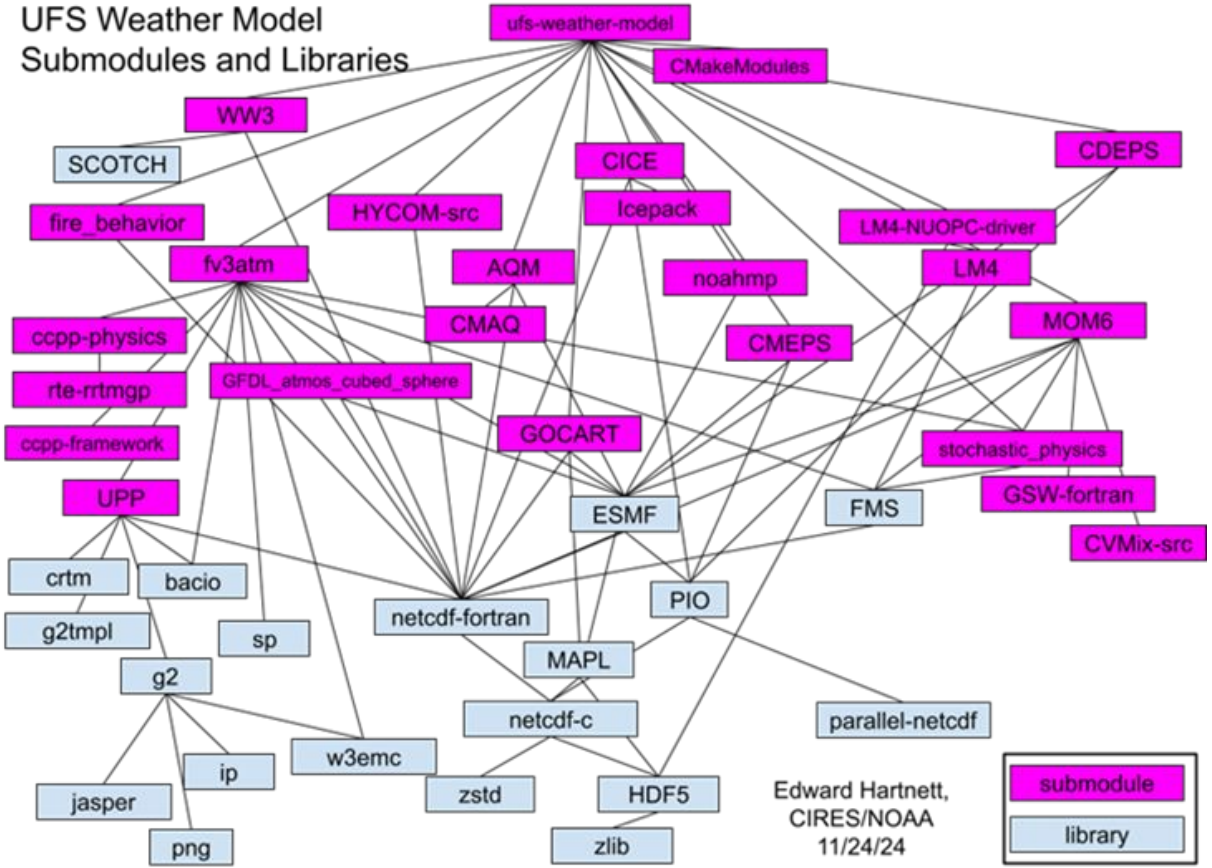
Addressing Core Modeling Infrastructure Barriers in Next-Generation UFS Weather Model Development

Daniel P Sarmiento*[1,2], Denise Worthen[1,2], Barry Baker[1], Ligia Bernardet[3], Robert L Jacob[4], Rahul Mahajan[1], Dustin Swales[3], Vijay Tallapragada[1], Ann Tsay[5], Jacob R Carley[1]

1 National Oceanic and Atmospheric Administration, Environmental Modeling Center
2 Lynker
3 National Oceanic and Atmospheric Administration Global Systems Laboratory
4 Argonne National Laboratory
5 NSF National Center for Atmospheric Research

Unified Forecast System (UFS) Weather Model (WM)



- Coupled Earth modeling system that spans a range of applications and operational forecast systems

UIFCW26
A UFS Collaboration Powered by EPIC

UFS Infrastructure Challenges

Community-Driven

Open & Collaborative Needs

- Designed fundamentally as a shared, public community model.
- Promotes rapid research, academic involvement, and shared code development.
- Prioritizes flexibility and ease of contribution to drive scientific innovation.

Operationally-Ready

Strict Operational Standards

- Serves as the critical backbone for operational products (e.g., GFS, RRFS, HAFS).
- Demands rigorous testing to ensure systems remain stable and reliable.
- Enforces tight verification rules and code standards.

The Intersection & Conflict

Drawing the line between **strict operational governance** and **open community collaboration** remains a complex, tricky balance to navigate.

UIFCW26
A UFS Collaboration Powered by EPIC

UFS Infrastructure Challenges

Community-Driven

Open & C

- Desig
shared
- Prom
involve
develo
- Prior
contrib
innova

Operationally-Ready

e for
RFS,
o
nd
s,

These are not mutually exclusive goals!

With the introduction of MPAS, we can take the opportunity to create a community-friendly system that promotes rapid development without sacrificing rigorous testing.

The Int

Drawing the line between **strict operational governance** and **open community collaboration** remains a complex, tricky balance to navigate.

UIFCW26
A UFS Collaboration Powered by EPIC

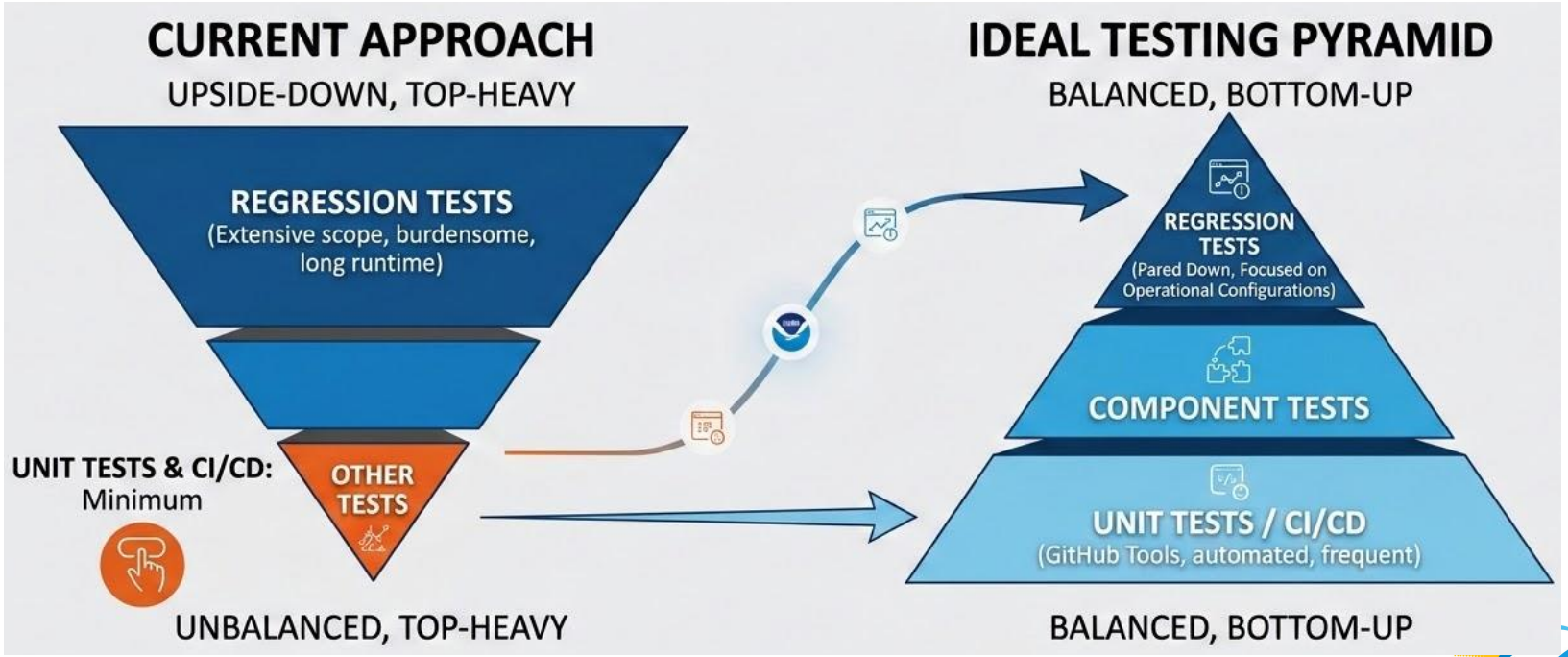
Regression and Unit Testing Rework

- Currently, every PR requires regression tests to be run to ensure that a multitude of configurations meet operational standards
  - -- 150 - 300+ regression tests are now run across 7 different systems
  - -- Runtimes can vary from 1.5 hours to 3 hours on different systems

- This causes delays for PRs and burdens both the code managers and the users

- The regression tests are a critical part of ensuring that the code maintains operational status and cannot be eliminated completely

Regression and Unit Testing Rework

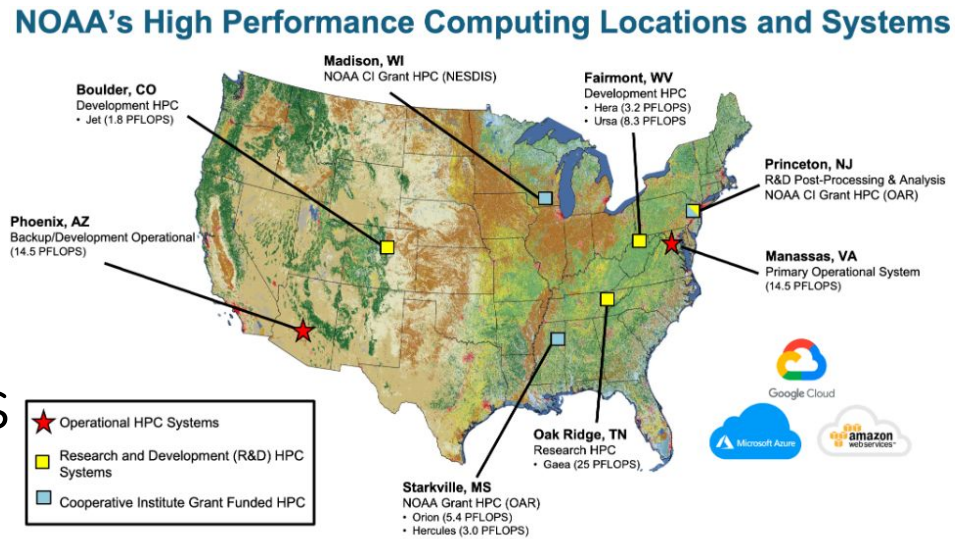


UIFCW26
A UFS Collaboration Powered by EPIC

Portability and Modularity

- The UFS WM is supported across several different systems

- If a user wants to use the UFS WM on a system that is not actively supported, then the building of libraries on those systems is a common roadblock



Portability and Modularity

- Reducing the reliance on external libraries and eliminating partial redundancies

- Discussing the benefits of versioned releases for UFS WM and its submodules

- Establishing UFS governance to collaborate with the community on supporting efforts to make code more portable

Ease of Use

- The ability for users to use the UFS WM for their experiments has become more difficult as the UFS WM has grown
  - Input file with 100s of variables, multiple modes and potential configurations
- Metrics and tools for helping the user with model stability and experiment creation are non-existent
- Although the UFS WM scripts and Global Workflow are robust, users might find them confusing requiring developer assistance to get started

Ease of Use

Community Feedback

Gathering community feedback via surveys to drive continuous improvements.

Toolsets & Utilities

Expanding the UFS WM toolsets to simplify workflow creation and model execution.

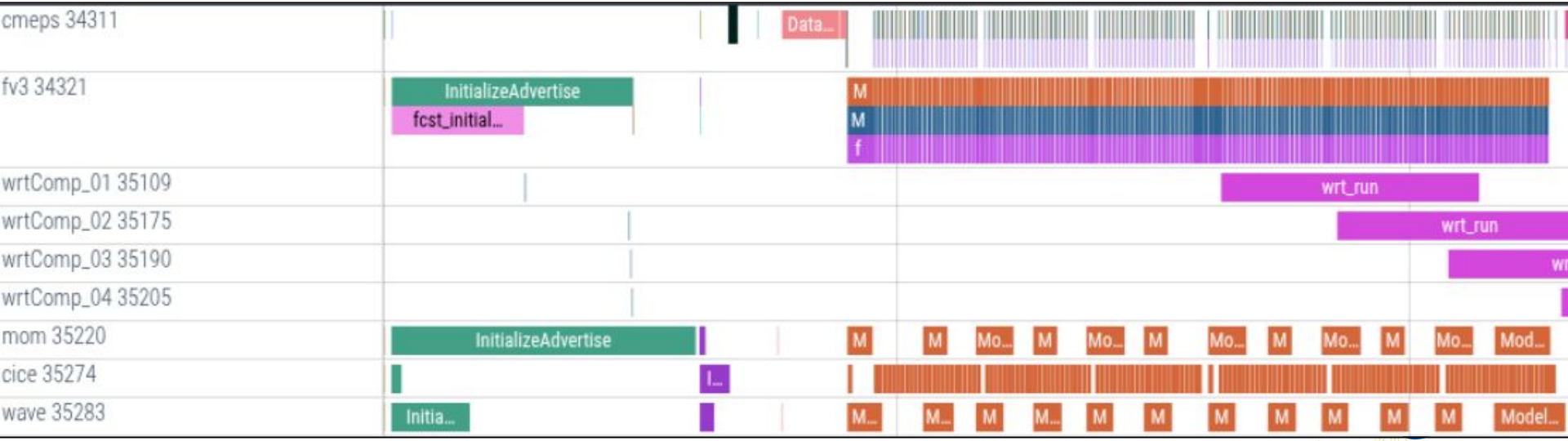
UIFCW26
A UFS Collaboration Powered by EPIC

Ease of Use

- Some basic yet powerful tools can be added to the UFS WM and SpackStack
- UFS Tracing will allow the user to see subroutine calls for the different components of the coupled model
    - Enables quick identification of bottlenecks and optimal core allocations
- Other lightweight tools for parallel program optimization (Scalasca) and memory management (Policeme) are being used with the UFS WM

Ease of Use

Example of UFS Tracing:



Using LLMs and other AI/ML methods

Global Workflow
MCP Tools (52 total)

1. Workflow Info (3 tools)

- `get_workflow_structure` - Structure and overview of the global-workflow system
- `get_system_configs` - HPC platform configs (Hera, Hercules, Orion, WCOSS2, Gaea)
- `describe_component` - Basic description of workflow components (file/directory)

2. Code Analysis (6 tools)

- `analyze_code_structure` - File structure, relationships, and dependencies
- `find_dependencies` - Upstream imports and downstream importers
- `trace_execution_path` - Execution path through call chains from a function
- `find_callers_callees` - Functions that call and are called by a target
- `trace_full_execution_chain` - Full chain across Shell → Fortran → Python
- `find_env_dependencies` - Scripts depending on/exporting environment variables

4. EE2 Compliance (5 tools)

- `search_ee2_standards` - Search EE2 compliance standards/documentation
- `analyze_ee2_compliance` - Analyze code/docs for EE2 compliance
- `generate_compliance_report` - Generate compliance report (summary/detailed/checklist)
- `scan_repository_compliance` - Batch-scan files for EE2 issues
- `extract_code_for_analysis` - Extract bash/python snippets for compliance

5. Operational (4 tools)

- `get_operational_guidance` - HPC operations guidance + platform notes
- `explain_workflow_component` - Component explanation with graph context
- `list_job_scripts` - List/categorize J-Job scripts
- `get_job_details` - J-Job inputs, outputs, configs, env vars

- A project is currently underway to use LLMs to build a RAG and agentic tools from the UFS Global Workflow repository

- Use cases include agentic assisted coding, refactoring for GPUs and code optimization, validation with implementation standards, and diagnosing model failures

UIFCW26
A UFS Collaboration Powered by EPIC

Communication is key!

- Many of the changes we are enacting are due to user and developer feedback. We are sending out surveys to gather more data.

- We are looking to learn from other institutions and other community modeling groups to learn about what works and what doesn't.

- Ability to create touch-points or collaborative efforts between NOAA EMC and other institutions is important, especially when there are many shared components and goals.

Email: daniel.sarmiento@noaa.gov

Link to the UFS Survey

https://forms.gle/rs5ywqaEskgm6nVa6

UIFCW26
A UFS Collaboration Powered by EPIC