CESM Workflow Refactoring

Sheri Mickelson (mickelso@ucar.edu)

Kevin Paul, Alice Bertini , John Dennis, Haiying Xu,

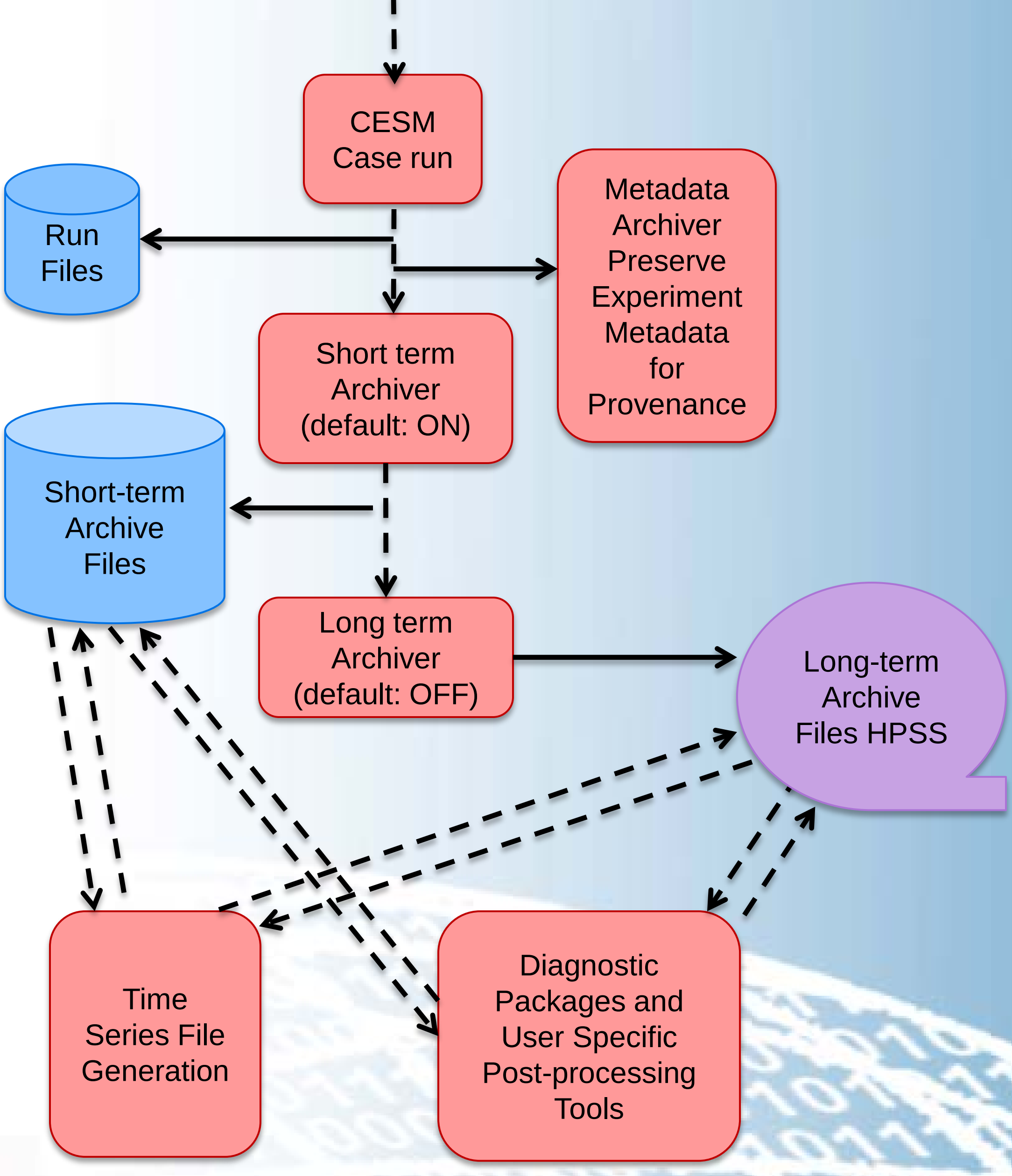
Dave Brown, Mary Haley, Jim Edwards,

Mariana Vertenstein

CISL Computational & Information Systems Laboratory

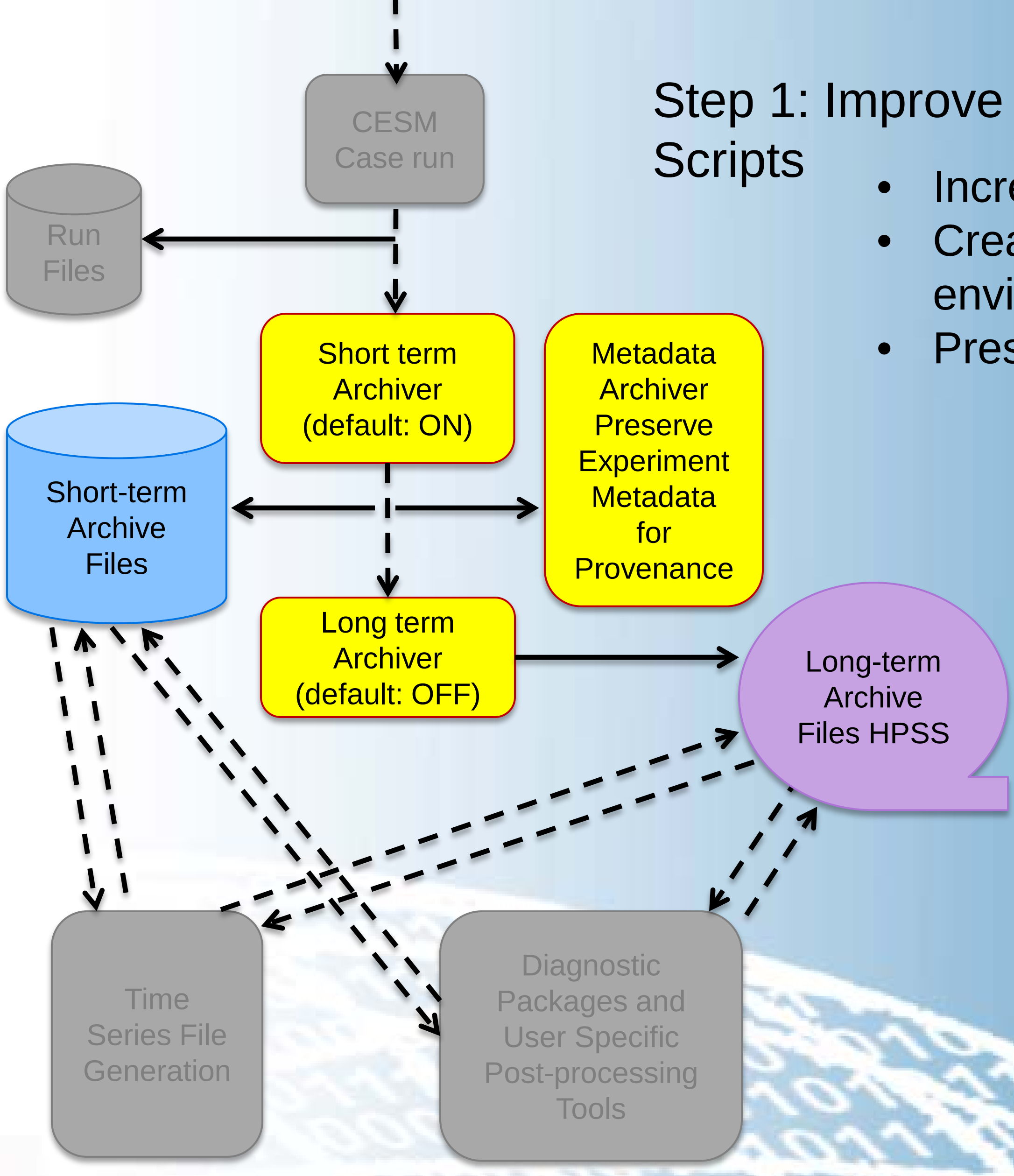
NCAR

Current Workflow



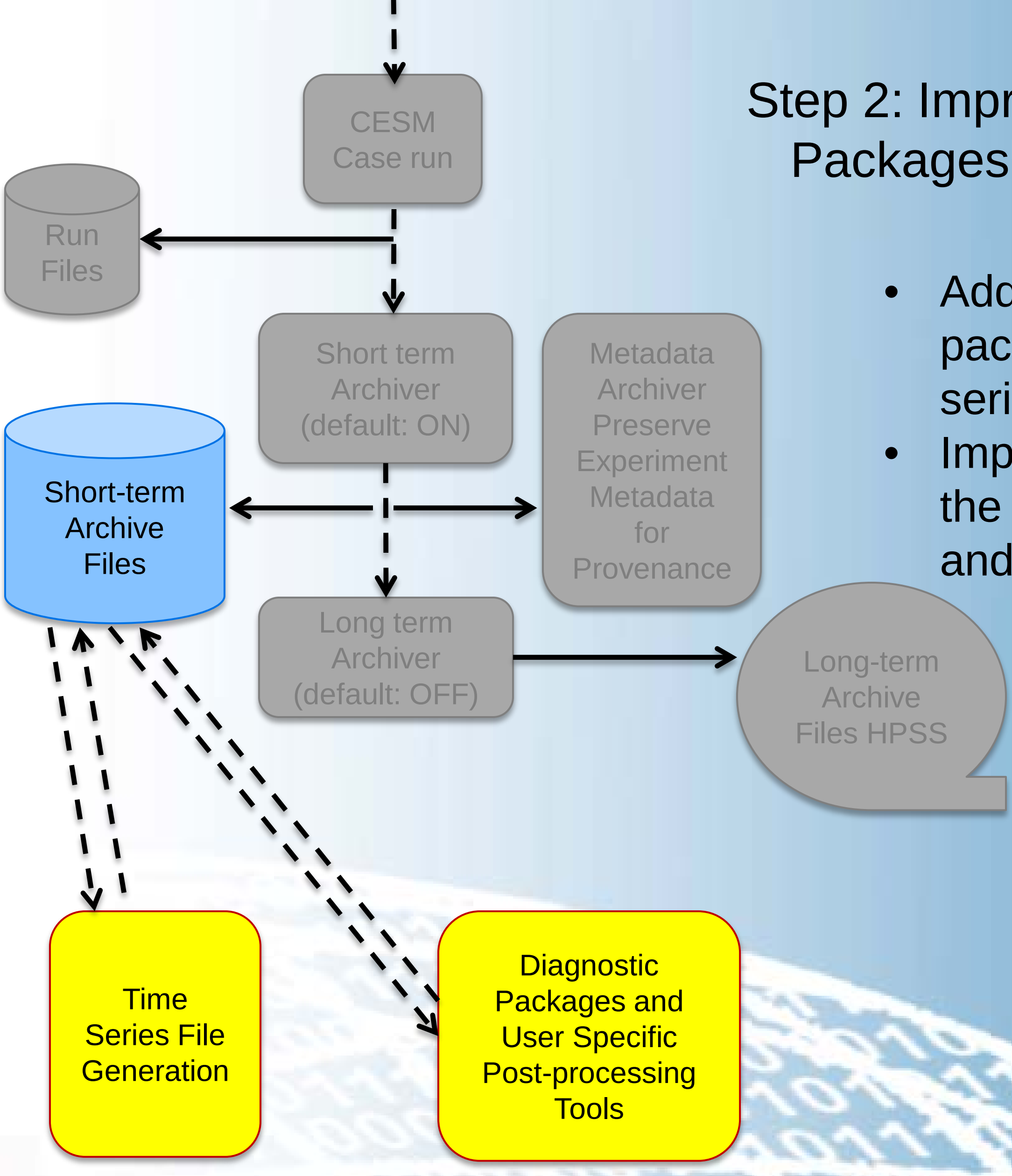
Step 1: Improve the Archiving Scripts

- Increase robustness
- Create a more flexible environment
- Preserve data integrity

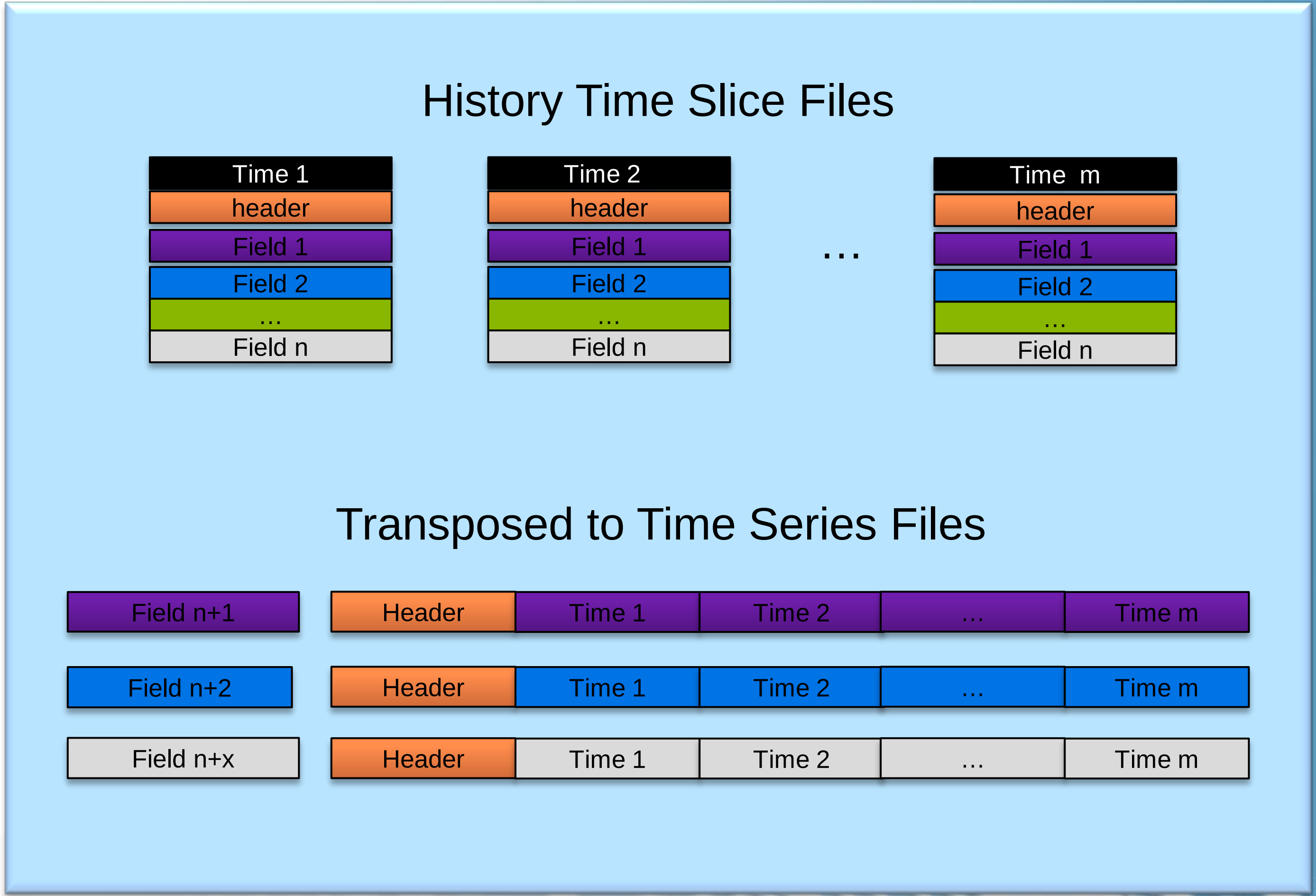


Step 2: Improving the Diagnostic Packages and the Time Series Generation Tool

- Add ability for the diagnostic packages to read time series files
- Improve the performance of the diagnostic packages and the time series script


CESM Case run
Run Files
Short term Archiver (default: ON)
Metadata Archiver Preserve Experiment Metadata for Provenance
Short-term Archive Files
Long term Archiver (default: OFF)
Long-term Archive Files HPSS
Time Series File Generation
Diagnostic Packages and User Specific Post-processing Tools

Time Series Generation Tool



Current Time Series Generation Tool

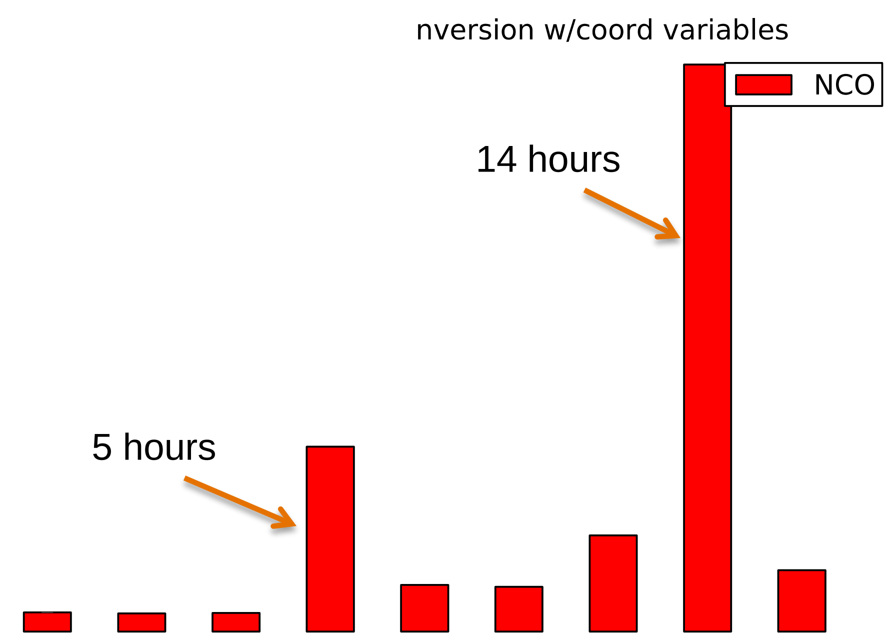
- CSH scripts
- Uses NCO tools to transpose the data

Problems with current tool:

- Lacks a flexible environment
- Runs in serial mode/no parallelization
- Takes a long time to run

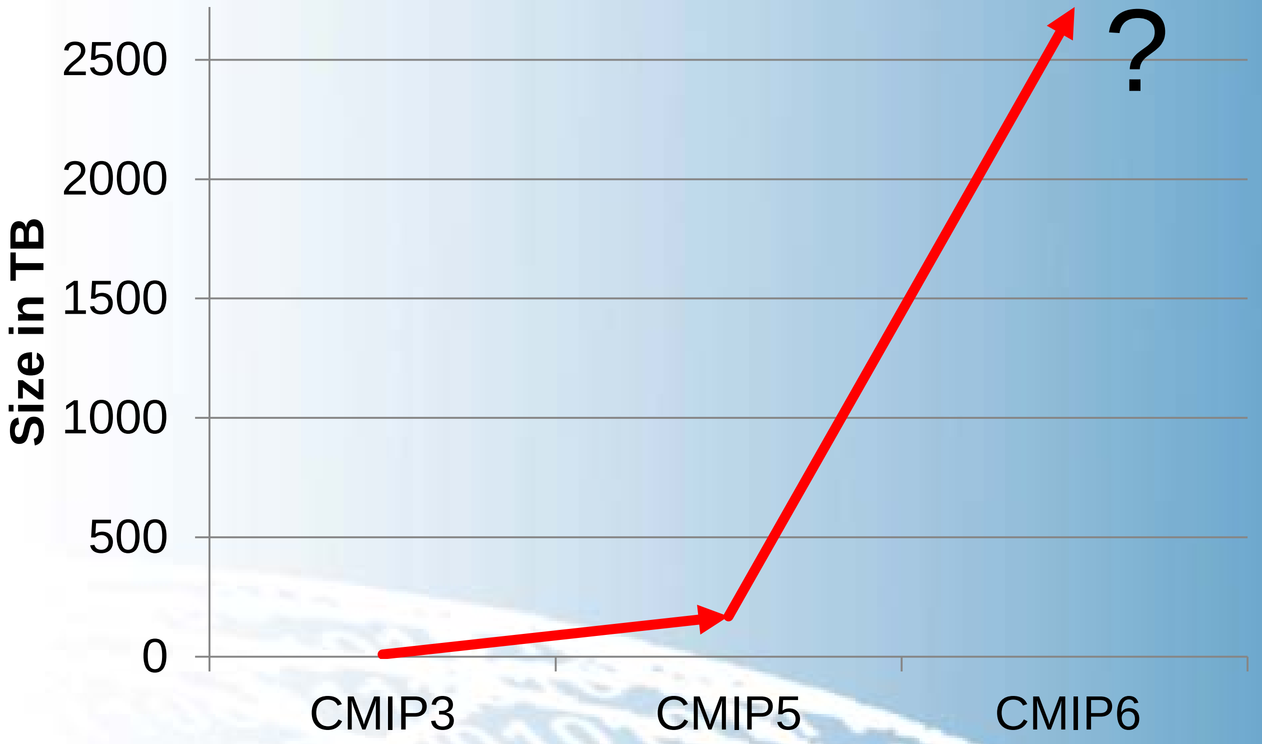
Current Time Series Generation Tool

Datasets – 10 yrs monthly history files	Size (Gbytes)
CAMFV-1.0	28
CAMSE-1.0	31
CICE-1.0	8
CAMSE-0.25	1077
CLM-1.0	9
CLM-0.25	84
CICE-0.1	570
POP-0.1	3184
POP-1.0	194



*** Comparing the time it takes to convert 10 years of monthly time slice data to time series data using the existing method

CESM/CMIP Archive Sizes

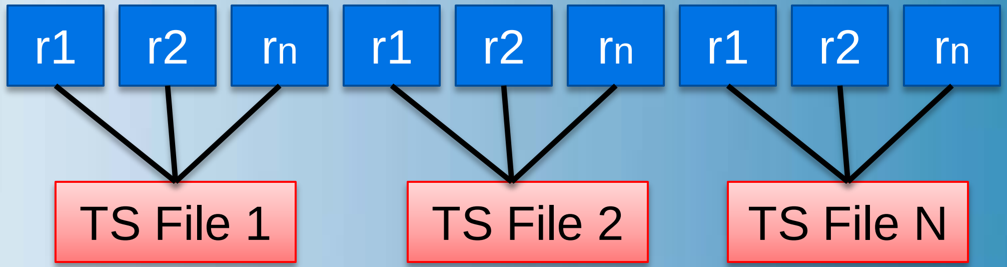


- It took 15 months to transpose the 170 TB of CESM data from time slice to time series

Approaches to Parallelism

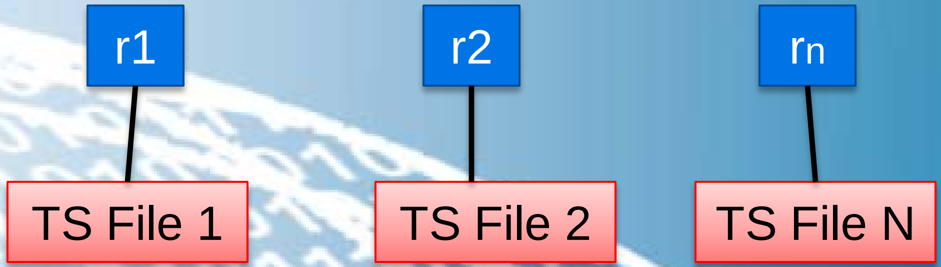
Data Parallelism:

- Divide a single variable across multiple ranks



Task Parallelism:

- Divide independent tasks across multiple ranks



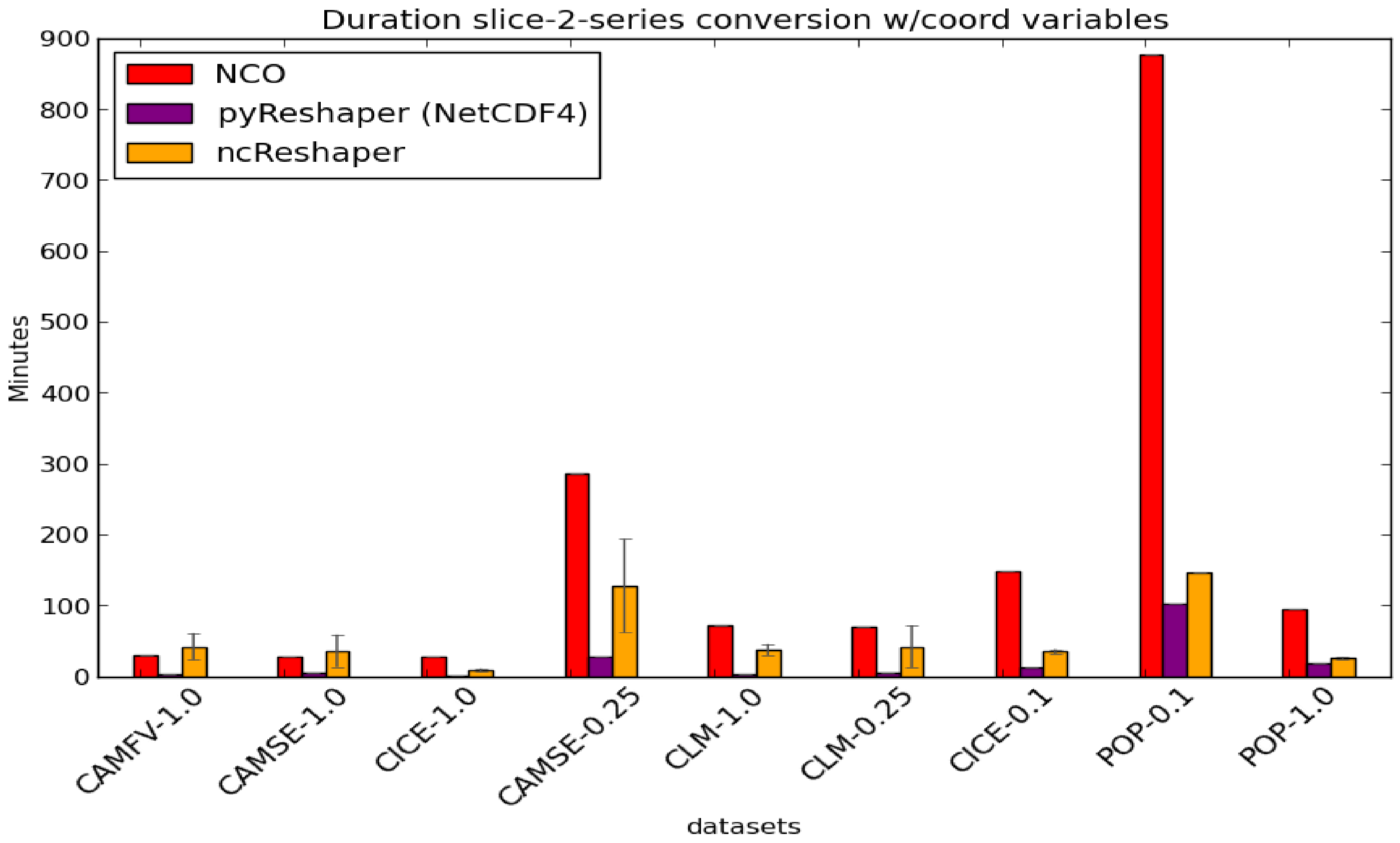
ncReshaper vs. pyReshaper

	ncReshaper	pyReshaper
Type of Parallelism	Data	Task
Code Specifications	Fortran MPI PIO – I/O Library in CESM	Python MPI4py pyNIO – NCL I/O Library

Experiment:

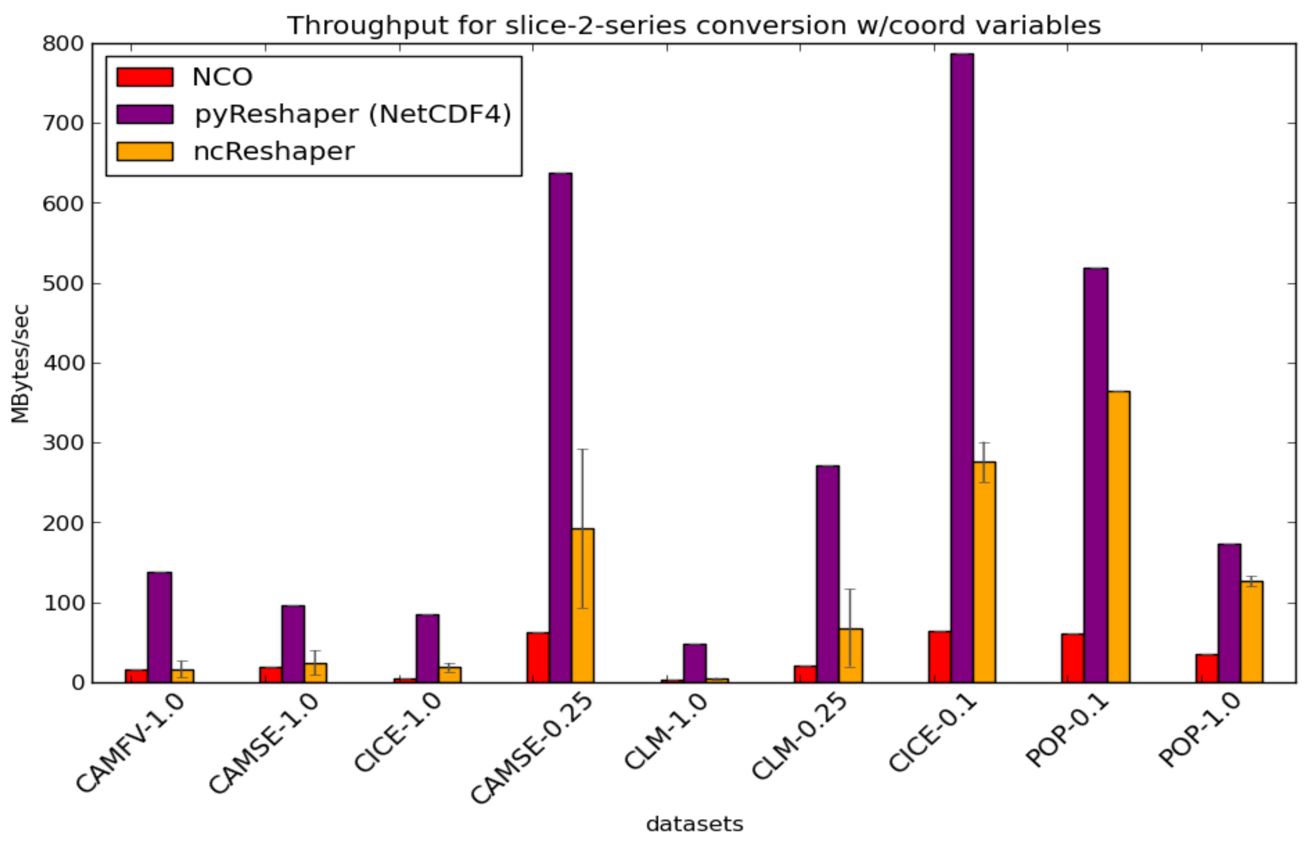
- Convert the 10 year datasets using both methods
- Compare the results

Duration



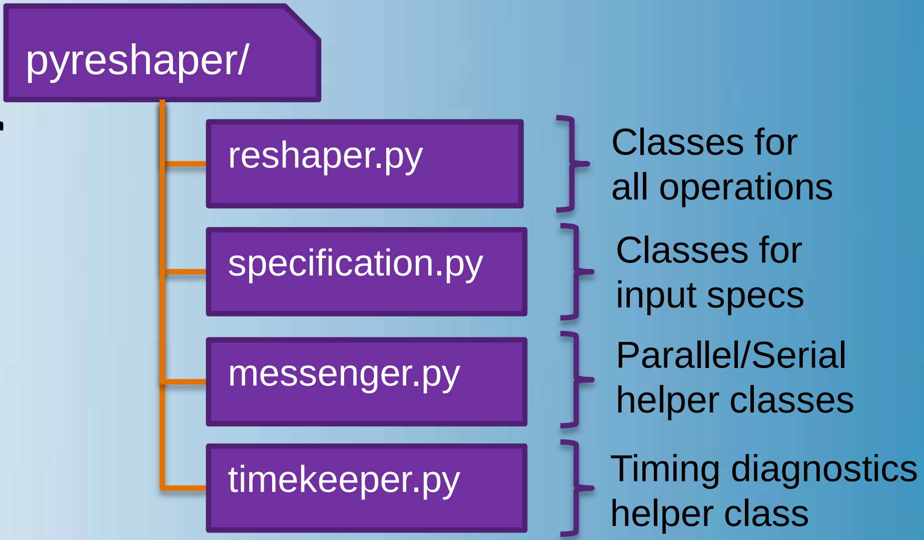
	NCO	ncReshaper	pyReshaper	Improvement
POP – 0.10	14 ½ hours	2 ½ hours	1 ½ hours	~ 8 X
CAMSE – 0.25	4 ¾ hours	2 hours	28 minutes	~ 10 X

Throughput Rates



Ongoing Work

Developing a release version of the pyReshaper


pyreshaper/
reshaper.py
Classes for all operations
specification.py
Classes for input specs
messenger.py
Parallel/Serial helper classes
timekeeper.py
Timing diagnostics helper class

- Easy installation (distutils)
- Simple usage:

```
from pyreshaper import reshaper
from pyreshaper import specification

spec1 = specification.create_specifier(spec_type="slice-to-series")
spec1.input_file_list = ["path/to/file1.nc", "path/to/file2.nc", ...]
spec1.output_file_prefix = "path/to/output/dir/prefix."
spec1.output_file_suffix = ".000101-001012.nc"
spec1.time_variant_metadata = ["time", "time_bound", ...]
…
rshpr = reshaper.create_reshaper([spec1, ...], serial=False)
rshpr.convert()
rshpr.print_diagnostics()
```

Ongoing Work

pyAverager

Problem: **To create yearly averages within the OMWG Diagnostic Package**	**Compute Time**
Time Slice: ncra ${CASENAME}.pop.h.${year}-??.nc ${CASENAME}.pop.h.${year}.nc	3 mins

Time Series:
foreach var ($var_list)
  foreach month (1 2 3 4 5 6 7 8 9 10 11 12)
    ncks -O -F -d time,$i,$i,1 TimeSeriesFile.nc temp_${yr}_${m_print}.$var.nc
  ncra -O temp_*.$var.nc   yearlyAve.$var.nc
foreach variable
  ncks –A yearlyAve.$var.nc   yearlyAve_${yr}_.nc
	40 mins

*** Compute time was the time it took to calculate 3 yearly averages in parallel for a 1 degree POP dataset w/ biogeochemistry variables added (230 variables/files) with 16 mpi tasks

New: Parallel python/pyNIO/numpy/mpi4py → 2 1/2 minutes

Summary

- We will be releasing the pyReshaper and new archiving tools within an upcoming CESM release
- We were able to speed up the process of converting time slice to time series by a least a factor of 8
- Looking to see if we're able to speed up other commonly run operations using similar methods

Thank you NSF for your support (grant #M0856145)

And thank you Gary Strand for the CMIP statistics.

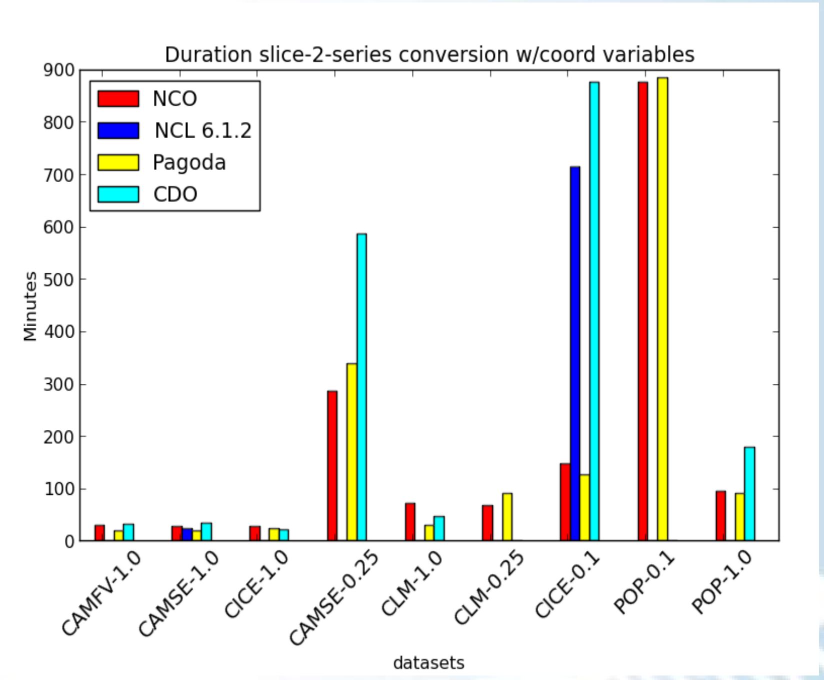
Extra

Dataset characteristics: 10-years of monthly output

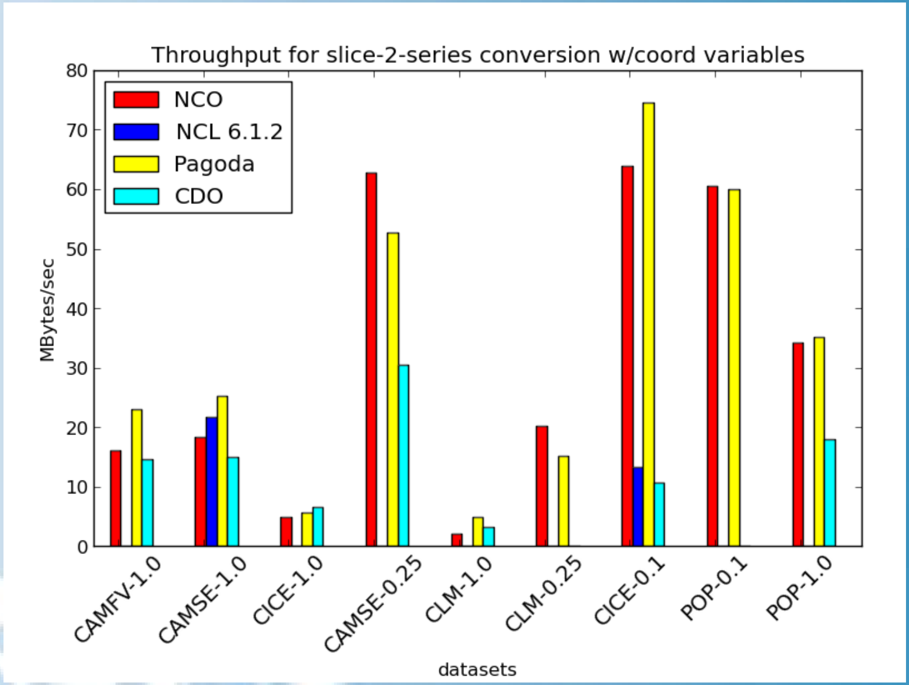
Dataset	# of 2D vars	# of 3D vars	Input total size (Gbytes)
CAMFV-1.0	40	82	28.4
CAMSE-1.0	43	89	30.8
CICE-1.0	117		8.4
CAMSE-0.25	101	97	1077.1
CLM-1.0	297		9.0
CLM-0.25	150		84.0
CICE-0.1	114		569.6
POP-0.1	23	11	3183.8
POP-1.0	78	36	194.4

Other methods that were tested

Duration



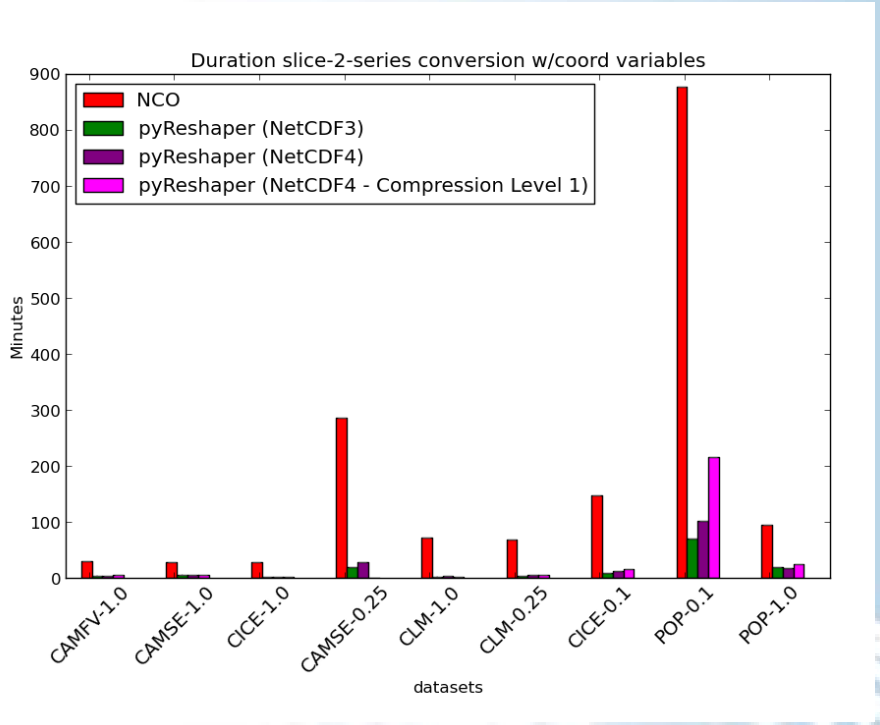
Throughput



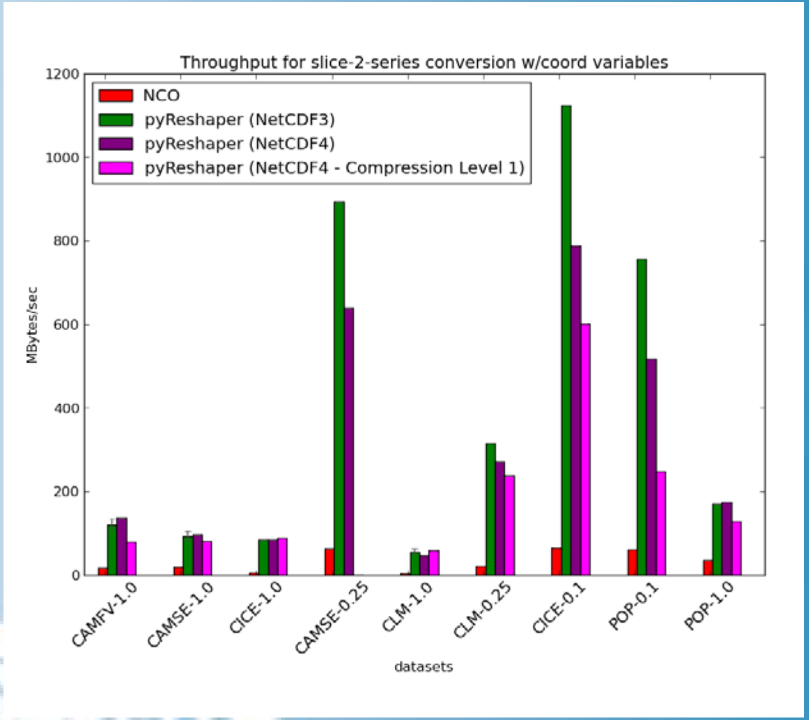
Computational & Information Systems Laboratory

PyReshaper testing results for different netCDF types

Duration



Throughput


Throughput for slice-2-series conversion w/coord variables
NCO
pyReshaper (NetCDF3)
pyReshaper (NetCDF4)
pyReshaper (NetCDF4 - Compression Level 1)
MBytes/sec
datasets
CAMFV-1.0
CAMSE-1.0
CICE-1.0
CAMSE-0.25
CLM-1.0
CLM-0.25
CICE-0.1
POP-0.1
POP-1.0