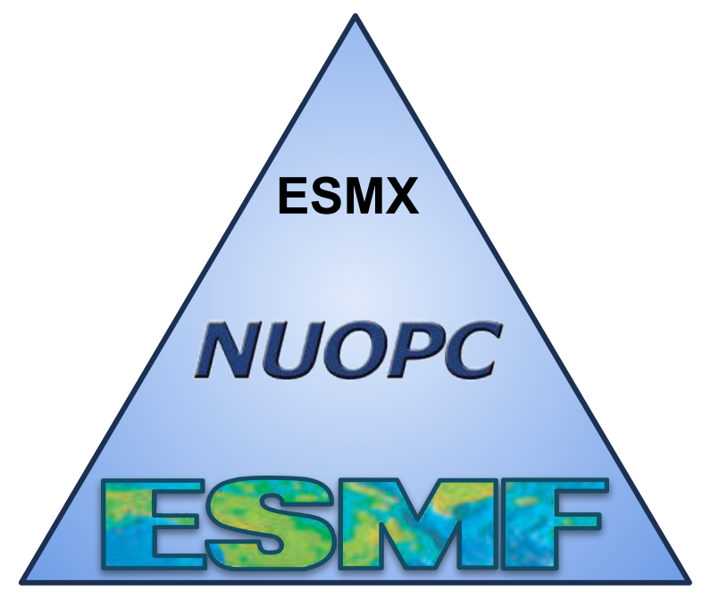
Data Assimilation, DART & CESM

Data assimilation (DA) integrates real-world data into climate models to capture the complexity of the Earth system and enhance prediction accuracy.

NCAR's **DART**, an ensemble DA software used with **CESM** (Community Earth System Model), balances model predictions and real-world observations to minimize uncertainties and represent a more accurate climate state.



ESMF, NUOPC, ESMX

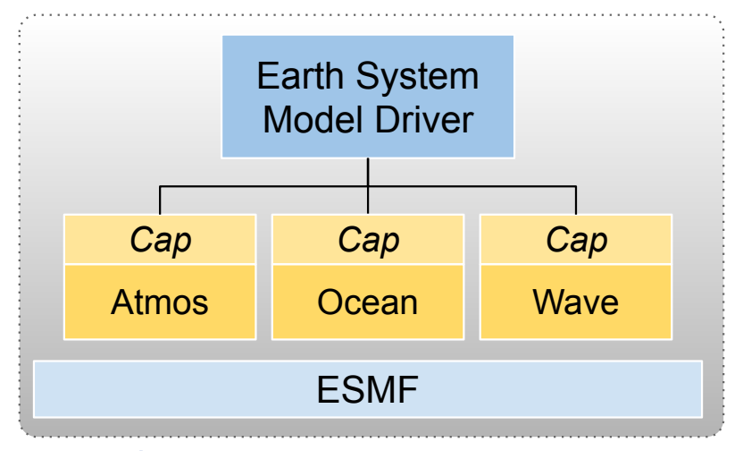


Software Stack

CESM uses **ESMF** (Earth System Modeling Framework) to couple climate models, simulating interactions between Earth components.

NUOPC (National Unified Operational Prediction Capability) **cap** is a software layer that standardizes data sharing within ESMF.

ESMX (Earth System Model eXecutable) is a layer built on top of ESMF and NUOPC APIs to simplify building, running, and testing NUOPC-based systems.

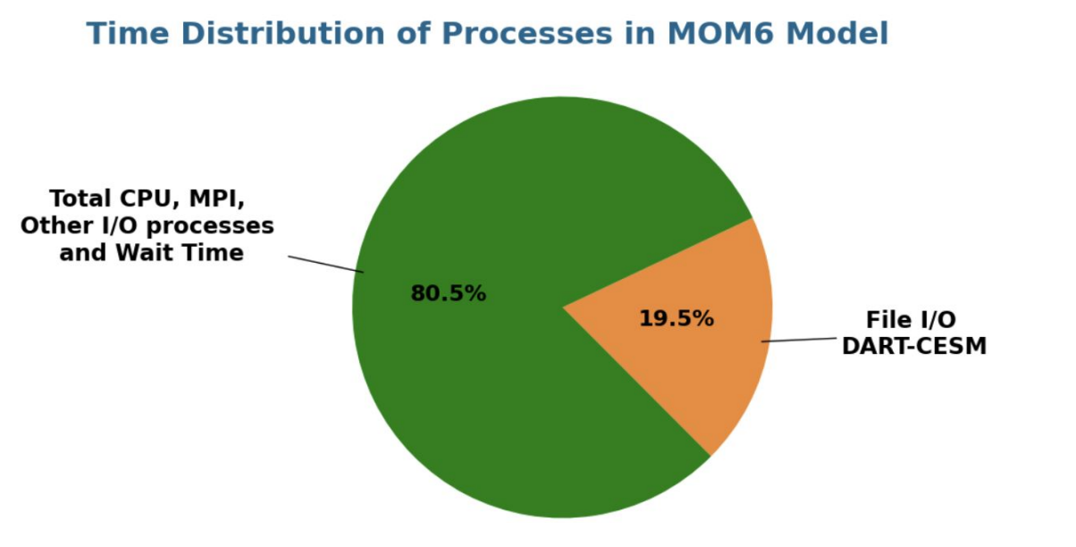


Coupling infrastructure
(includes the NUOPC Cap)

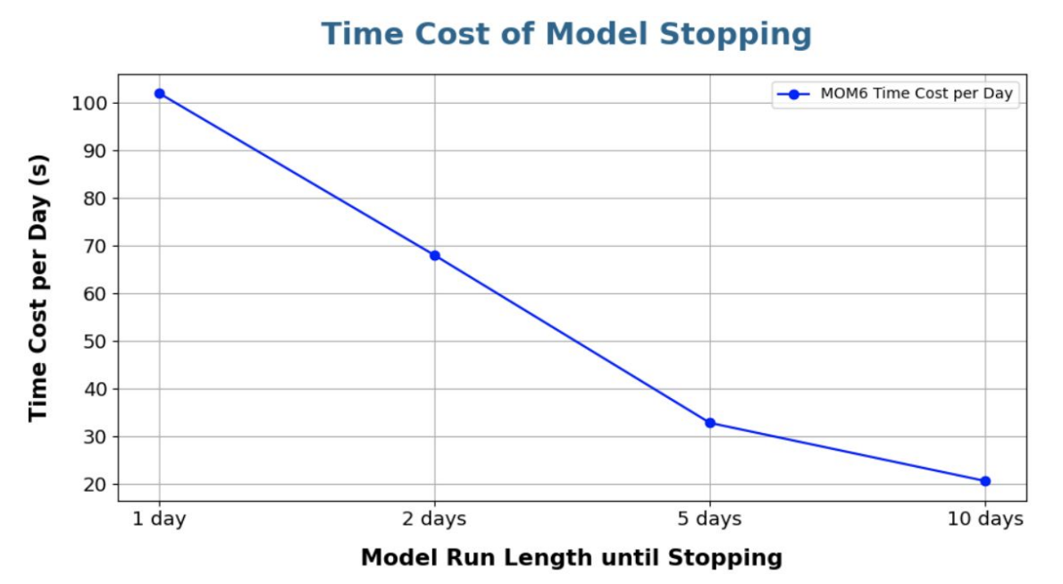
2. MOTIVATION

DART-CESM I/O Bottleneck

DART traditionally modifies "restart" files written into disk to update model states, requiring CESM models to stop for data assimilation and then restart. This process causes significant I/O bottleneck, leading to high computational costs, even on large supercomputers, especially for high-resolution models.



Benchmark profiling shows read/write operations consume significant runtime, even at relatively low-resolution (⅔ deg).

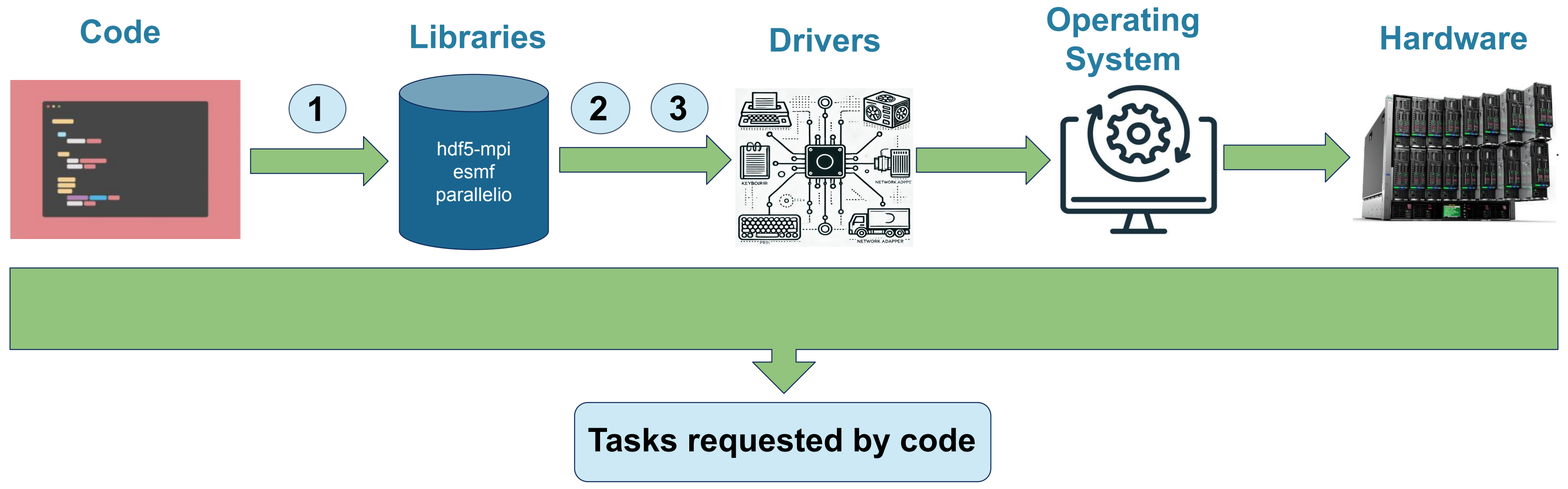


The more frequently we stop the model, the higher the fractional time cost. DA typically stops at least once per day.

3. OBJECTIVES

- Build infrastructure to develop and test a **DART-NUOPC cap** to allow DART to access CESM model states **directly in memory**, avoiding an I/O bottleneck from traditional file system data transfer and reducing resource waste.
- Validate the cap's functionality by passing 2D fields (data) in memory.

4. SOFTWARE INFRASTRUCTURE



- A code extension that provides a platform to coordinate, optimize and integrate distinct software components.
- 1, 2, 3 indicates the corresponding steps in the build process in section 6.

5. INFRASTRUCTURE CHALLENGES

- The sophisticated ESMF infrastructure, with 2 million lines of code, requires extensive communication with external labs, exceeding time and resource constraints.
- Building the interface requires integrating DART and CESM with minimal disruption to their core structure and existing functionalities.
- Integrating systems with varying object forms, infrastructures, and data formats is complex.

6. THE BUILD PROCESS (DART-X)

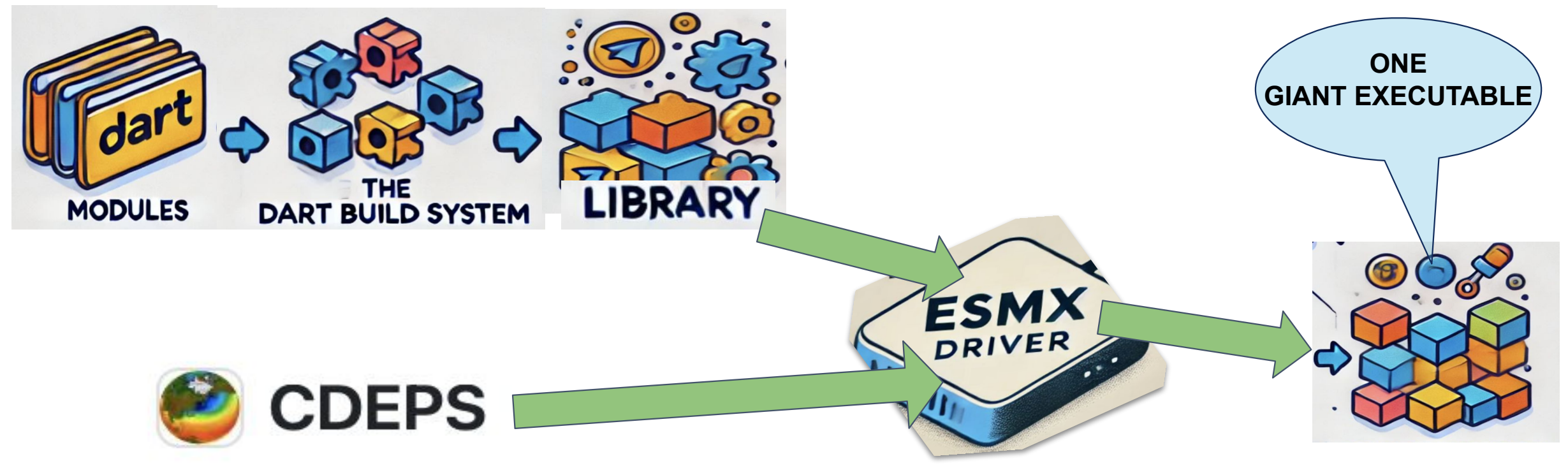
DART-X Build Process

1. Select environments and frameworks

- ➢ Build the cap on NCAR's Derecho supercomputer for its robust ESMF and NUOPC support.
- ➢ Utilize **ESMX** to simplify NUOPC-based development, handle orchestration, and isolate cap testing.
- ➢ Use **CDEPS** for targeted testing without a fully coupled system model to speed up development.
- ➢ Ensure build dependency compatibility for High Performance Computing (HPC) with build templates.

2. Transform and Unify Objects

- ➢ ESMX expects input components to be compiled as modules or libraries for a unified executable.
- ➢ However, DART outputs executables by default.
- ➢ DART-ESMX integration (DART-X): Modify DART to function as a library to incorporate into ESMX.



3. Build Configuration

- ➢ ESMX helps avoid extensive Fortran coding, streamlining compilation and linking via the human-readable YAML language.
- ➢ In the esmxBuild.yaml configuration file, locate DART in memory by pointing to its address and include the cap code as a module.

DART as a component

```
dart:
  build_type: script
  build_script: quickbuild.sh
  source_dir: /glade/u/home/sumansh/Altrun/DART/models/cdeps/work
  install_prefix: /glade/u/home/sumansh/Altrun/DART/models/cdeps/work
  fort_module: dart_comp_nuopc.mod
```

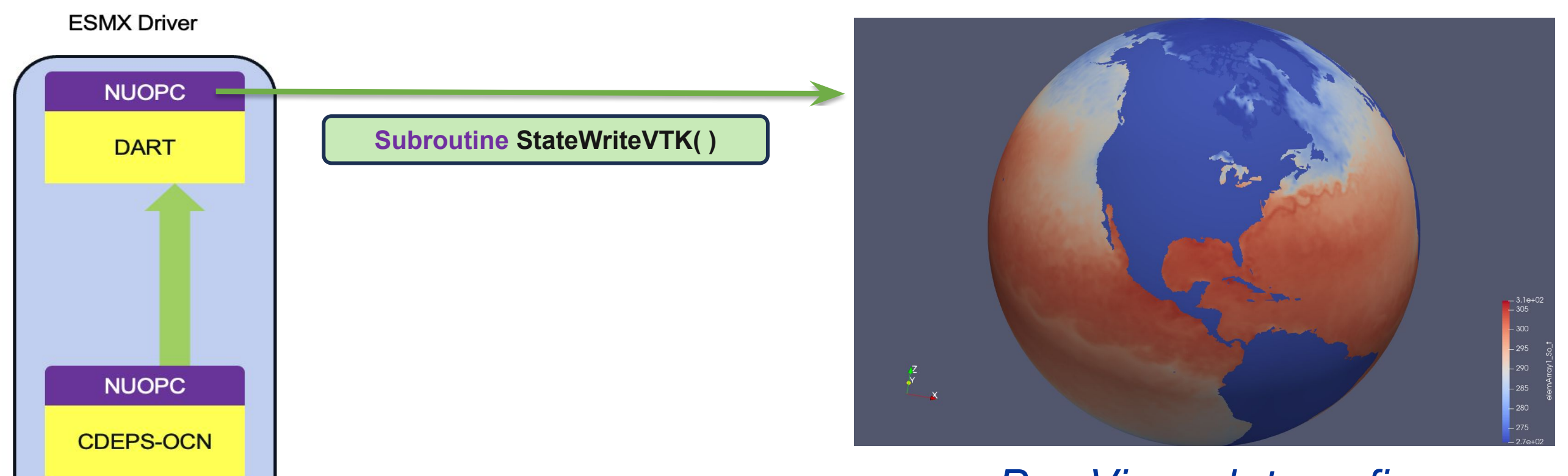
DART-NUOPC cap

7. RESULTS

First in-memory DART ⟺ CESM

Using DART-X, the DART-NUOPC cap successfully transfers 2D field Sea Surface Temperature (SST) from the ocean model (OCN) to DART.

The Model Advance subroutine in the DART-NUOPC cap writes the SST field to a VTK file, validating in-memory transfer.



ParaView plot confirms DART receives the 2D field

With the DART-NUOPC Cap integrated, DART advances in time like a model component.

8. CONCLUSIONS

- Developing robust software infrastructure minimizes disruptions and enables new feature development and testing. DART-X supports the DART-NUOPC Cap, the first in-memory data transfer prototype between DART and CESM.
- The first DART-NUOPC cap prototype paves the way to explore direct memory sharing, potentially reducing disk I/O bottlenecks. This enables DART to act as a model component, speeding up data assimilation and reducing computational costs.

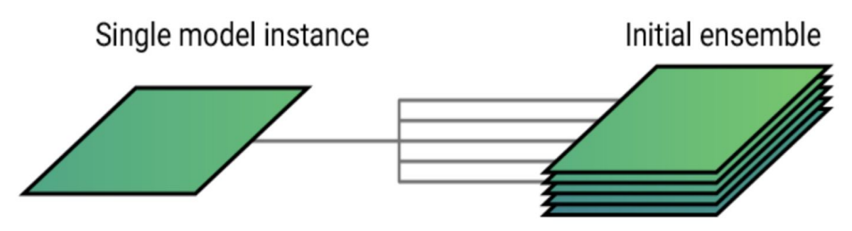
9. FUTURE WORKS

Full DART-ESMF integration

Advance from a successful ESMX prototype, DART-X, to a full DART-CESM cap using ESMF driver.

Ensemble Field Transfers

Continue testing DART as a model component for ensemble models to transfer ensemble fields.



Profiling: Memory vs. Disk Improvement

Profile: quantify performance improvement for DA-model with in-memory data transfer versus traditional disk transfer.

Handling Missing Fields In Memory

The cap currently exchanges fields; future work will address missing data in DART-CESM communication.

Generalizing in-memory data passing

In-memory data passing techniques in this prototype may extend to other data systems and ML frameworks.

REFERENCES AND ACKNOWLEDGEMENT

Thanks to my project partner, Suman Shekhar, for his dedication and collaborative spirit. I am grateful to my mentors Dan Amrhein, Helen Kershaw, and Ufuk Turuncoglu for their expertise, vision, and generous support throughout the project. I deeply appreciate the support from SIParCS Program Director Virginia Do and admins for making this internship a fruitful experience. Thank you to my fellow interns for the opportunity to connect and share knowledge. Special thanks to the DAReS team: Jeffrey Anderson, Moha Gharamti, Kevin Raeder, Helen Kershaw, Marlee Smith, Ben Johnson, Ann Norcio for their technical support on DART and the ESMF team: Dan Rosen, Ann Tsay, Jim Edwards, Bill Sacks, Ufuk Turuncoglu for their technical support on ESMF. Thanks to NCAR, UCAR, CISL, and NSF for funding and facilities.

DART ESMF