CESM CMIP6 Data Workflow

Sheri Mickelson
CISL/TDD/ASAP

2015, September 14

CESM Workflow Refactoring Team

- Ben Andre
- Alice Bertini *
- John Dennis
- Jim Edwards
- Mary Haley
- Jean-Francois Lamarque
- Michael Levy
- Sheri Mickelson *
- Kevin Paul *
- Sean Santos
- Jay Shollenberger *
- Gary Strand
- Mariana Vertenstein

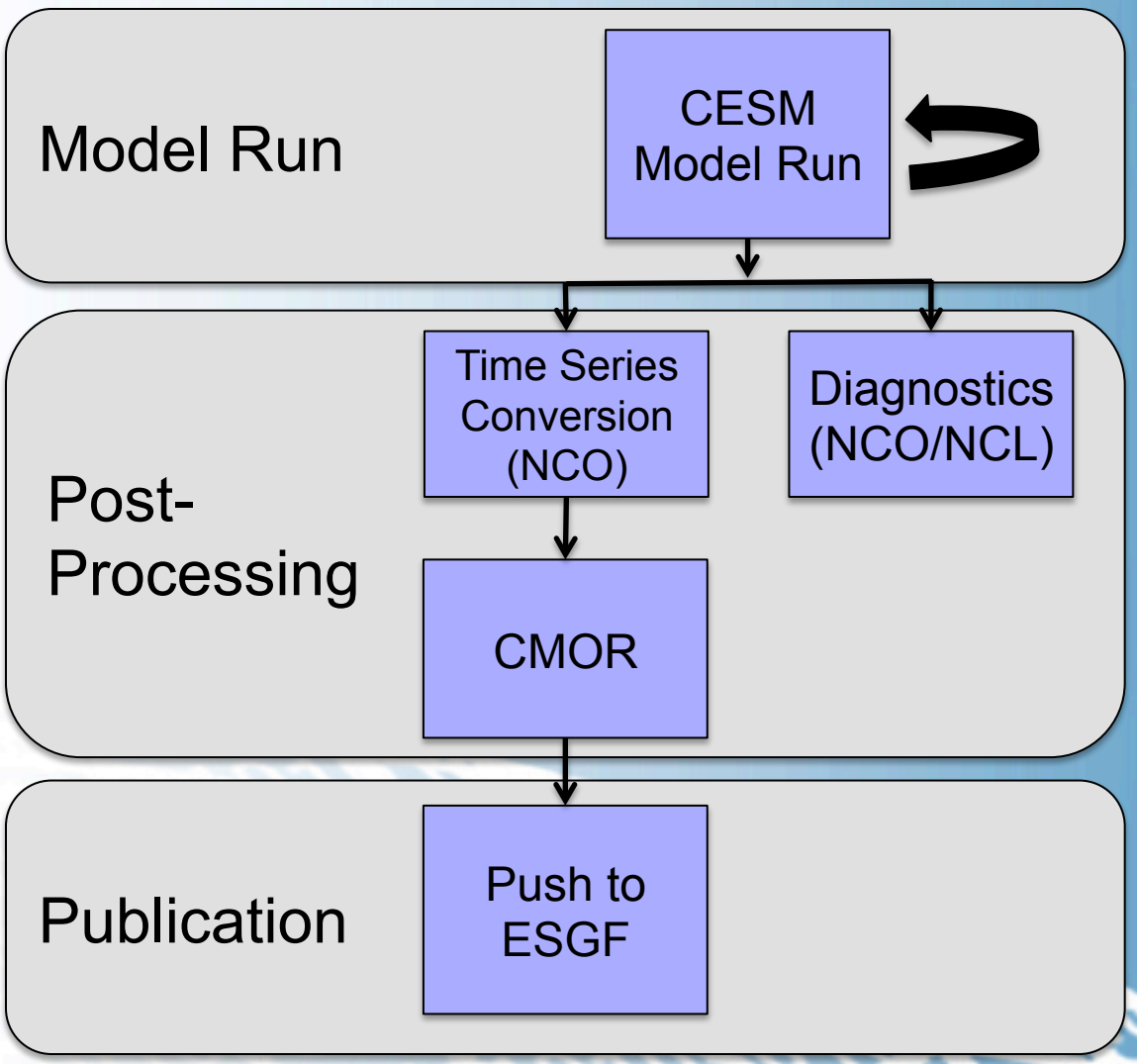
(* Developer)

NCAR UCAR | CESM COMMUNITY EARTH SYSTEM MODEL

CISL Computational & Information Systems Laboratory

NCAR

CMIP5 Workflow



Lessons We Learned From CMIP5

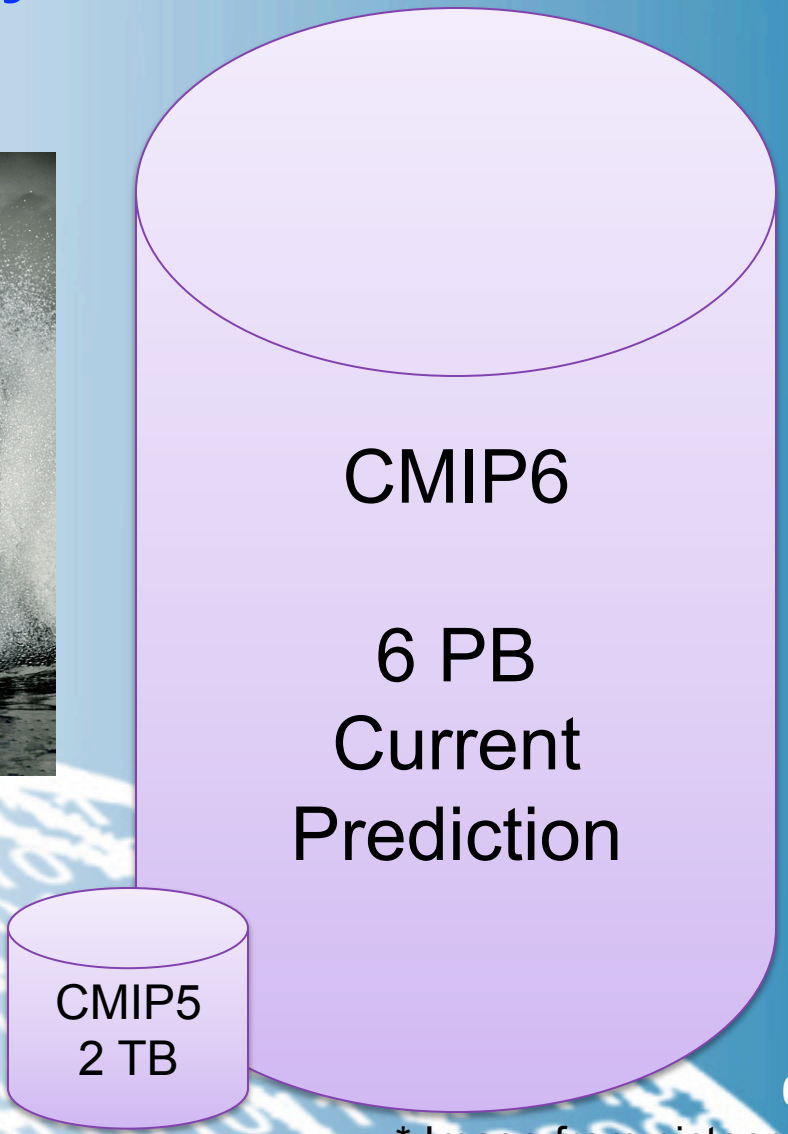
- **CESM was the first model to complete their simulations, but the last to complete publication. Why?**
  - All post-processing was serial
  - Workflow was error prone and took time to debug
  - Too much human intervention was needed between post-processing steps

Plans for CMIP6 (Preliminary)

- **Participate in 26 MIPS**
- **Low Resolution: 1º atm, 1º ocn**
  - Throughput: 14.5 simulated years per wall clock day
  - Cost: ~2,000 core hours per simulated year
  - Total estimated cost: 150 million core hours
- **High Resolution: 25km atm, 1º ocn**
  - Throughput: 1.84 simulated years per wall clock day
  - -Cost: ~215,000 core hours per simulated year
  - Total estimated cost: 350 million core hours
- **Data sizes:**
  - Raw size: ~12 PB
  - Published: ~6 PB = **we will have to process 5TB a day for 3 years**

To quote Jim Kinter, how are we going to post – process the flood of data from CMIP6?

Our current boat won't be able to process the flood of data

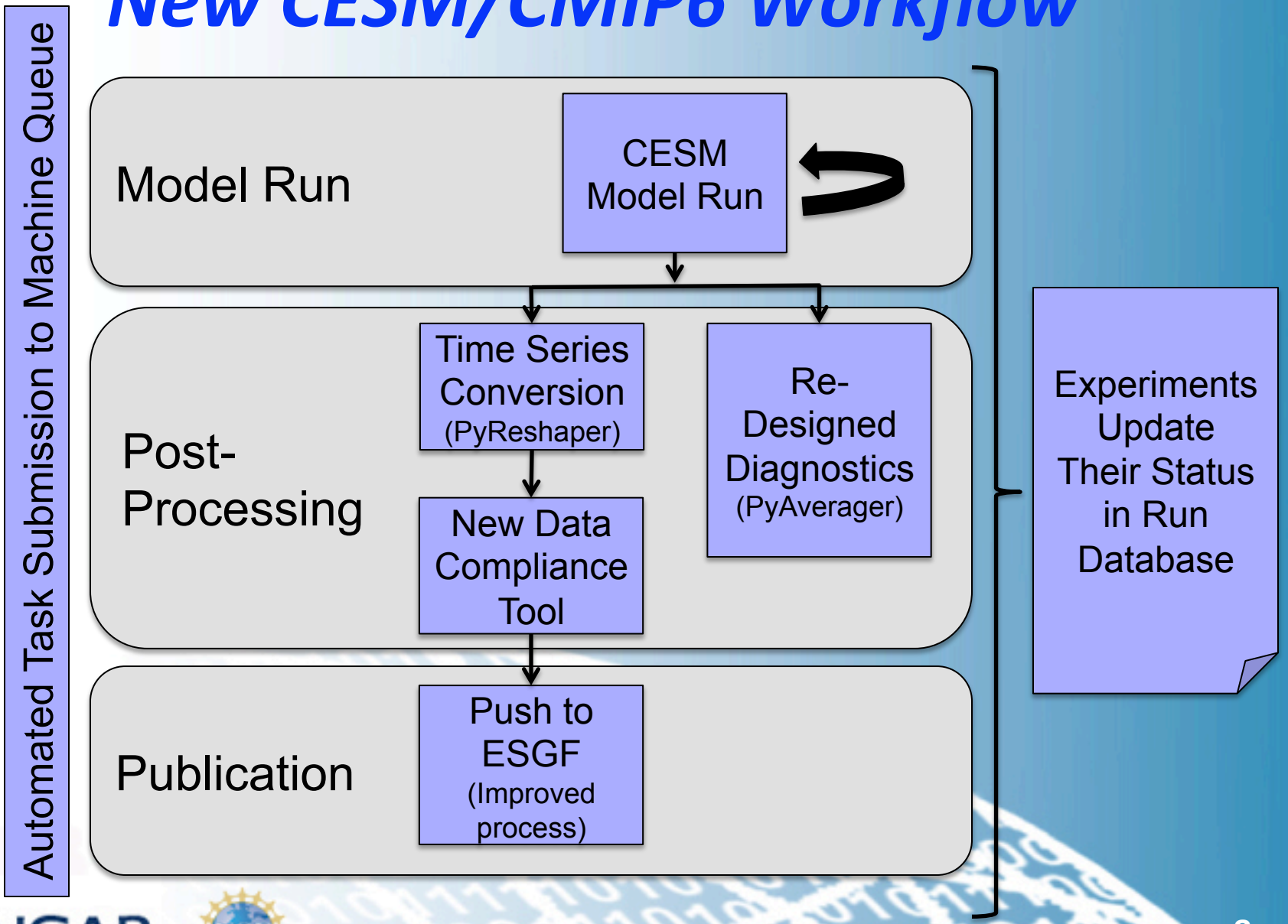


* Image from pictoor.com

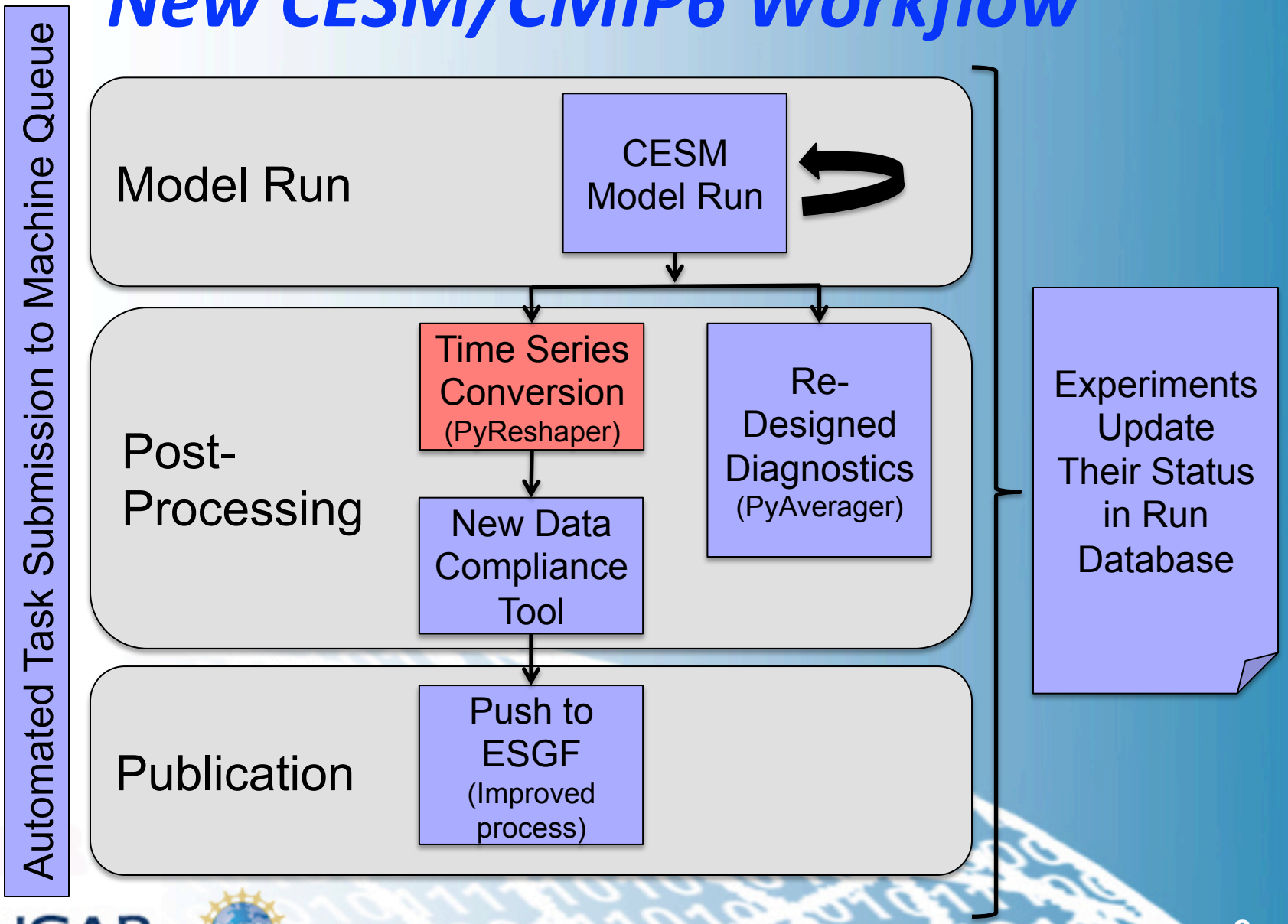
New CESM/CMIP6 Workflow

- We have been examining the individual pieces of the workflow and improving it where necessary
  - **Increasing performance:** Adding parallelization into the workflow
  - **Reducing Human Intervention:** Adding in automation
  - **Project Management:** more formal approach

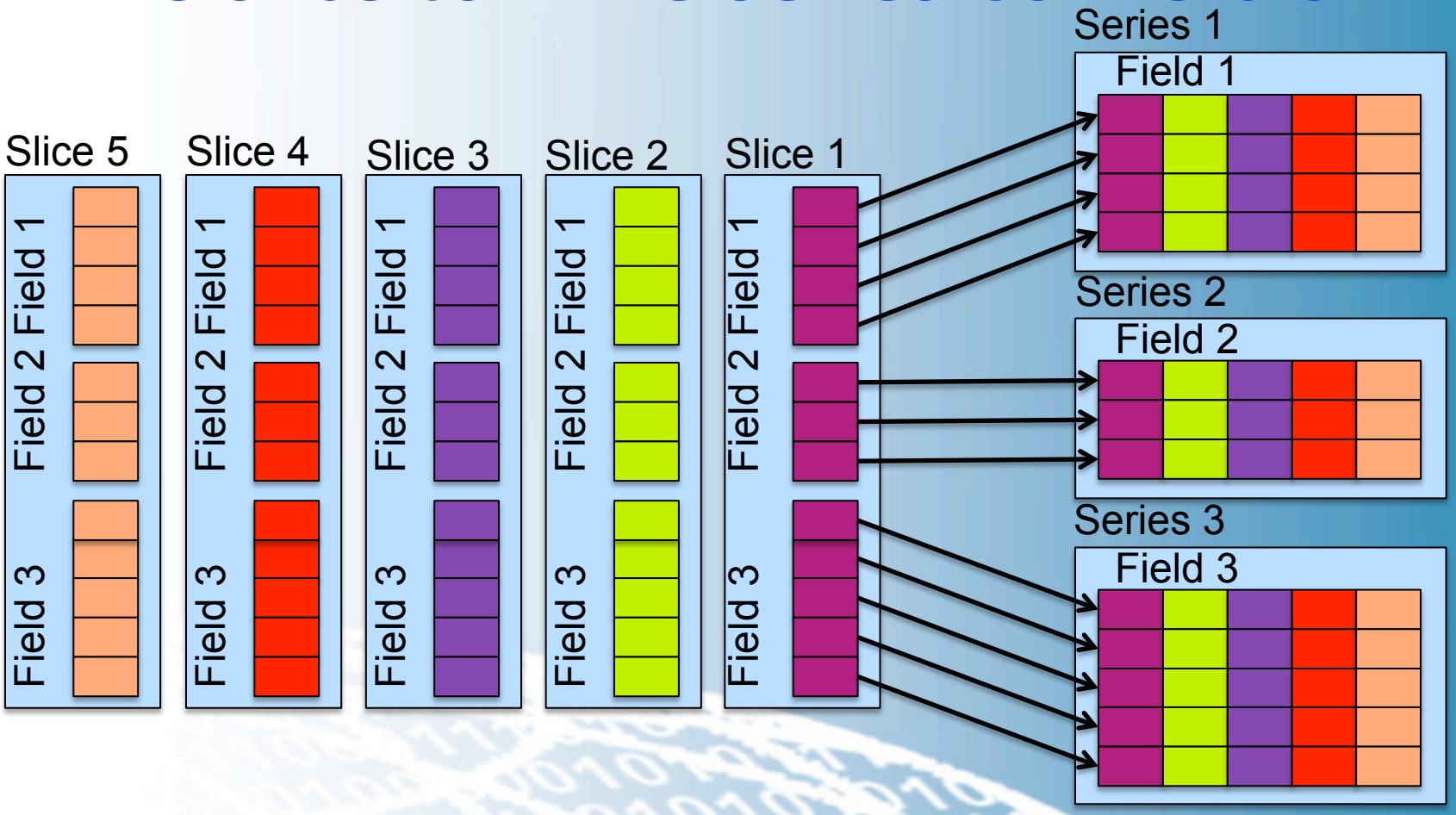
New CESM/CMIP6 Workflow



New CESM/CMIP6 Workflow



Time Slice to Time Series Conversion



- The previous method worked in serial using NCO
- This was one of the most expensive CMIP5 post-processing steps

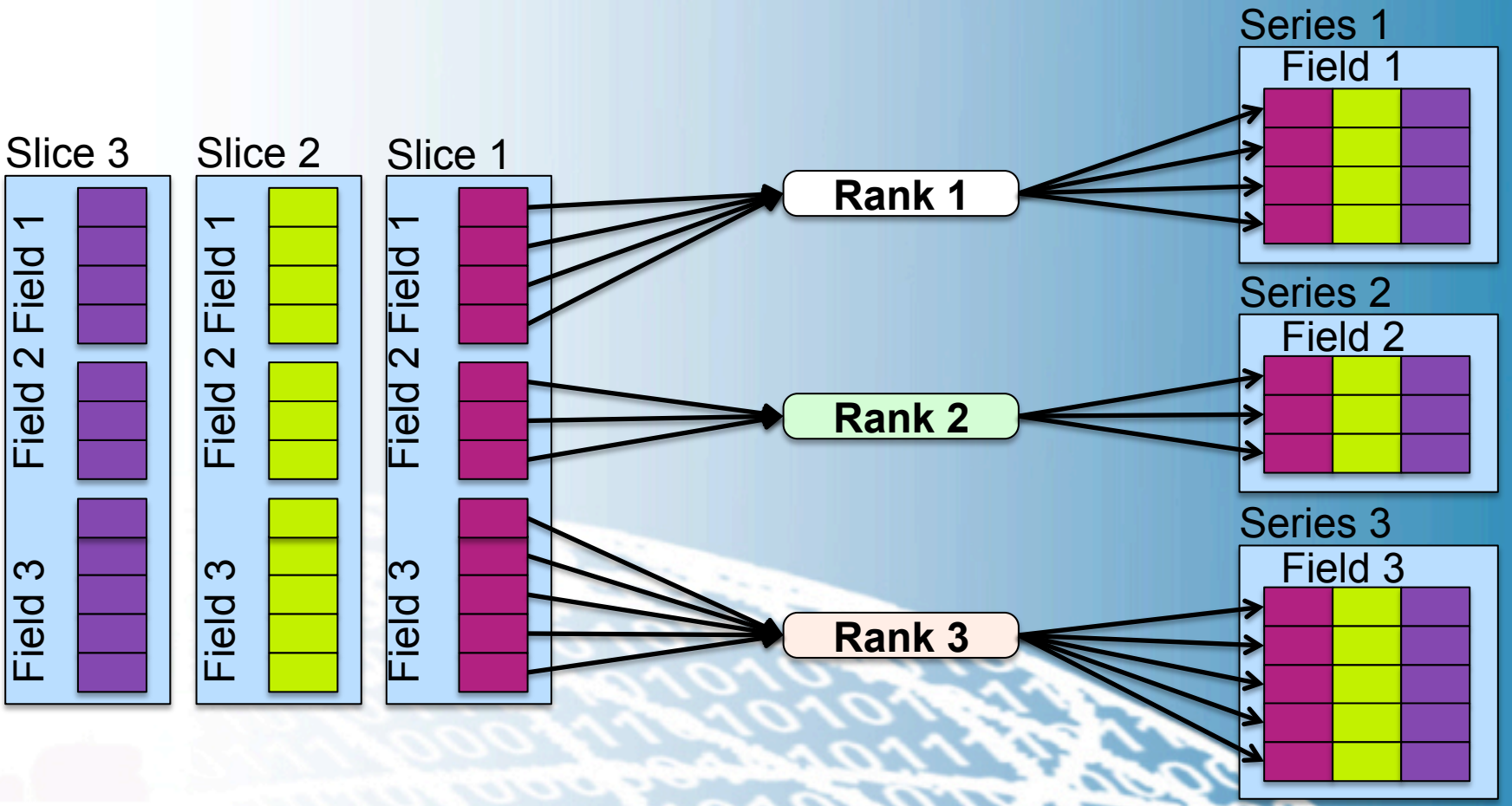
PyReshaper Details

The PyReshaper is a light weight custom Python concatenation tool

- **We chose Python for its flexibility and its fast development rate.**
- **For easier portability, we rely on only 3 packages**
  - PyNIO (Python version of the NCL I/O utilities)
  - mpi4py (Package used for parallelization)
  - NumPy (Used for data storage)

PyReshaper Parallelization Scheme

Each rank is responsible for writing one (or more) time-series variables to a file



Time-Slice to Time-Series Conversion

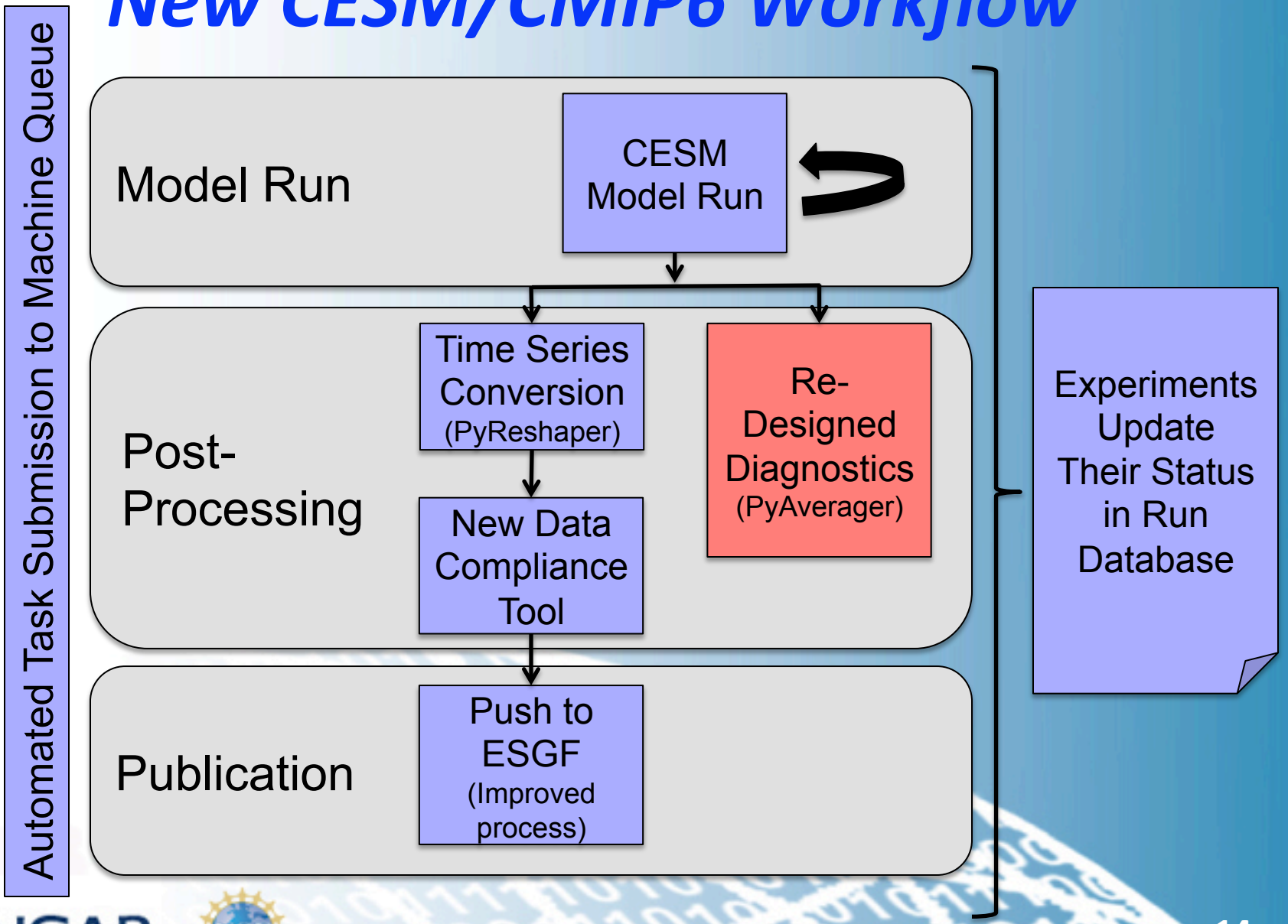
Timing Statistics

Existing Method (NCO)	Time (per MIP per Year)	Average Throughput (per run)	Throughput per Calendar Day
1° atm, 1° ocn	225 minutes	1.85 MB/sec	0.15 TB
25km atm, 1° ocn	478 minutes	4.85 MB/sec	0.40 TB

New Method (PyReshaper)	Time (per MIP per Year)	Average Throughput (per run)	Throughput per Calendar Day
1° atm, 1° ocn	4 minutes	104 MB/sec	8 TB
25km atm, 1° ocn	8 minutes	290 MB/sec	23 TB

- Conversions were ran on Yellowstone using 4 nodes/4 cores (16 cores total)
- The PyReshaper increases performance by 50-60x and achieves better machine utilization

New CESM/CMIP6 Workflow



Original Diagnostic Packages

- **Work focusing on redesigning**
  - Our 4 main component Diagnostic Packages
  - ILAMB Package
- **The Original packages**
  - Contain top level control scripts
  - Create climatology files with NCO tools
  - Create hundreds of plots with NCL scripts
  - Create web pages that allow users to browse through plots
  - The ILAMB Package creates climos and plots with NCL
- **Problems:**
  - Contain no parallelization
  - They often break at high resolution
  - They do not work with time series files

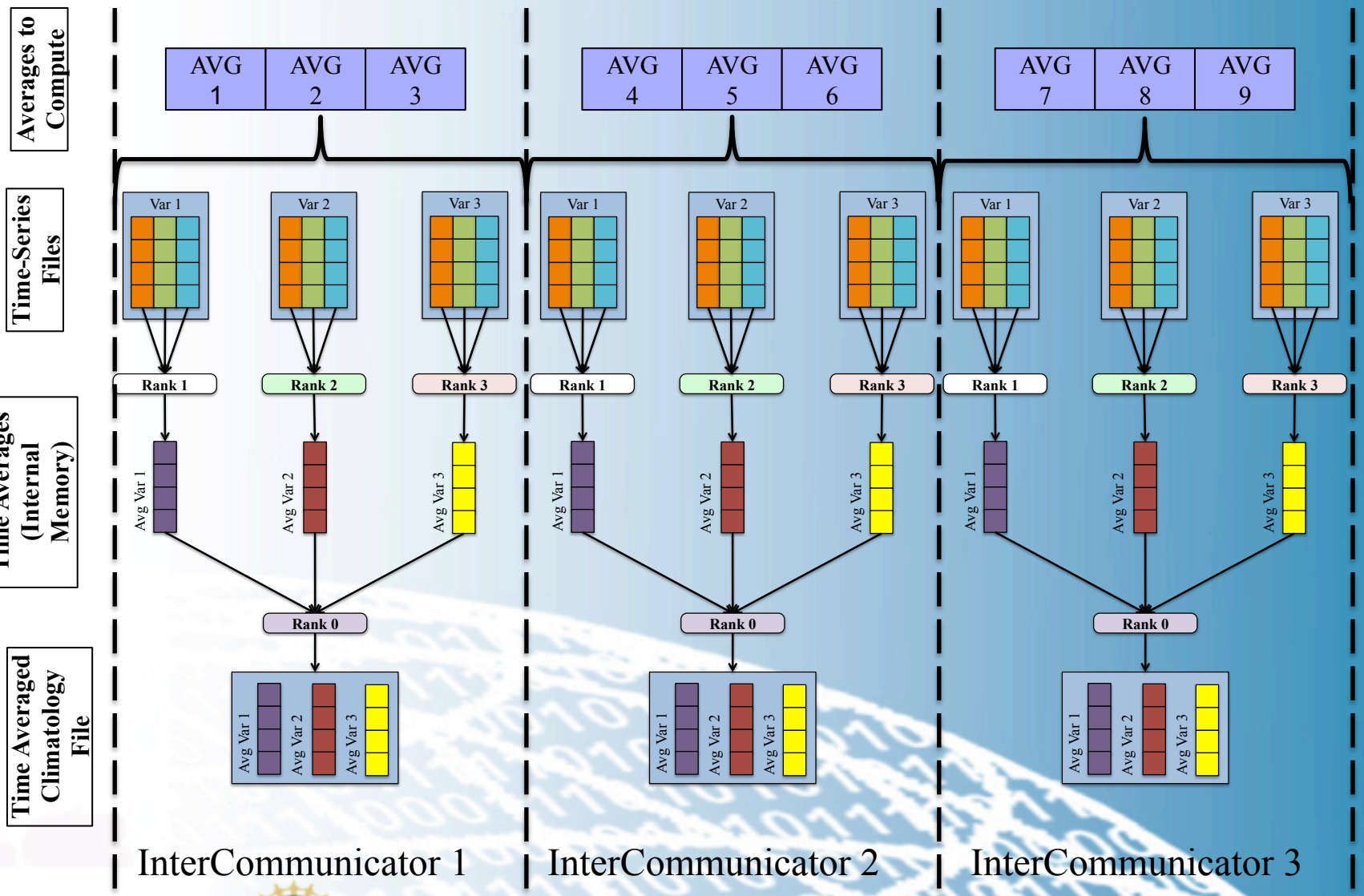
Re-Design of Diagnostic Packages

- **Add in Parallelization**
  - Instead of NCO, use the PyAverager to create the climatology files in parallel
  - Run the NCL plotting scripts in parallel
- **Allow the packages to work with either time slice or time series files**

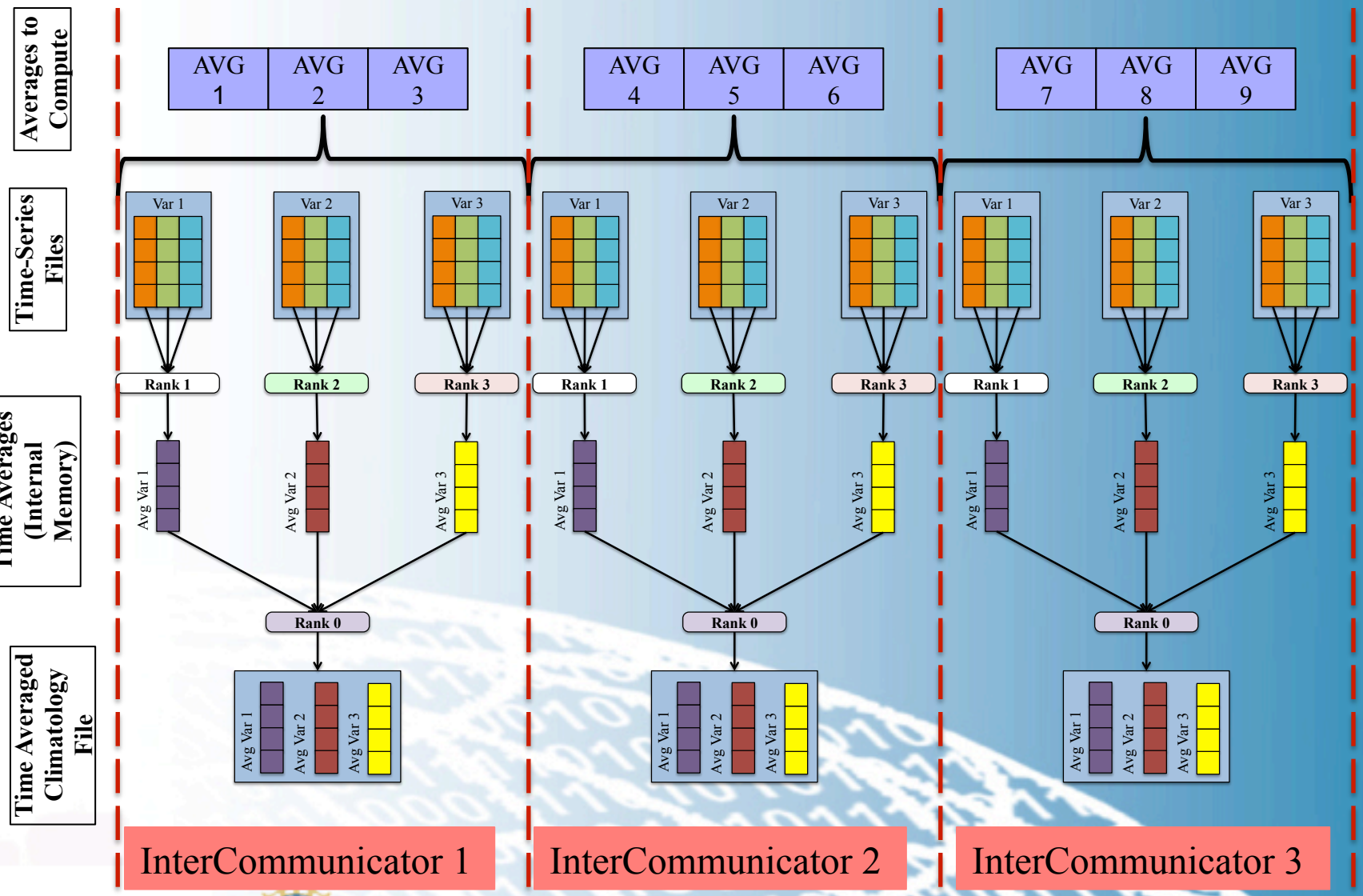
PyAverager Details

- **A light weight custom Python averaging tool**
  - Parallelizes over variables and averages
  - Depends on PyNIO, mpi4py, and NumPy
- **Computes temporal averages**
  - Seasonal, Yearly, Annual, Monthly (weighted optional)
- **Can concatenate in parallel**

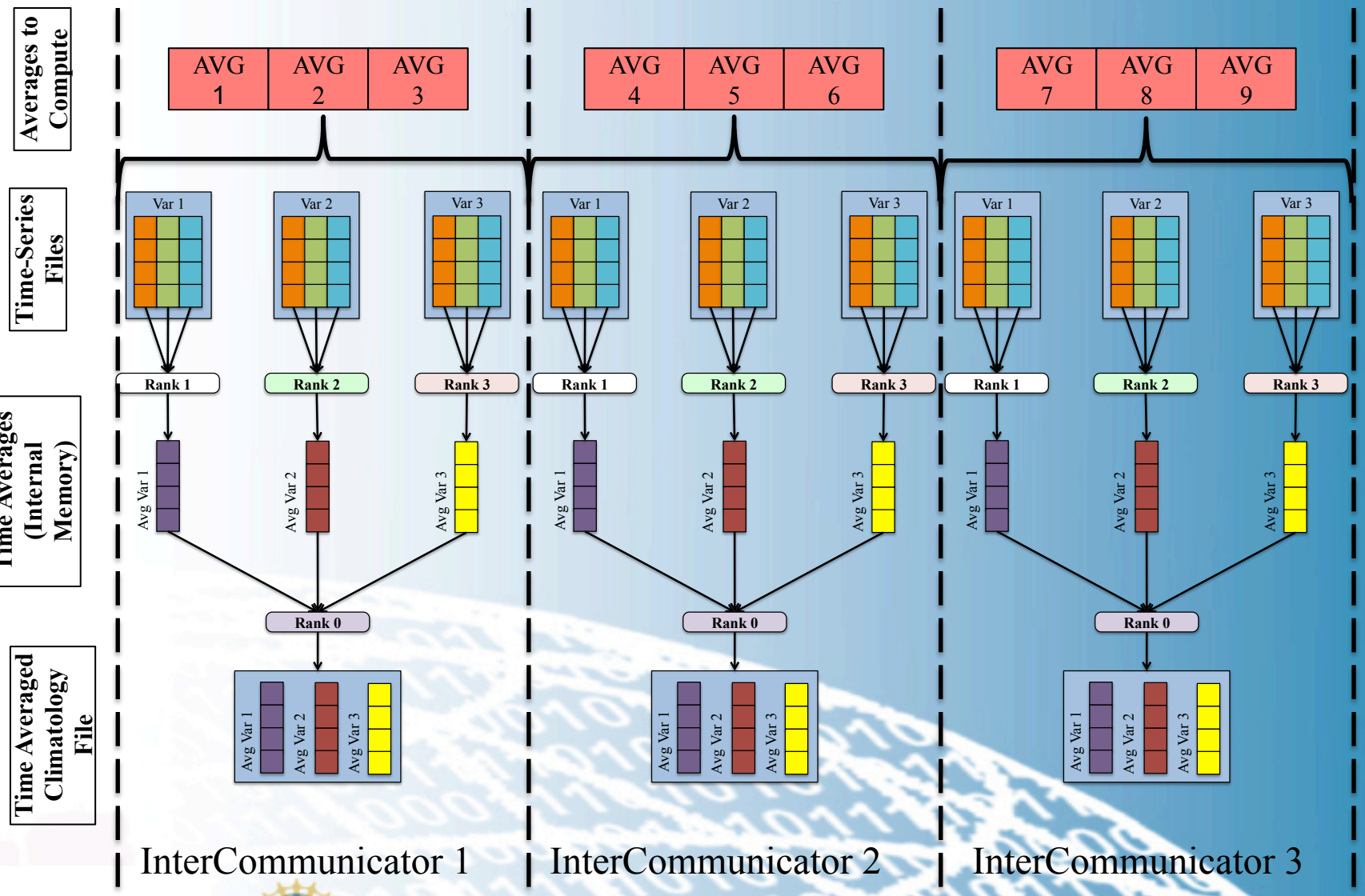
PyAverager Parallelization Scheme



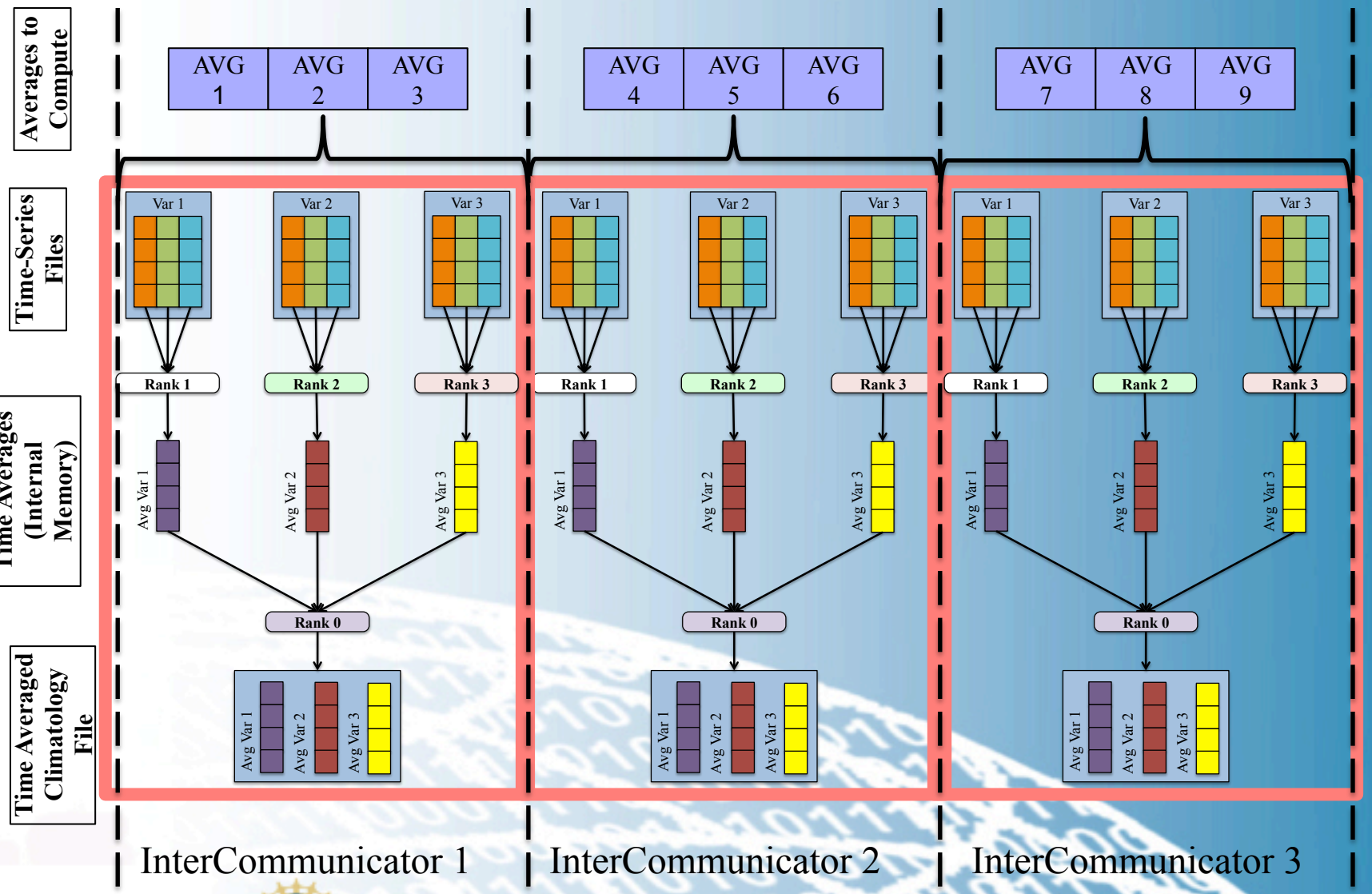
PyAverager Parallelization Scheme



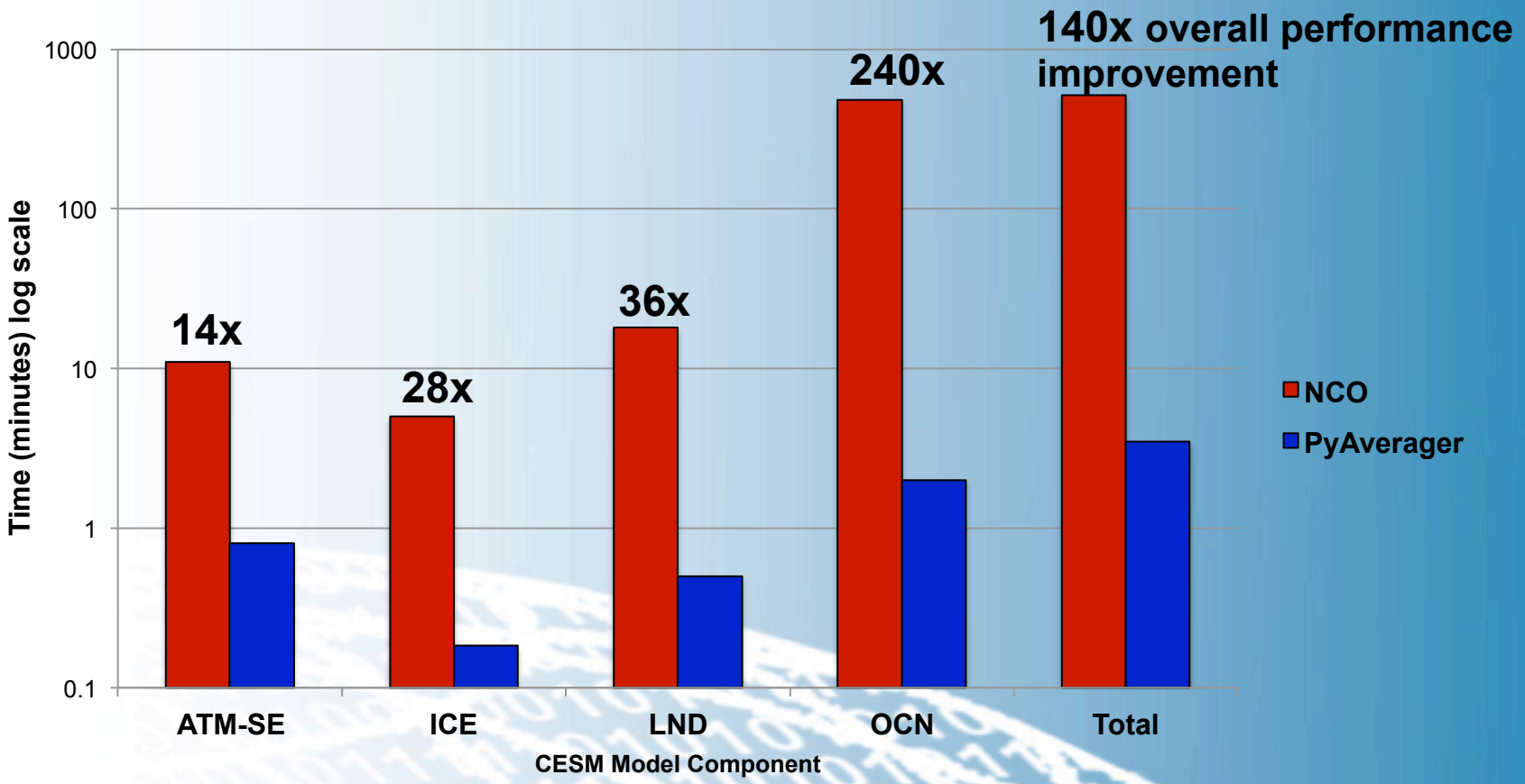
PyAverager Parallelization Scheme



PyAverager Parallelization Scheme

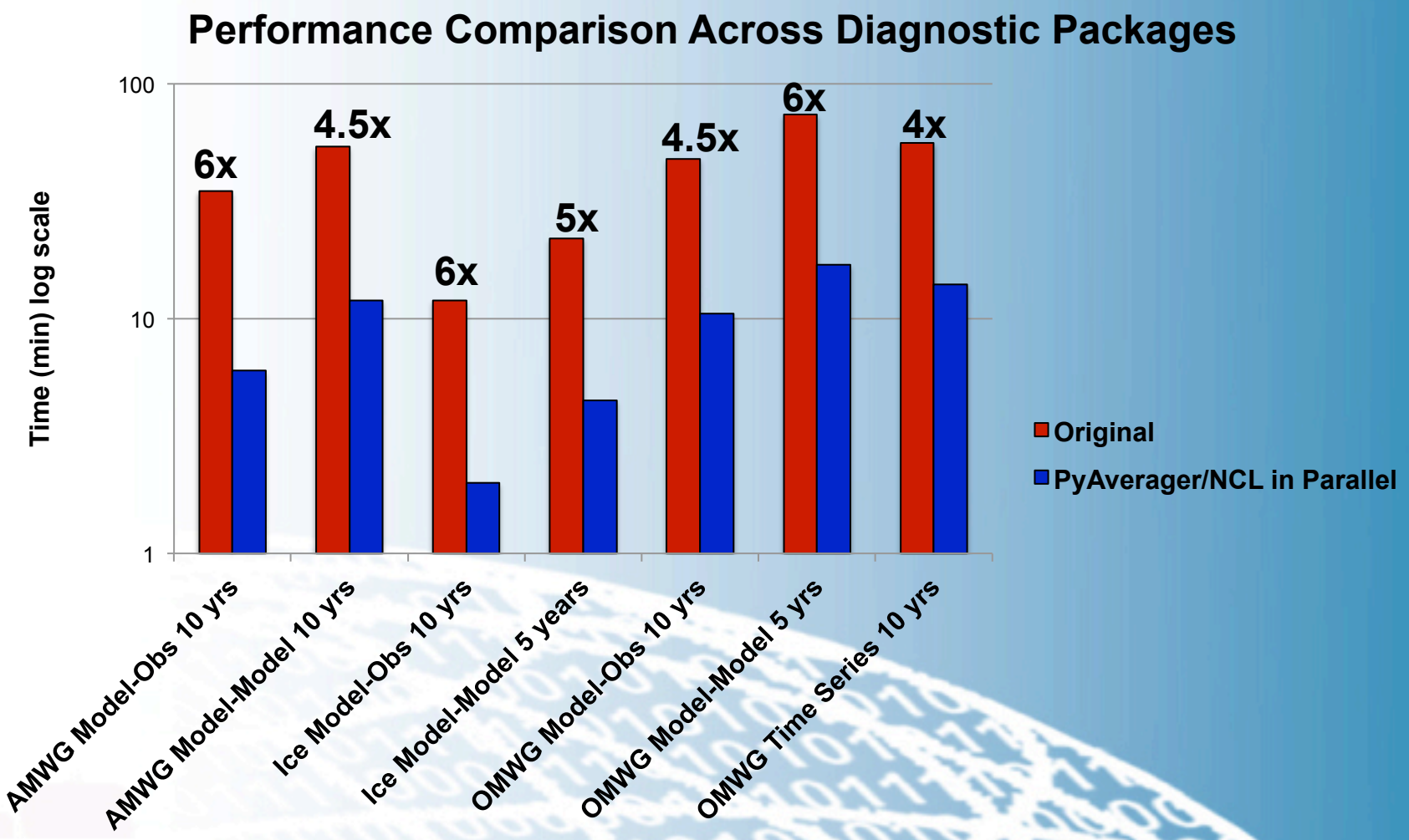


PyAverager Performance Using CESM Data



Time to compute climatology files for 10 years of CESM monthly time slice files.
The PyAverager ran on 120 cores on yellowstone.

Diagnostic Performance



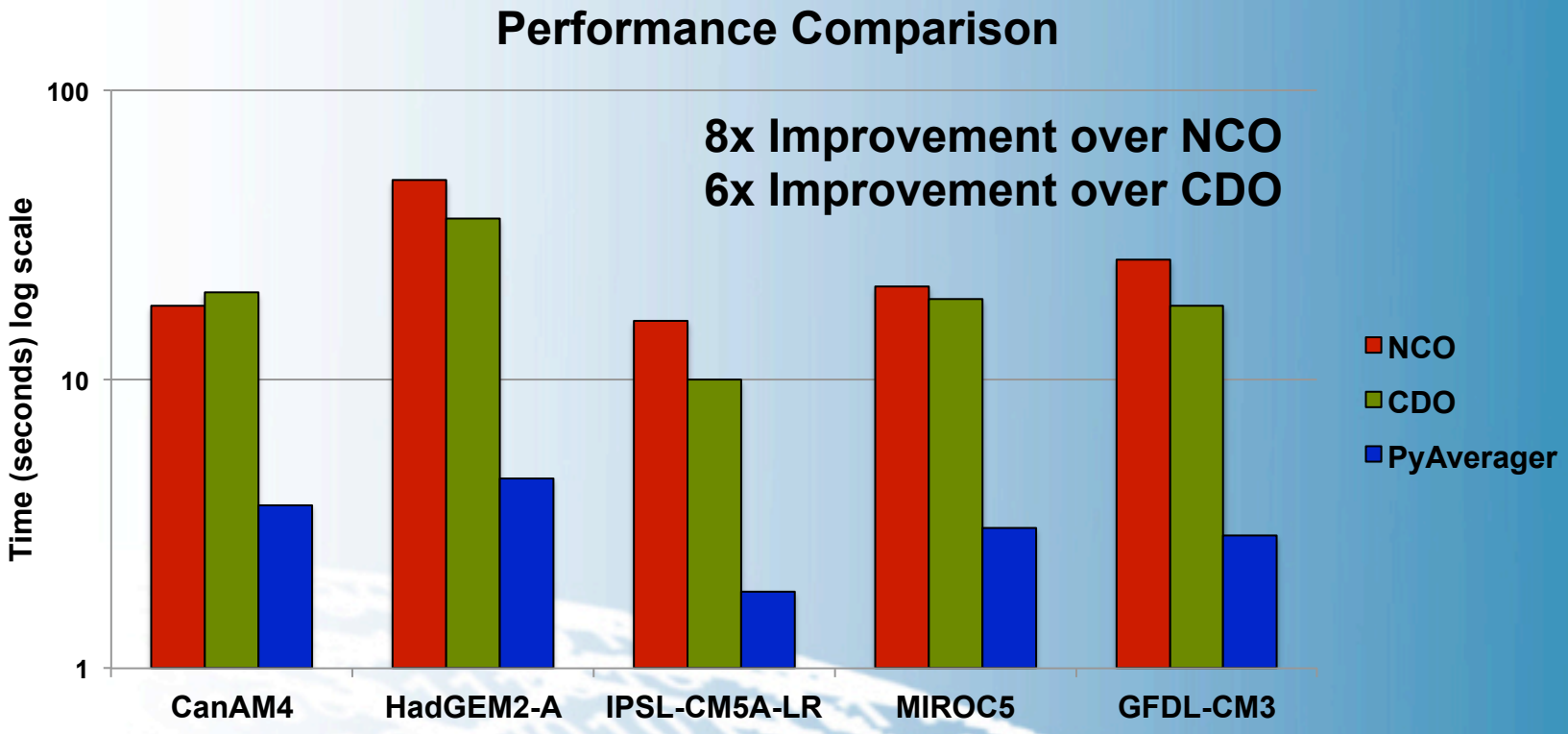
All parallel diagnostics were ran on 16 yellowstone cores

Making Our Tools General Enough to Handle Other Modeling Data

- **Both the PyReshaper and the PyAverager can operate on non-CESM data**

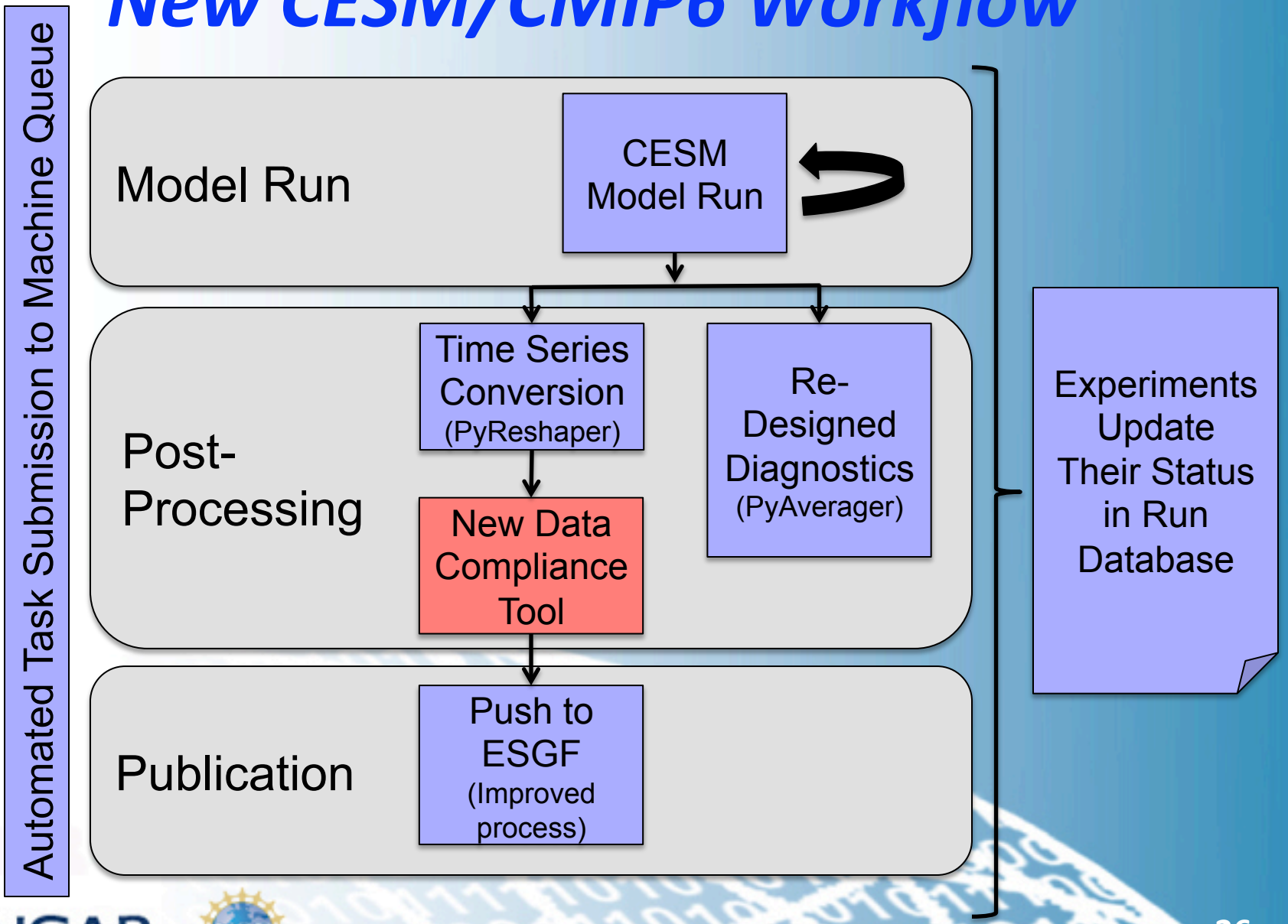
- **Because we chose PyNIO for our I/O library, we can read in any data type that NCL can handle**

PyAverager Performance Using Data From Other Models



CMIP5/AMIP monthly data. 5 seasonal and 12 monthly averages were computed over 29 years for 5 variables. The PyAverager was ran on 36 cores on yellowstone. The combined operation option was used for CDO.

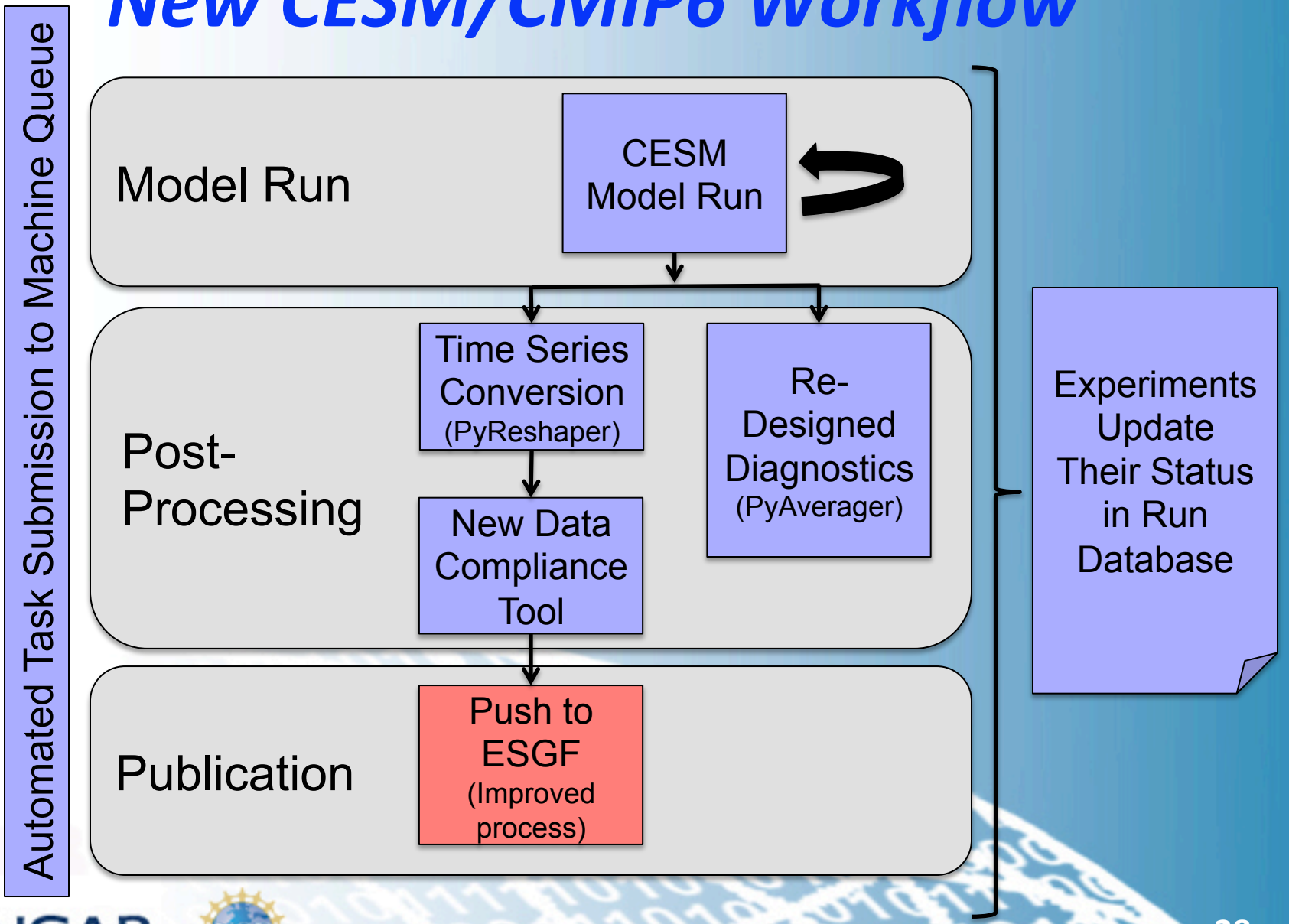
New CESM/CMIP6 Workflow



New Data Compliance Tool

- **Main Goals**
  - Provide a simple user interface
  - Add data transformation and calculator abilities
  - Add parallelization to increase performance (this was another bottleneck in CMIP5)
- **We are using similar techniques that were used by the PyReshaper and PyAverager**

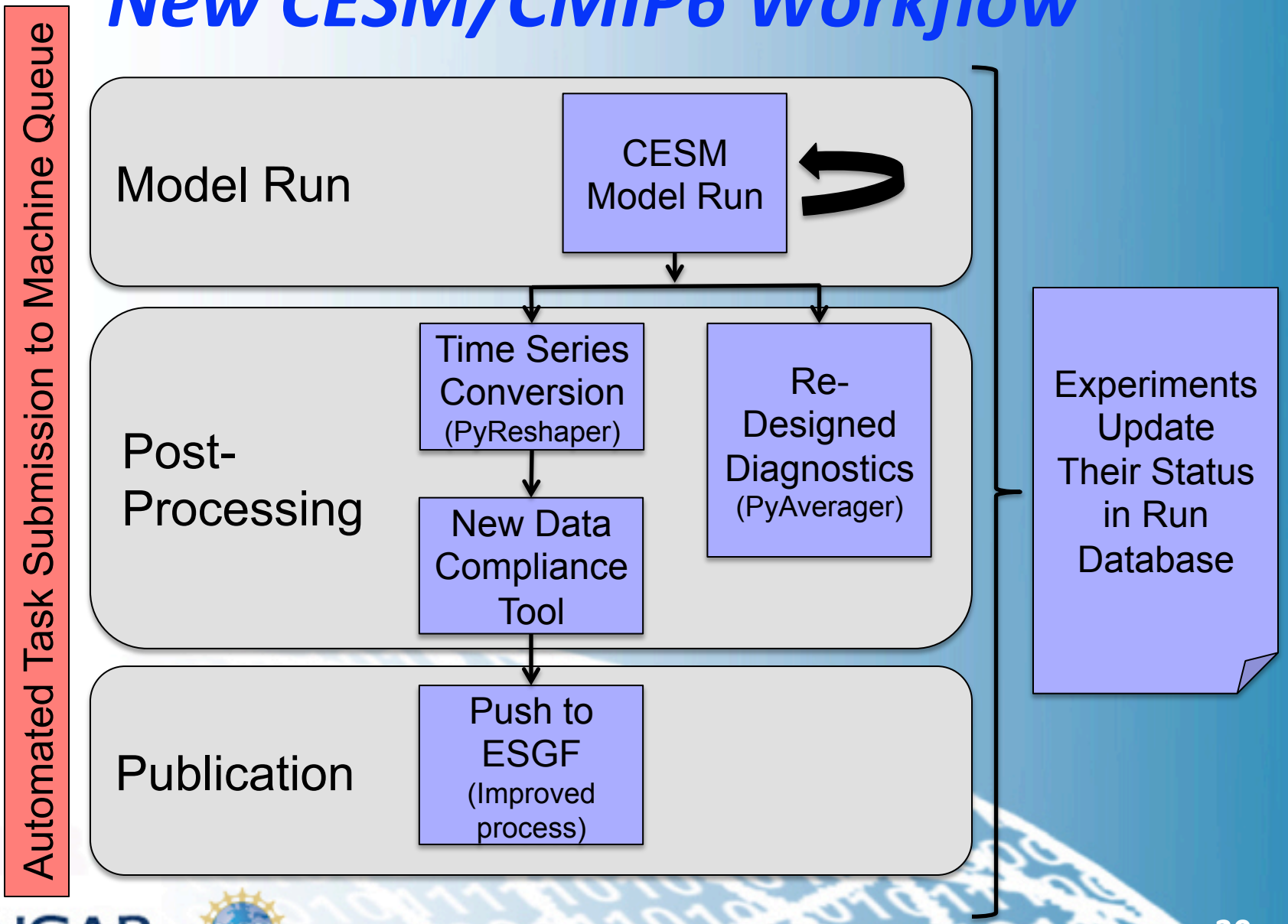
New CESM/CMIP6 Workflow



ESGF Publication

- **Move the data staging, directory structuring, and versioning responsibilities into the new compliance tool**
- **Streamline the submission process**
- **ESGF currently undergoing a major overhaul to address performance and reliability concerns**
- **New version of TDS (Unidata)**
  - Better memory and resource management
- **We will be setting up a test ESGF node to test new features and to access this portion of the workflow**

New CESM/CMIP6 Workflow



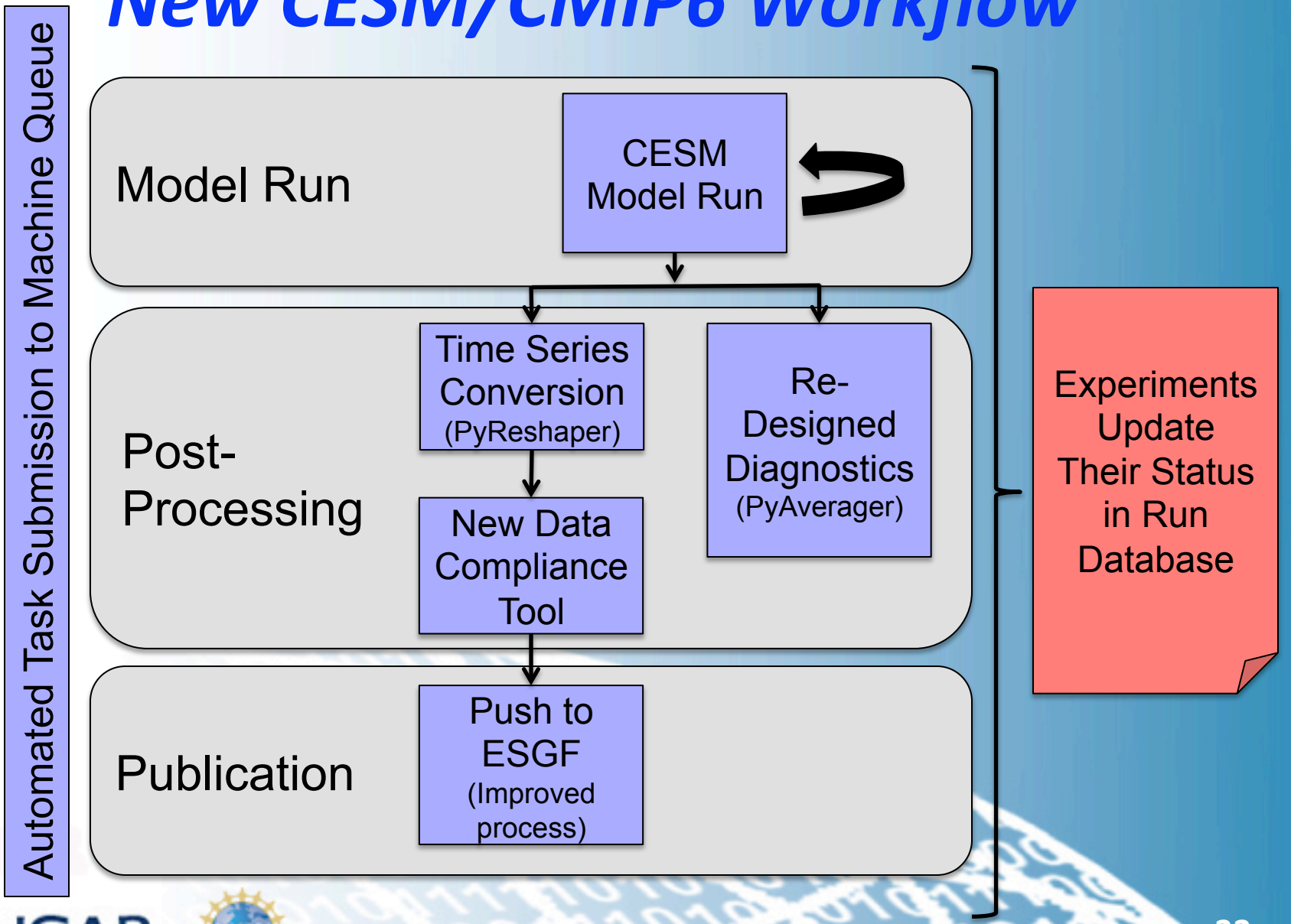
Requirements for an Automated Workflow Management System

- **We need a light weight and portable option**
  - Needs to run on everything from laptops to shared supers per CESM user community requirements
- **We evaluated Cylc and Rocoto**
  - Both are very impressive, but lacked the portability that CESM required
- **Since we have already developed an experiment database that we can extend, we only lacked a task scheduler**

Our New Workflow Management System

- **The system evaluates user options and submits workflow to the machine's queue as dependency jobs**
- **The management system will automatically:**
  - Run the PyReshaper after a select number of years has passed
  - Detect if and when to run diagnostics based on user selected date ranges
  - Run the data compliance tool after the PyReshaper successfully finishes

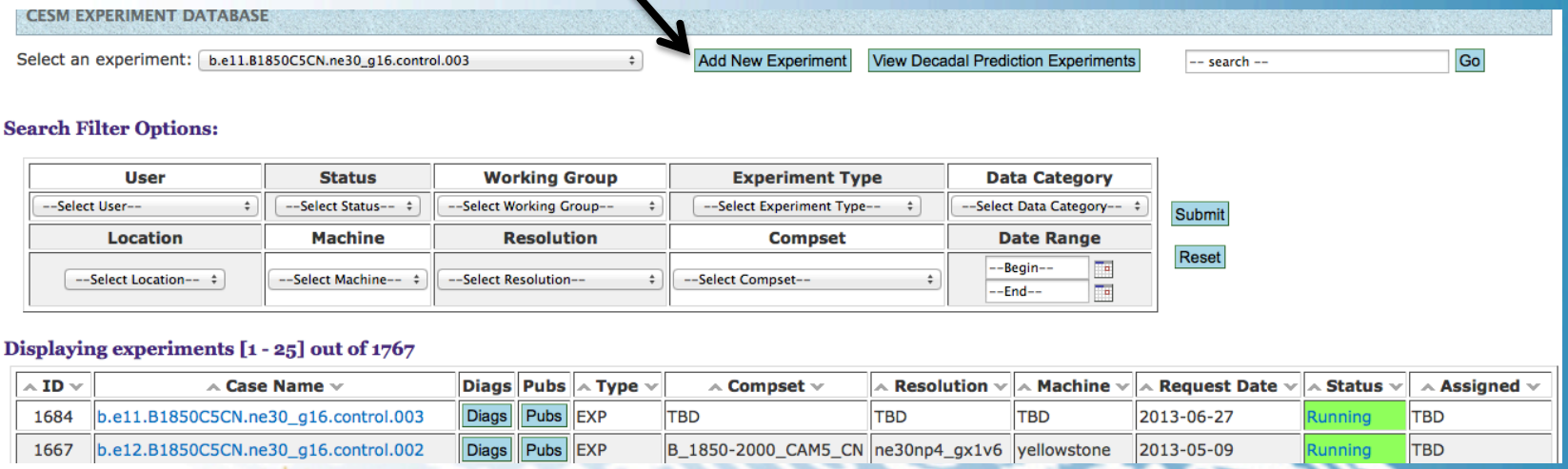
New CESM/CMIP6 Workflow



Current Run Database

Basic Catalog of Experiment Information

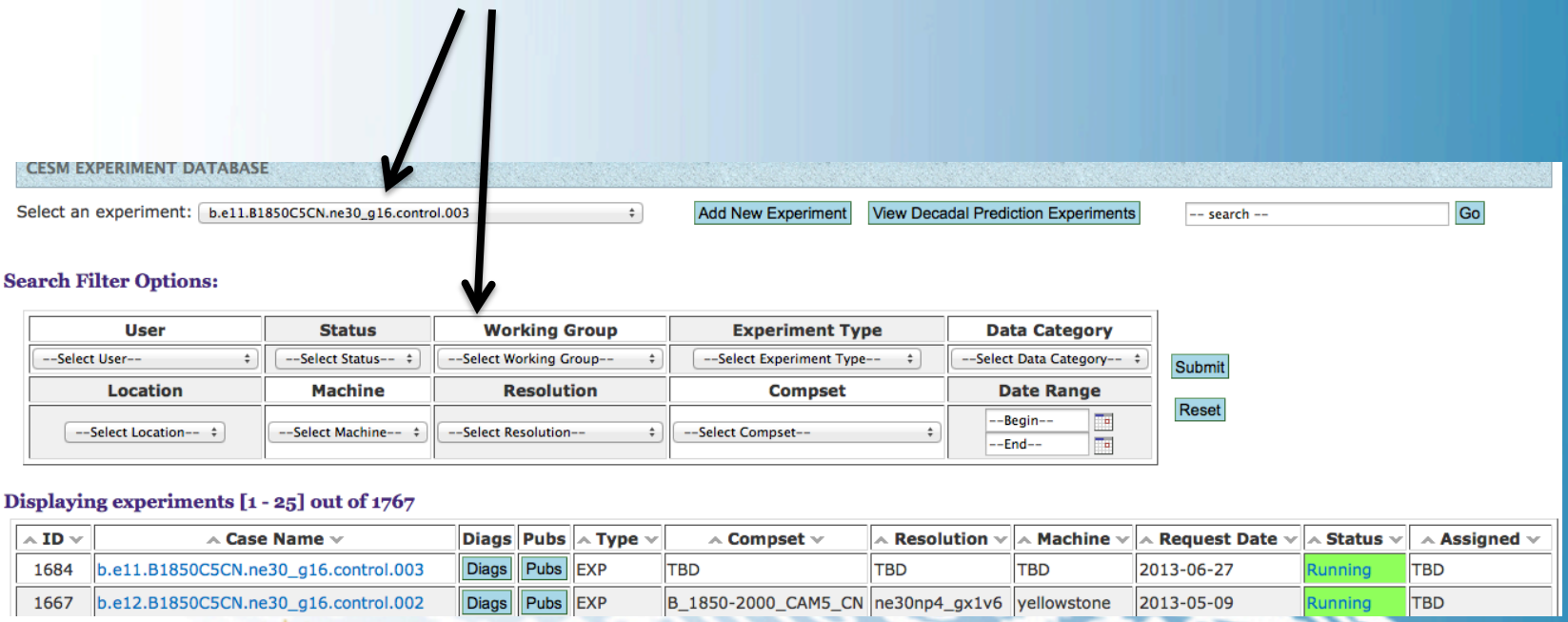
- Users manually enter experiments into the DB



Current Run Database

Basic Catalog of Experiment Information

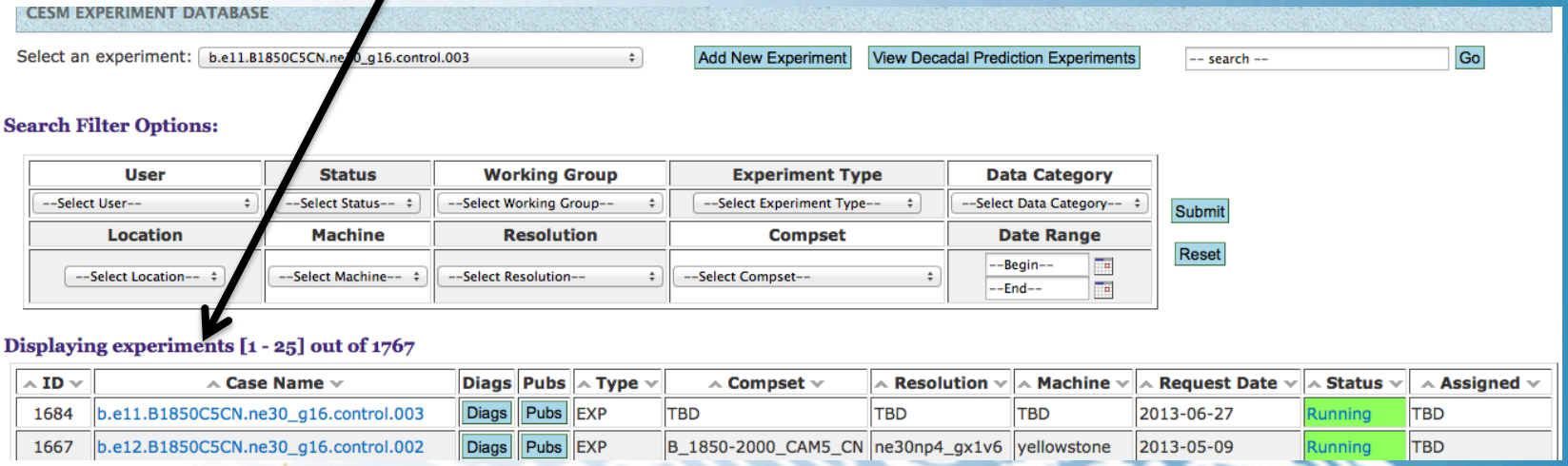
- Users manually enter experiments into the DB
- Contains basic search utilities and search filters



Current Run Database

Basic Catalog of Experiment Information

- Users manually enter experiments into the DB
- Contains basic search utilities and search filters
- Lists available experiments

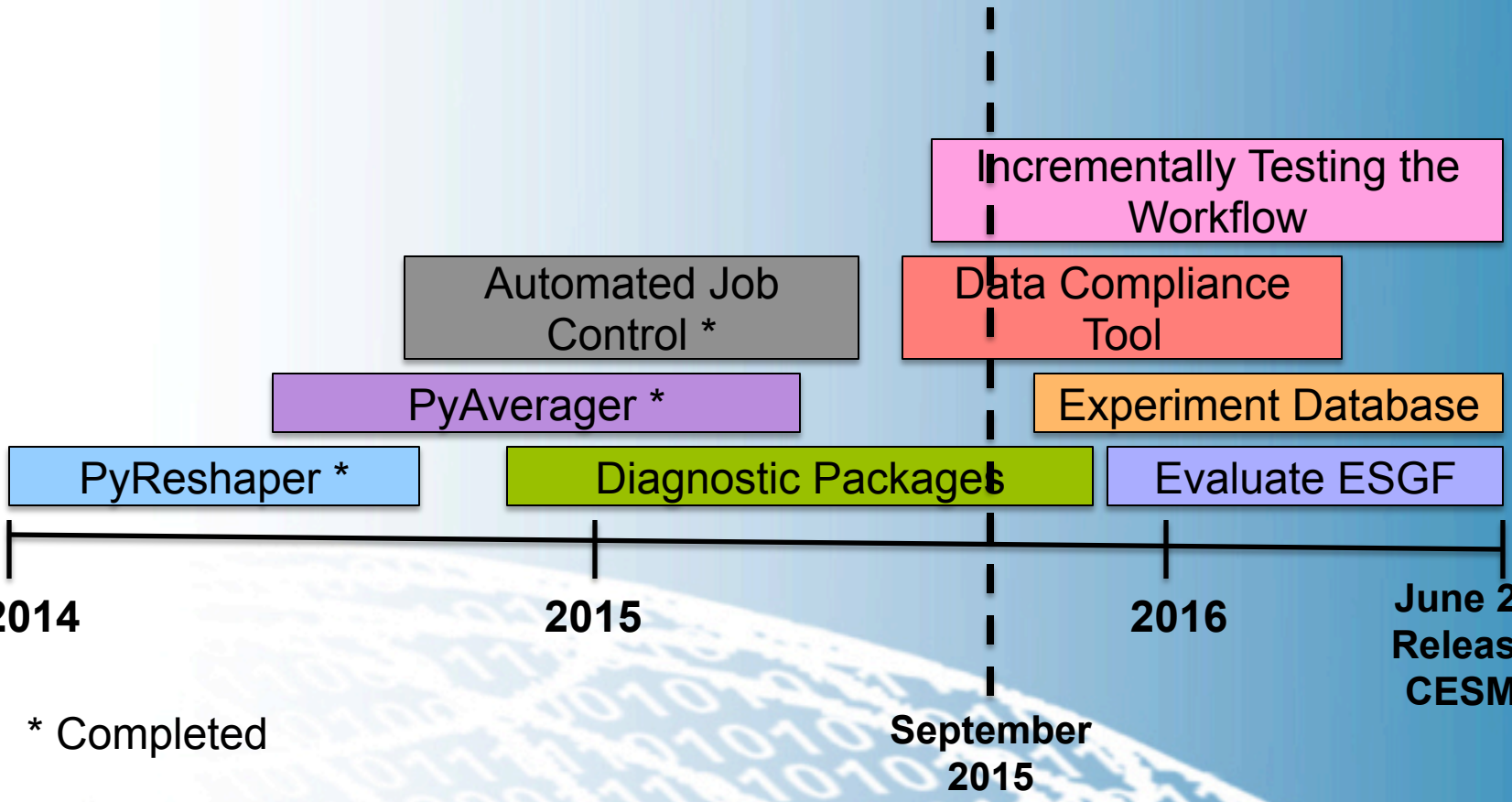


Enhancements to the Run Database

- **Automating the process of adding an experiment to the database**
- **Adding a separate section for CMIP6 experiments**
- **All experiments will update their run status to the database (simulation progress/color coded run status)**
- **CMIP6 timeline views**
- **Resource tracking (i.e. available disk space)**
- **Optional link to diagnostic web pages**

Project Timeline

January 2014 – June 2016



* Completed

Computational & Information Systems Laboratory

CISL

NCAR

Conclusions

- CMIP5 stressed our workflow and showed us where we needed improvements
- We are introducing incremental changes and adding them into our current workflow and testing them before CMIP6 starts
- Our new tools provide significant performance improvements
  - PyReshaper: 50-60x speedup
  - PyAverager: 14-240x speedup

Conclusions Continued

- **Our new Python tools are general across all models, do not have many dependencies, and can be easily integrated into a workflow**
- **We continue to building additional tools that will improve our workflow**

Questions?

Python Tool Availability

Github

- https://github.com/NCAR-CISL-ASAP/PyReshaper
- https://github.com/NCAR-CISL-ASAP/pyAverager
- https://github.com/NCAR-CISL-ASAP/ASAPPyTools

PIP

- pip install PyReshaper
- pip install pyAverager
- pip install ASAPTools

Contact Information

mickelso.at.ucar.edu

https://www2.cisl.ucar.edu/tdd/asap/parallel-python-tools-post-processing-climate-data