Improving the Speed and Scalability of the Data Assimilation Research Testbed

Jiachen (Ed) Liu[1], Helen Kershaw[2], Jeffrey Anderson[2]
[1]Department of Civil, Architectural, & Environmental Engineering, Drexel University
[2]Data Assimilation Research Section (DAReS), National Center for Atmospheric Research

INTRODUCTION

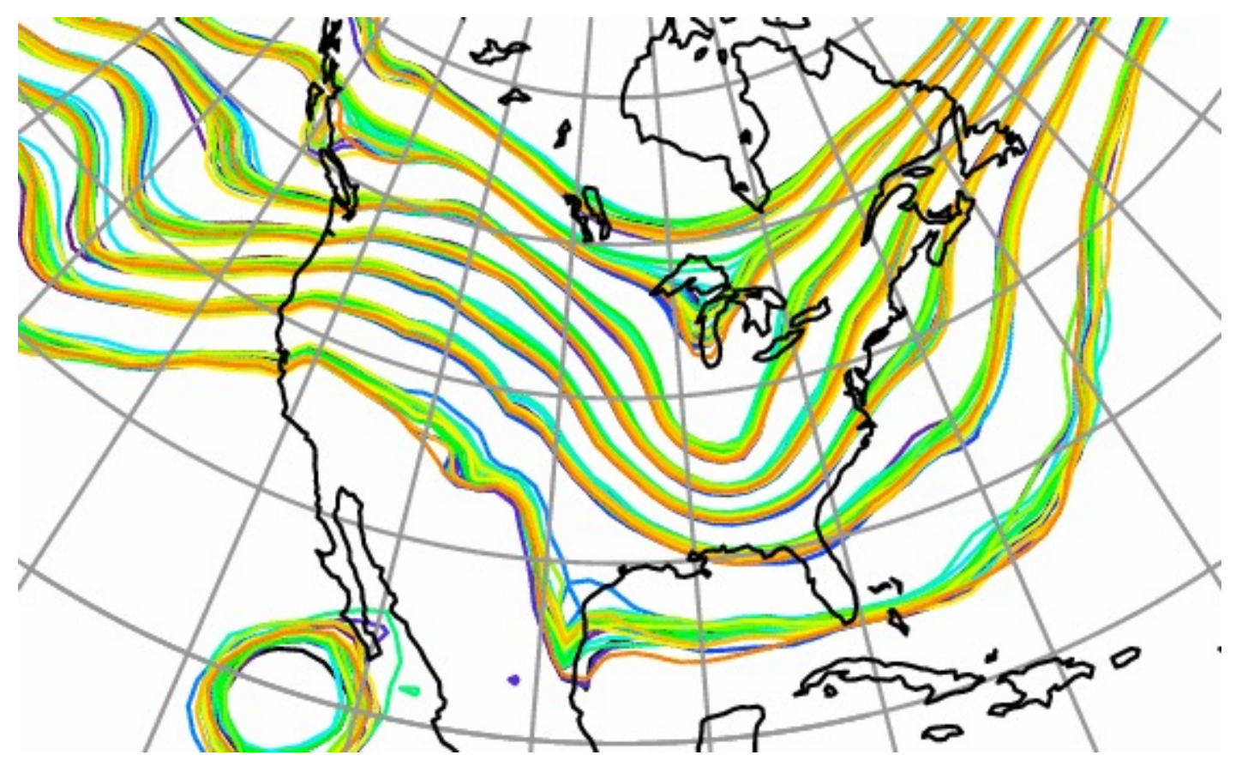
Ensemble Data Assimilation

Ensemble data assimilation is a method developed to combine model forecasts with observations to generate better forecasts.

For example, ensemble data assimilation can be used to improve weather predictions. If we have a group of different forecasts of the temperature and observations made at different time and locations, we can use ensemble data assimilation to produce a better estimate of the temperature predictions.

Data Assimilation Research Testbed

The Data Assimilation Research Testbed (DART) was developed by the Data Assimilation Research Section (DAReS) to help researchers conduct ensemble data assimilation with different models.



OBJECTIVES

Improving the Speed and Scalability of DART

In this work, we aim to improve the speed and scalability of DART.

The first step is to conduct code profiling of DART to identify the computational barriers within the code. The second step is to improve the speed and scalability of DART through algorithmic changes based on the profiling results.

RESULTS

Initial Code Profiling Results

- We conducted code profiling of DART with different models using arm-forge MAP, a profiling tool for parallel C++ or Fortran programs available on Cheyenne.
- The results of a profiling of the Finite Volume Community Atmosphere Model (CAM-FV) are shown to the below in Table 1.

# of nodes	# of processors	Runtime from MAP [s]	filter_mod, compute (%)	filter_mod, mpi (%)	mpi_utilities (%)
2	36	2401.83	70	26.2	3.8
4	36	1071.377	44.5	47.1	8.4
8	36	658.034	24.5	61.8	13.6
10	36	339.992	39.2	34.1	26.6
20	36	285.397	25.8	41.5	32.7
10	4	1184.76	85.2	7.3	7.5
10	16	446.941	60.5	19.4	20

Table 1. Profiling results with different number of processors and nodes for CAM-FV

Identification of Redundant Caching

- The results of profiling the Atmospheric Component of Model for Prediction Across Scales (MPAS-ATM) exposed a computational barrier shown below.

```
last_close_state_ind(:)  = close_state_ind(:)
last_close_state_dist(:) = close_state_dist(:)
```

- These two lines of array copying **consume almost 40% of the total runtime** of DART. Removing these two lines reduced the computation time from **260 seconds to 64 seconds**.
- Further investigations show that these two lines are unnecessary and can be removed.

Reorganizing the input files to improve speed and scalability

Problem Description

- The MIT General Circulation Model for the ocean (MITgcm-ocean) is a numerical model that can compute parameters related to the ocean.
- DART cannot run due to memory overflow.
- The grid has land (shown in Figure 1 to the right) which is not used in the data assimilation process.
- In the state file, these values are usually fill values. Analysis shows **92%** of the grid are fill values for this case.

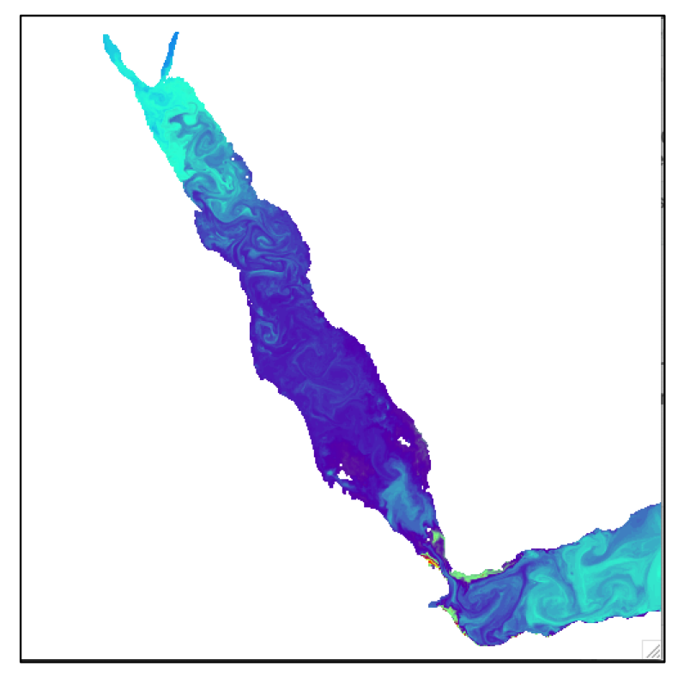


Figure 1. MITgcm-ocean output over the red sea. The white areas are all fill values

Original Approach in DART

- An example of the original approach is shown in Figure 2 below.
- DART reads everything in a large 1-D vector, with all the fill values.
- The missing values (land in an ocean model) stay in the vector.

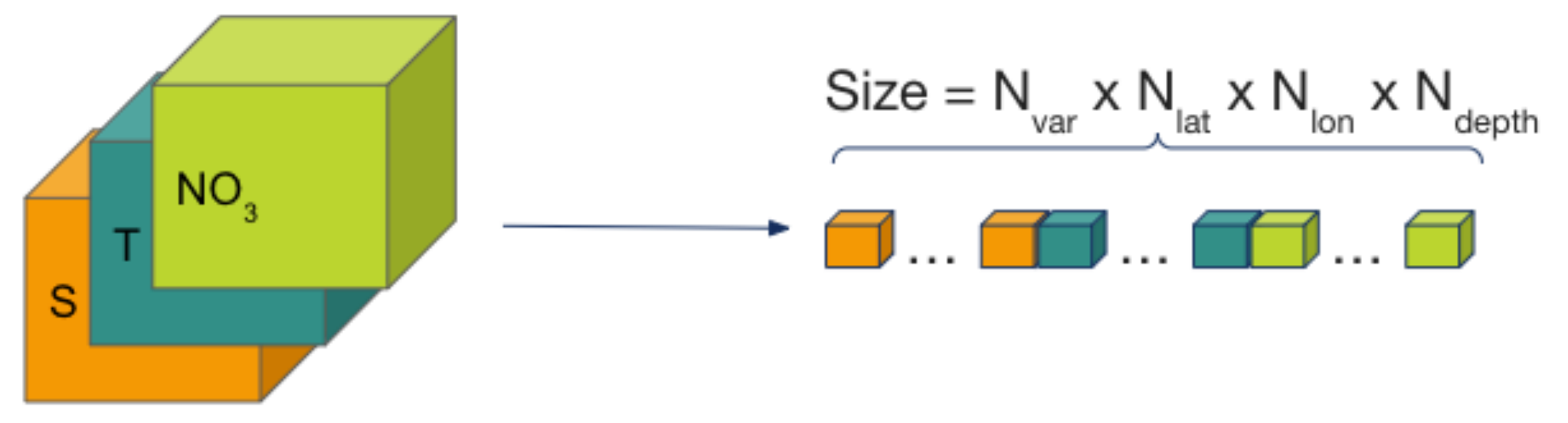

NO_3
T
S
Size = $N_{var} \times N_{lat} \times N_{lon} \times N_{depth}$

Figure 2. Scheme for the offline preprocessing of the DART input state file with fill values.

The "Squished State" Approach

- Preprocessing can reduce the input file sizes by squishing out all the fill values.
- The new input file does not have any missing values, and all new vectors are reduced to 1-D.
- The squishing process removes all the dimension information from the file, so latitude, longitude, depth are added as new 1-D arrays.
- For this specific case, the input file size is reduced **from 10.4 GB to 988 MB**.
- DART can now run on this large case, and runtime for a smaller case is significantly reduced. Results are shown in Table 2 below.

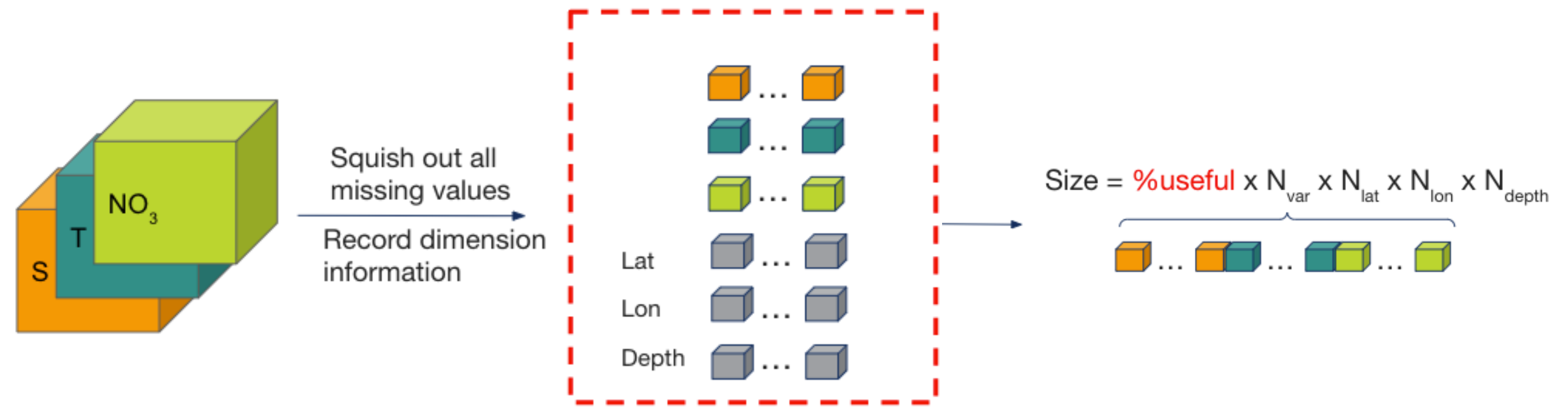

NO_3
T
S
Squish out all missing values
Record dimension information
Lat
Lon
Depth
Size = %useful $\times N_{var} \times N_{lat} \times N_{lon} \times N_{depth}$

Figure 3. Scheme for the squishing of the DART input state file with fill values.

	Medium Case (500x500x50)	Large Case (2000x2000x50)
Original	361s	N/A
Squished State	150s	1500s

Table 2. Performance improvement from the squished state approach

ACKNOWLEDGEMENTS

I would like to thank my mentors Helen Kershaw and Jeff Anderson for their support. I would also like to thank the entire DAReS team, the CODE team, and CISL help desk who provided help during the summer.