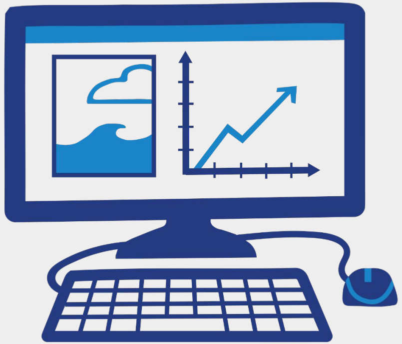


Modeling Systems Division (MSD)

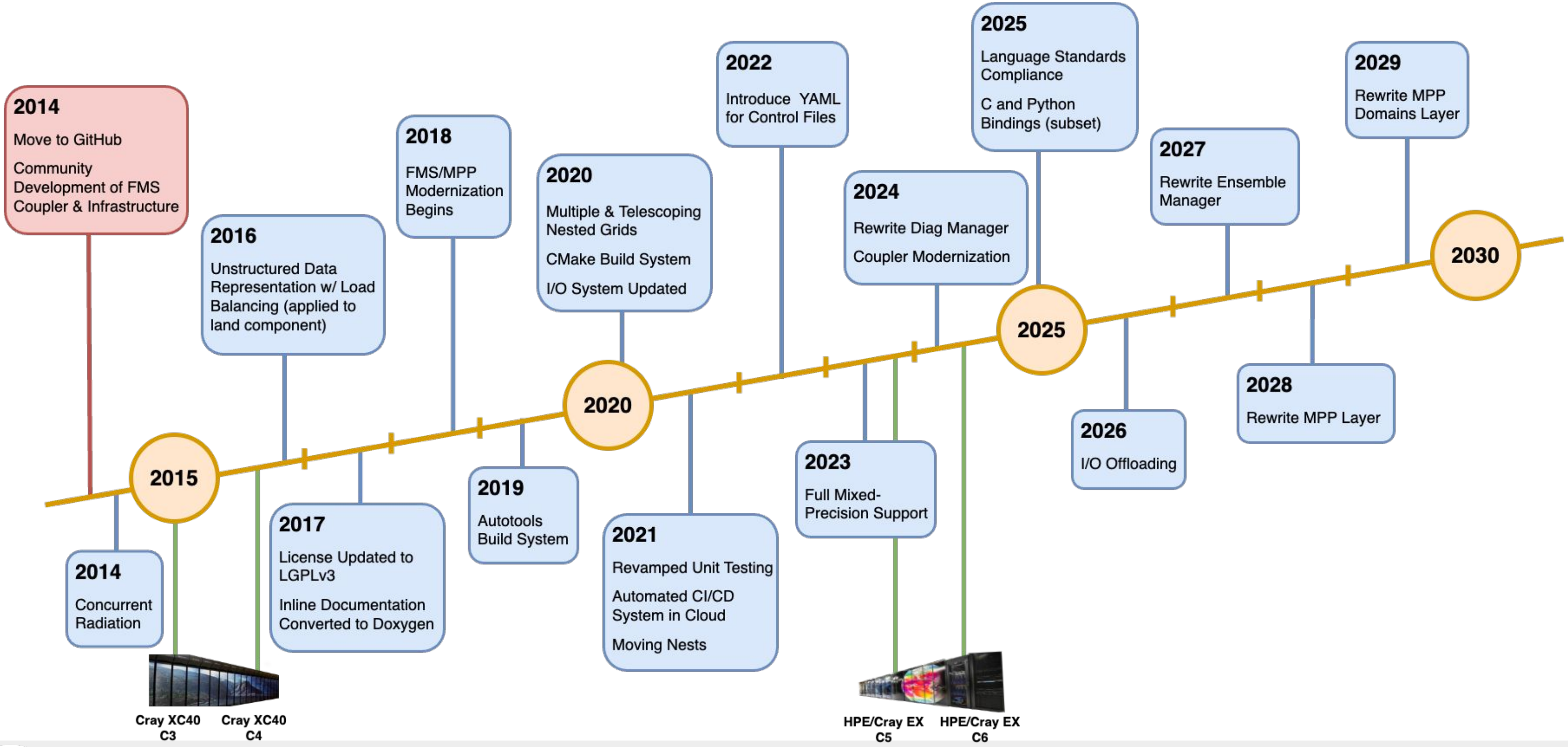
Rusty Benson, Chris Blanton, Andrew Brooks, Ciheim Brown, Lauren Chilutti, Halle Derry, Jeff Durachta, Josh Eisinger, Avery Kiihne, Ian Laflotte, Mikyung Lee, Jesse Lentz, Jess Liptak, Frank Malatino, Jacob Mims, Ryan Mulhall, Aparna Radhakrishnan, Uriel Ramirez, Kris Rand, Tom Robinson, Dana Singh, Seth Underwood, Hans Vahlenkamp, Carolyn Whitlock, Chan Wilson, Niki Zadeh

Q1: Concerning GFDL's core strength of building and improving models of the weather, oceans, and climate for societal benefits, how can GFDL leverage advances in science and computational capabilities to improve its key models? What are the strengths, gaps, and new frontiers?

NOAA
GEOPHYSICAL FLUID DYNAMICS LABORATORY

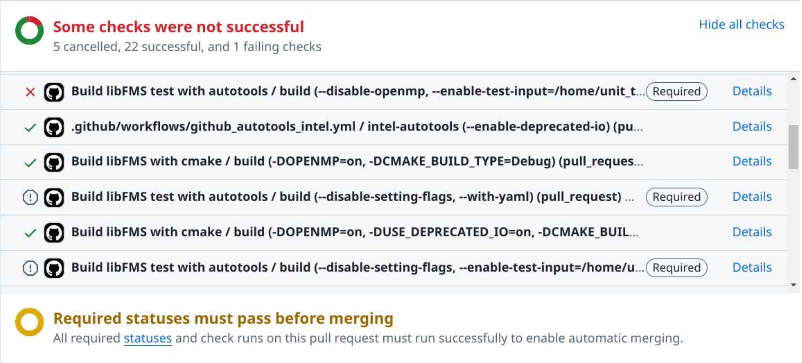
5-YEAR REVIEW
JANUARY 28-30, 2025

HPC / Flexible Modeling System (FMS) Timeline



FMS: Model Coupling Infrastructure Reliability

- Reliability through modernization and encapsulation
  - Object oriented Fortran updates
  - Mixed precision support
  - Fortran 2018 and 2023 incorporation
    - `do concurrent` loops added
    - 2018 standard MPI
- Reliability through extensibility
  - Single YAML format for all ASCII input
    - Schema validation for pre-run error checking
    - Easy to add features and parse
  - Reorganization and development to maximize code reuse
- Reliability through improved Testing
  - 517 automated unit tests executed before merging any code updates
    - Approximately 3500 tests are executed in different configurations with the GCC compiler alone
  - Guidelines and quality control standards for code reviews



GitHub displays the status of automated tests

FMS Runtime Environment (FRE) 2025 Workflow Features

FRE 2025 Workflow

Goal: Establish and build a GFDL workflow community through standardized structures, modular tools, and community developed workflow configurations

Modular Tools

- Ecosystem of tools written in python
- Modern command-line interface: ***FRE-cli***
- Broken-down 'tool subtool' structure:

```
fre make create-checkout
[options]
```

- *Enhances flexibility*
- *Modernizes code*
- *Increases maintainability of code*

Open Source Development

- Developed on **github**
  - fre-cli (toolbox)
  - fre-workflows (workflow definitions)
  - gfdl-msd-schemas (configuration validation & documentation)
- *Encourages collaboration*
- *Structure standardization*

Yaml Configurations

- Simple & common
- Easily parsed as python dictionaries
- Validated with json schemas
- fre-examples (public model examples)
- *Motivates unification*
- *Enhances diversity and flexibility*

Methods of Deployment

- Conda installable FRE-cli tools
- Containerized FRE-cli deployment
- *Increases portability*
- *Improves extensibility and reproducibility*

CI Tested

- FRE-cli tools tested for successes and expected failures
- *Improves code quality*
- *Quicker bug catches and fixes*
- *Increases maintainability of code*

Engineered Workflow Ecosystem Supporting GFDL Science

1 Diverse community requirements

Weather
GFDL FV3 will drive the NOAA weather model and GFDL scientists run SHiELD configurations on an operational-like schedule.

Ensembles
Seasonal and decadal research requires large ensembles to study uncertainty and predictability.

Climate
GFDL physical and earth system model results are among the most skilled and are used in national and international climate assessments.

Opportunities
Compatible and extendable tooling (fre-cli) and workflow definitions (fre-workflows) encourage flexibility while preserving workflow best practices and efficiency.

Dev / Testing
Shorter simulations to ensure component and coupled model correctness and performance

Users: Build, Run, Analyze, Publish

Workflow liaisons: Community, culture, tech

2 Shared lab-wide requirements

High automation
Multiple experiments on multiple HPC sites with little intervention

Mask HPC Complexity
Leverage HPC innovations while minimizing tool changes

Shared and User-Defined Tools
Modular shared workflow toolbox (fre-cli) with user-defined scripting

Provenance
Repeatable workflows with rich logging and the EPMT metadata toolkit provide evidence for tracing results or examining numerical divergences.

Cylc Workflow Engine

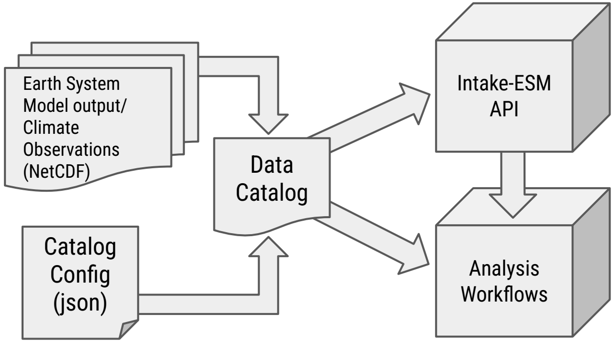
Robust and reliable open-source standard
Dozens of worldwide institutions rely on Cylc for production and research weather and climate workflows

Extendable
Plug-in capability provides additional flexibility

Scales with Complexity
Flexible, cycling, and advanced scheduling capability supports heterogeneous HPC

Data Catalogs

- GFDL developed an open-source catalog builder, available as a standalone conda package and as part of the FRE ecosystem.
- With these catalogs, users can leverage community data exploration tools to efficiently query and analyze large and diverse Earth System Model datasets.
- Data catalogs are comprised of catalog specification (json) file and a csv file with user defined granularity. This granularity is defined in the catalog specification in the form of the CSV headerlist.



Data catalogs allow for varying data sets to be virtually aggregated and searched in novel fashion through use of community-developed Intake-ESM API. Analysis workflows utilize this searchability to improve data discovery.

Earth System Modeling (ESM) oriented HPC containers

- Containers are
  - software that encapsulates an operating system, a model, and the software dependencies of the model
  - not confined to the system they were created on and can run on different computing platforms
- GFDL models will no longer be confined to the system they were developed on
  - Sharing model containers will foster collaboration and scientific advancement
  - Container software environments are identical across different systems leading to more reproducible science
- Proof of concept complete
  - Same performance as running a native build application
  - Seamlessly built into the GFDL FRE workflow
- Same User Interface for building model and containers
  - Using the same FRE-cli front-end, scientists can build and run containers
  - Minimal knowledge in building or running containers is necessary for users
  - The FRE backend handles all container components
  - We can track provenance and enhance reproducibility standards

Software Development Best Practices

- Software Change Management and Testing Automation
  - Using Github and Gitlab for software change management streamlines collaboration, enhances maintainability, and ensures the integrity of our research software allowing for more efficient version control and better reproducibility of experiments
  - Seamless continuous integration: utilizing cloud and on-prem HPC resources for automated small-scale unit testing and large-scale regression testing of models such as AM5 and SHiELD
  - Well defined code review and collaboration guidelines promote community engagement while maintaining quality and reliability
- Model Testing
  - Model liaisons test alpha/beta pre-release code versions with GFDL models
  - Public code releases with bug fixes and new features are scheduled multiple times per year
- Model Development collaboration
  - FMS model liaisons work with science teams to ensure next generation models take full advantage of the current code base
- HPC System testing
  - Early access testing of GFDL models on latest available hardware
  - Improving software accessibility with spack
  - Leveraging container technology to alleviate model reproducibility issues during system and software upgrades

SENA: The NOAA Software Engineering for Novel Architectures Initiative

NOAA funds HPC Technology Research with potential to provide revolutionary improvements for ESM at GFDL and across OAR

Program Goal: Create the pool of knowledge and experience necessary to enable transition of NOAA's model suites and workflows to exascale capable infrastructure.

Towards this goal, GFDL works in four project areas:

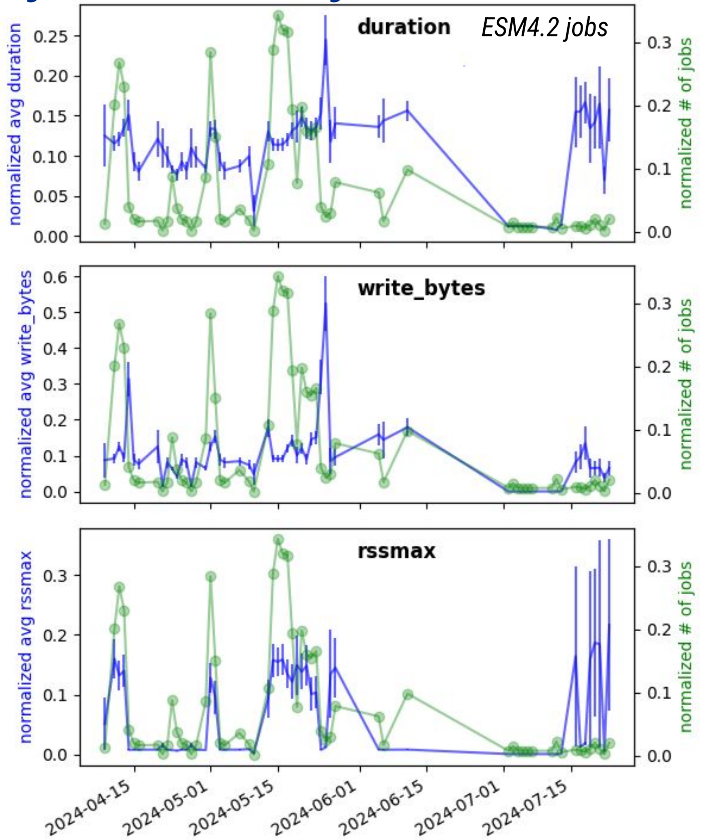
- Model Performance & Portability via a Community Domain Specific Language - [N]DSL (3 FTEs [1])
- Workbench for Modernization of Software Applied to Emerging Hardware Technologies Using Pre- and Post-Processing Toolsets for Case Study (1 FTE)
- Workflow Optimization Through Modern Python Analytics (2.5 FTEs)
- Model & Workflow Portability via Container Technology - ESM Oriented HPC Containers (4.5 FTEs)

[1] Full Time Equivalent

See also FY24 High-level Accomplishments / FY25 Project Goals

Workflow Throughput Optimization via Python Analytics

- Increasingly complex climate models (workflows) have incentivized ***constant compute hardware/infrastructure upgrades***
- HPCs catering to this are ***large, complex, and heterogeneous***
  - E.g., PPAN houses several generations of system components
- ***Problem:*** High system AND workflow complexity leads to…
  - ***low visibility into issues***, reliance on copious log output
  - ***hundreds to thousands of GBs*** worth of logs to sift through
  - can easily lose whole day(s) debugging, yielding no solutions
  - lost time ⟺ ***lost scientific productivity*** ⟺ lost dollars
- ***Solution:*** targeted annotation w/ data scraping + modern software
  - *FRE 2025 + Cylc:* ***modular/clear configuration*** in yaml, bash, python
  - *EPMT*: facilitates ***metadata annotation, aggregation, retrieval, transport***, provides detailed performance metrics non-invasively
- ***Result:*** gain insight with standard data-science tools into workflows
  - ***Fig. (right)*** *shows* data scraped FRE-bronx ESM4.2 workflows via EPMT.
  - ***short/long-term trend analyses*** of interactions between PPAN and our standard workflow software are now possible → we want data-driven engineering insights!



Tiny improvements become gigantic when integrated over time

Enhancing model development via Data Services and Model Diagnostics Task Force

- Framework development for NOAA's Model Diagnostics Task Force (MDTF) is led by GFDL, connecting model development with academic and private sector expertise to improve model scientific fidelity.
- Globus enhances data sharing, driving modeling innovations through stronger internal and external collaborations.
- GFDL internal Unified Data Archive-driven processes streamline model development by providing centralized, accessible input and supporting data for faster analysis.
- GFDL pilots cloud-based projects for elastic testing, data storage, and analysis-ready computing, such as NOAA's Open Data Dissemination Program.
- GFDL collaborates with the Earth System Grid Federation to make CMIP simulation data publicly available[1].
- Seamless integration with the modeling workflow, user analyses, and data services achieved through data catalogs.

[1] See A.Radhakrishnan presentation, Question 3