

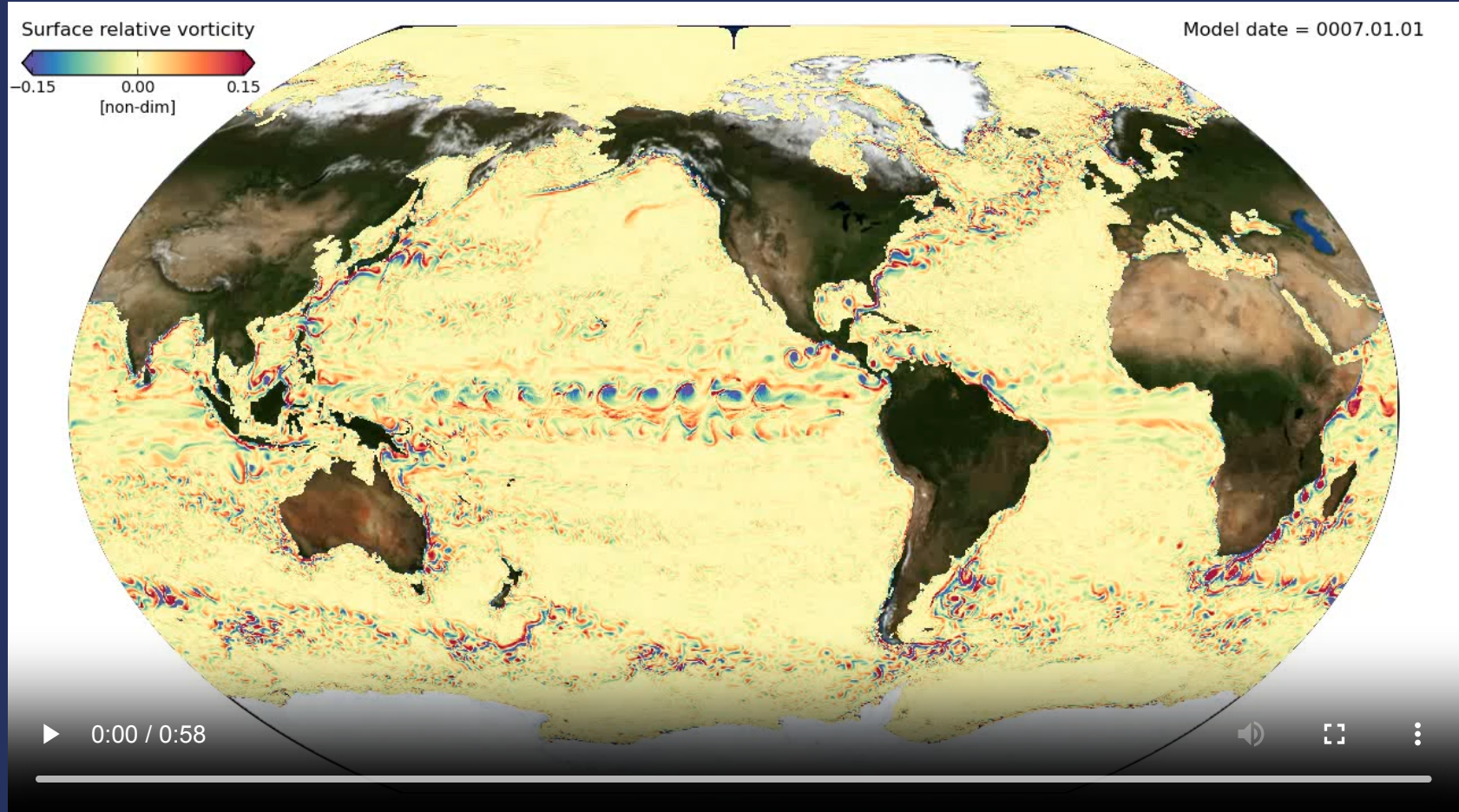
# THE MOM6 DEVELOPMENT CYCLE

- Marshall Ward
- Robert Hallberg
- Alistair Adcroft

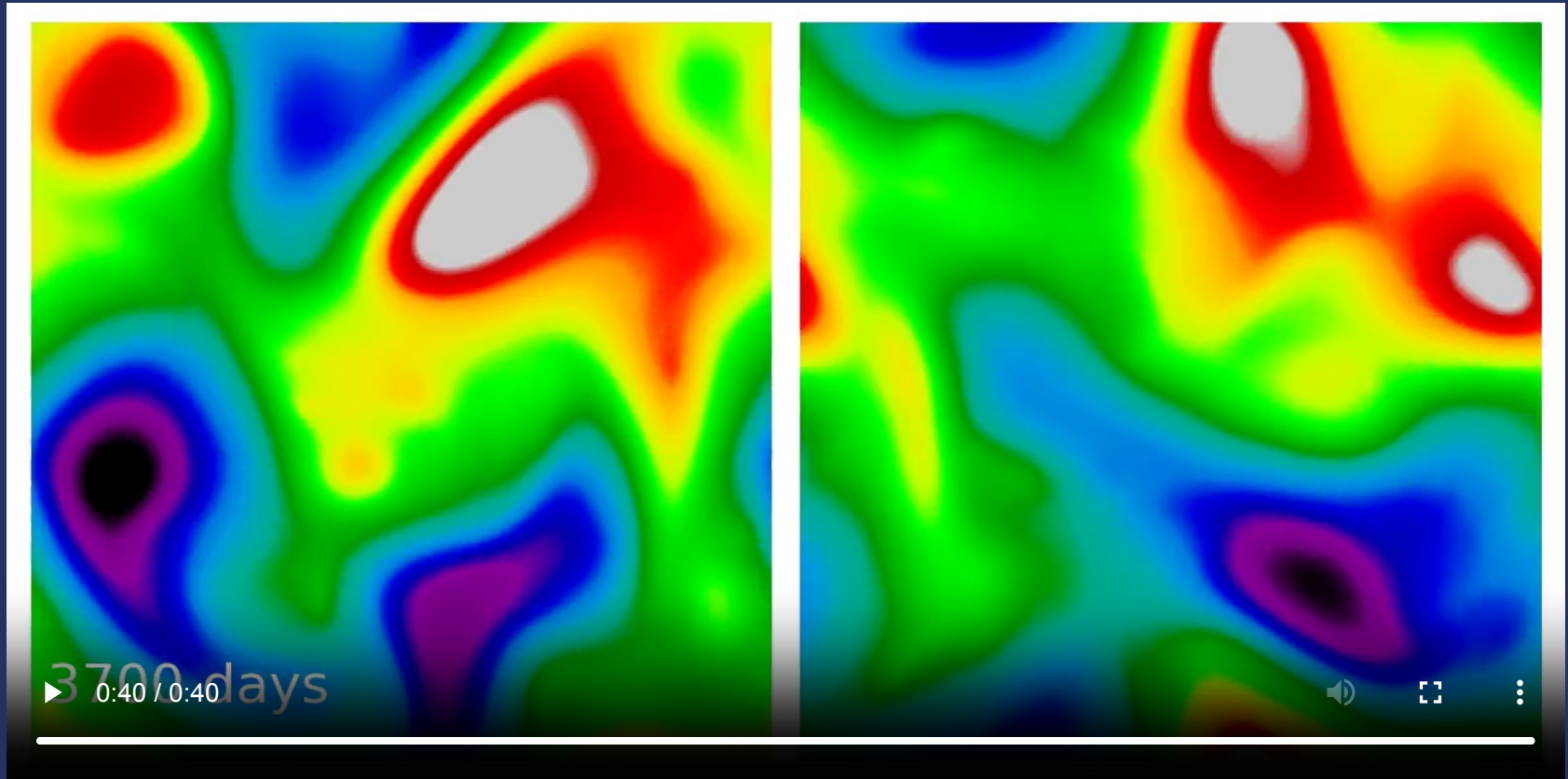
NOAA-GFDL / Princeton U.

2023-11-09

# MOM6 OCEAN MODEL



# MOM6 SENSITIVITY



Minor deviation leads to  $O(1)$  difference

# REPRODUCIBILITY RULES

- Numerical expressions must be *bit-reproducible*
- Code must not change existing solutions
- Changes must not disrupt existing research

# WHAT WE *DON'T* REQUIRE

When can answers differ?

- Aggressive optimization (`-O3 -mavx ...`)
- Hardware (Intel, AMD, A64FX, ...)
- Compilers (gcc, ifort, ...)
- Libraries (libm.a, ...)

But we still try to maximize reproducibility.

# FLOAT ORDER-OF-OPERATIONS

What is  $10^{-16} + 1 - 1$ ? Ambiguous!

$$(10^{-16} + 1) - 1 \equiv 0$$

$$10^{-16} + (1 - 1) = 10^{-16}$$

Residuals below ULP  $\left(2 \times 10^{-16}\right)$  are lost.

*Use parentheses to set the order of operations*

# TRANSCENDENTALS

How is  $\sin(x)$  computed? Ambiguous!

$$f(48^\circ) = 2\Omega \sin\left(\frac{48\pi}{180}\right)$$

```
glibc 2.22: 0.108381727637274115E-03 (3F1C695FE71A3FE4)  
2.26: 0.108381727637274128E-03 (3F1C695FE71A3FE5)
```

*Avoid transcendentals where possible, manage dependencies when necessary.*

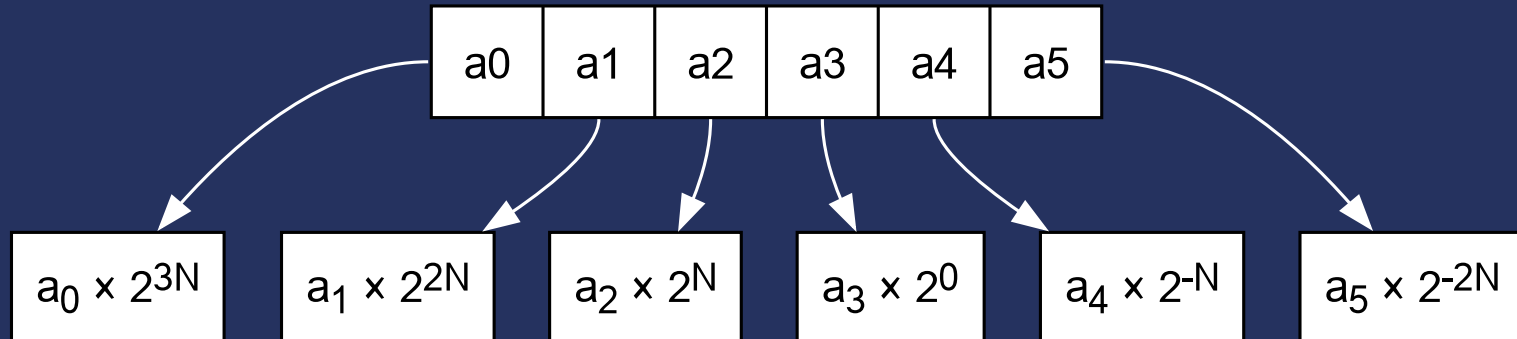
# GLOBAL SUMMATION

The order of `sum()` is ambiguous!

Parentheses work, but may have cumulant errors:

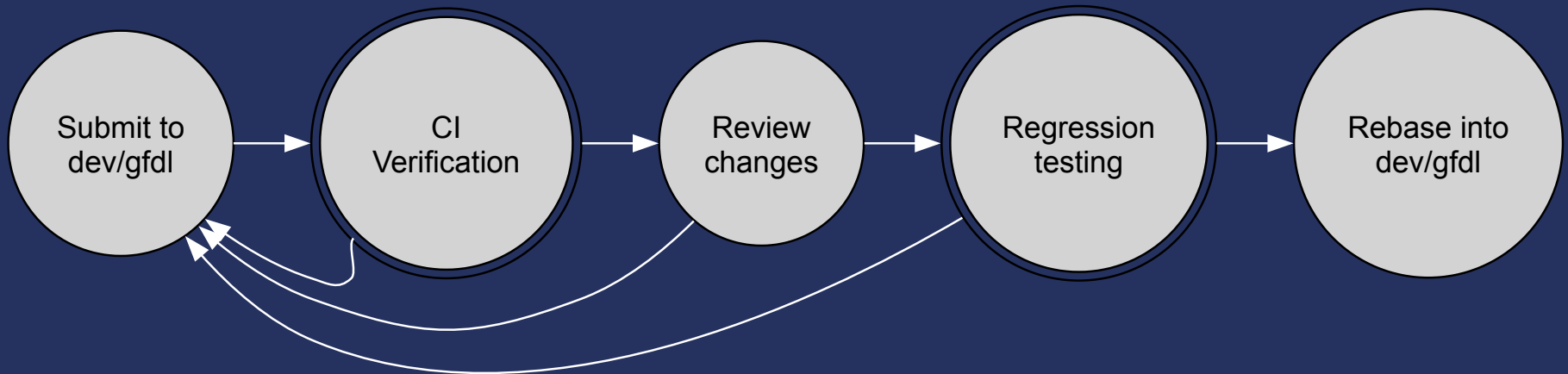
$$\sum \phi = \phi_1 + (\phi_2 + (\phi_3 + \dots))$$

Sums are stored in fixed-precision using six integers:



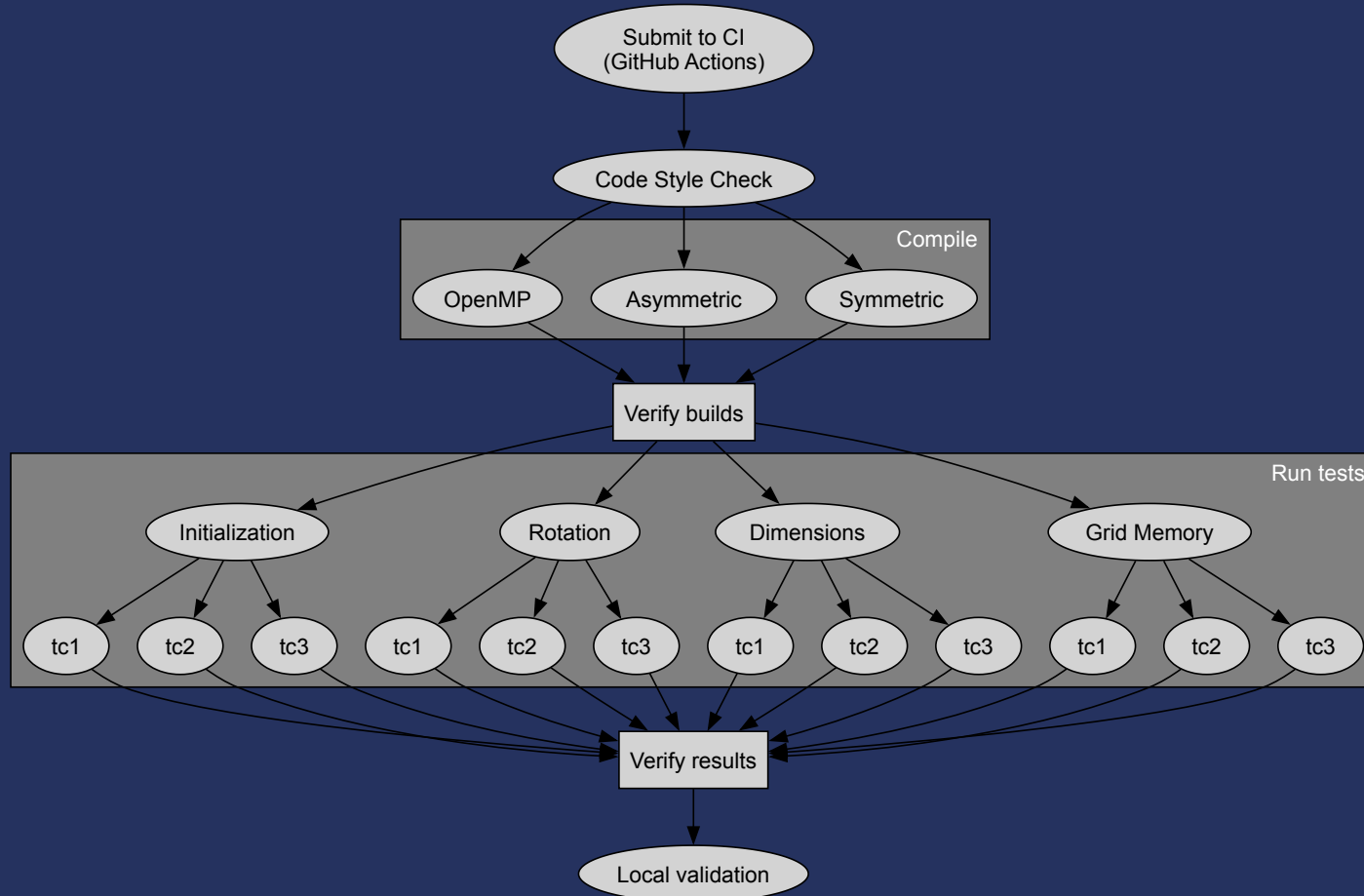


# GFDL PR LIFE CYCLE



1. User submits to node by PR
2. Automated verification testing
3. Code review (by a human)
4. Test for regressions on production machine
5. Rebase into node codebase

# VERIFICATION TESTING



CI testing is *platform independent*

# STATE OF THE MODEL

Global energy tracked to full precision

Step	Days	Energy/Mass [m2 s-2]	Mean Sea Level [m]	...
0	0.00	7.2161166068132286E-27	1.8190E-12	...
12	0.50	2.7781004671136538E-04	1.1369E-12	...
24	1.00	2.7734897826598717E-04	1.8190E-12	...

Diagnostics use multiple metrics

```
u-point: ocean_model-u
  min  = -6.7187595818683776E-03  max  =  3.3480219779204019E-02
  mean =  1.1239682303793666E-04  bits = 21851
v-point: ocean_model-v
  min  = -8.3469699425156359E-03  max  =  6.8420831486068704E-03
  mean =  1.2076392816784489E-03  bits = 18606
h-point: ocean_model-h
  min  =  9.999999999999915E-04  max  =  5.6265092225099863E+02
  mean =  3.6490088139048595E+02  bits = 18673
...
```

# EXAMPLE TESTS

## "Local regression"

New code does not (inadvertently) change answers

## Parallel Layout

1×2 and 2×1 domain decompositions

## Restart/Pickup

One  $\Delta t$  run must equal two  $\frac{1}{2}\Delta t$  runs

## Aggressive initialization

NaN-initialization arrays vs. uninitialized arrays

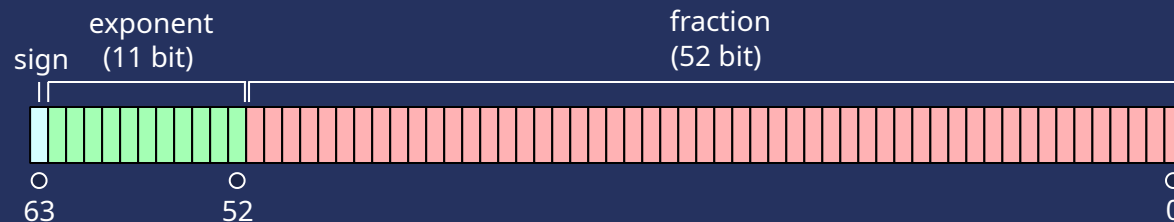
# DIMENSION TEST

Dimensionally correct equations are invariant to scaling:

$$u^{n+1} = u^n + \Delta t \times F$$
$$2^{L-T} u^{n+1} = 2^{L-T} u^n + 2^T \Delta t \times 2^{L-2T} F$$

Solutions should also be invariant

# FLOATING POINT REVIEW



$$\phi = (-1)^s \times 2^M \times (1 + \alpha)$$

- Smallest fractional diff:  $2^{-52} \approx 2.2 \times 10^{-16}$
- 17 digits to uniquely specify a result

# SHALLOW WATER EXAMPLE

$$u_t + uu_x + vu_y = -gh_x$$

$$v_t + uv_x + vv_y = -gh_y$$

$$h_t + hu_x + hv_y = 0$$

Dimensions:

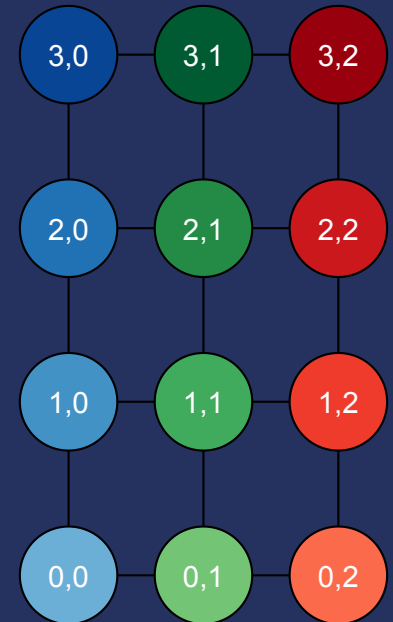
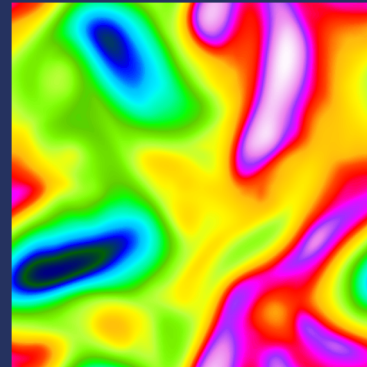
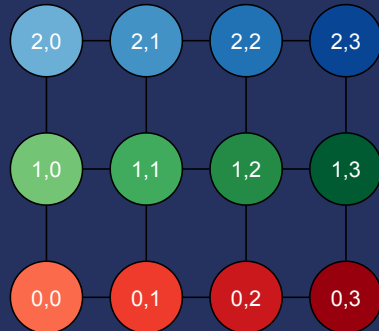
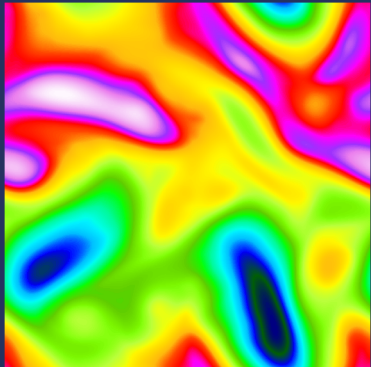
- $L$  (horiz. length)
- $T$  (time)
- $H$  (layer depth)

Invariants:

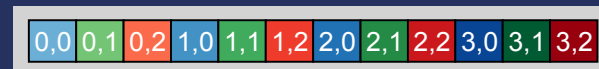
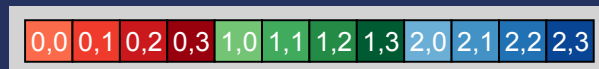
- $[u, v] = LT^{-1}$
- $[g] = L^2 H^{-1} T^{-2}$

# ROTATION TEST

Equations should be invariant to rotation



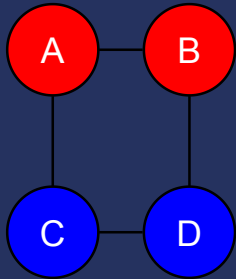
Perhaps should be called an "index" rotation





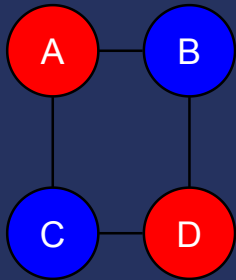
# INVARIANT STENCILS

$$\phi_{i,j}^{(c)} = \frac{1}{4} (\phi_A + \phi_B + \phi_C + \phi_D)$$



$$\frac{1}{4} ((\phi_A + \phi_B) + (\phi_C + \phi_D))$$

$$\frac{1}{4} ((\phi_A + \phi_C) + (\phi_B + \phi_D))$$



$$\frac{1}{4} ((\phi_A + \phi_D) + (\phi_B + \phi_C))$$

# DESCRIPTIVE COMMITS

```
commit d210cc6cdfd03150306c8ba41612e3380d66c281
Author: Robert Hallberg <Robert.Hallberg@noaa.gov>
Date:   Wed Aug 16 17:11:35 2023 -0400
```

```
+Remove build_grid_arbitrary
```

```
Removed the unused (and unusable) routine build_grid_arbitrary.
This routine could not have been used because it had a hard-coded
STOP call, and comments in it indicated that it should have been
deleted in July, 2013. The run-time parameter setting that would
have triggered a call to this routine has been retained for now,
but with a fatal error message explaining that this routine has
not been implemented. All answers are bitwise identical in any
cases that ran before.
```

```
commit 615e57f854db8be8c75a9edba6bb05e3f04a6eb7
```

*Also: No "fixed my typo" commits!*

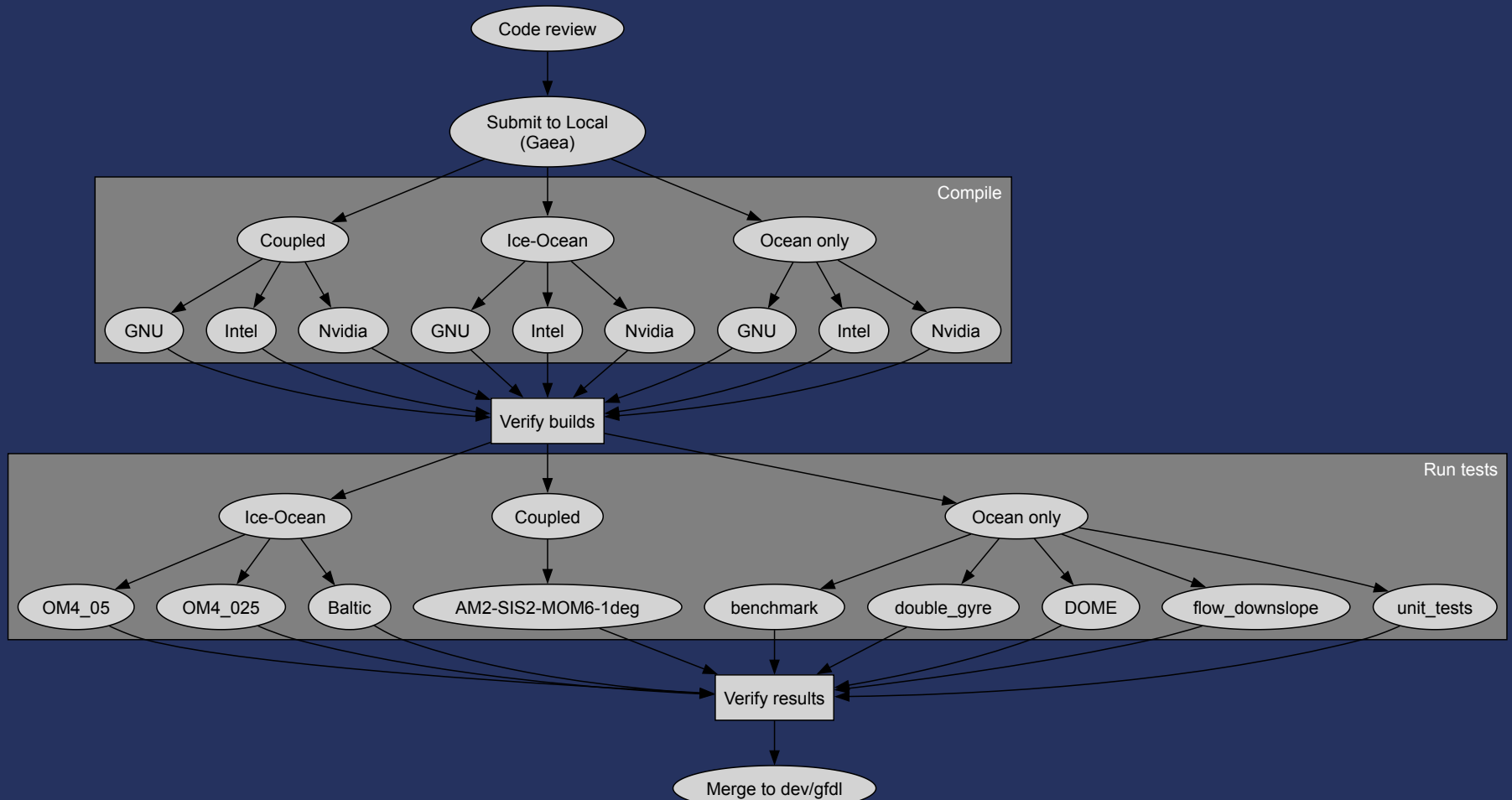


*Commit messages to me are almost as important as the code change itself.*

*If you can explain the code change to me, I will trust the code.*

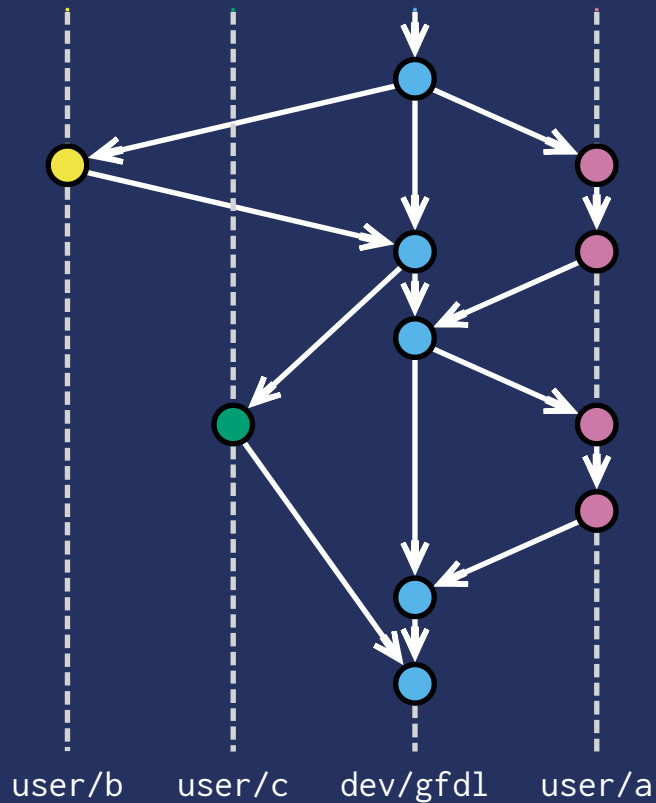
Linus Torvalds, Linux OSS 2020

# REGRESSION SUITE



Validate 61 tests using 3 compilers

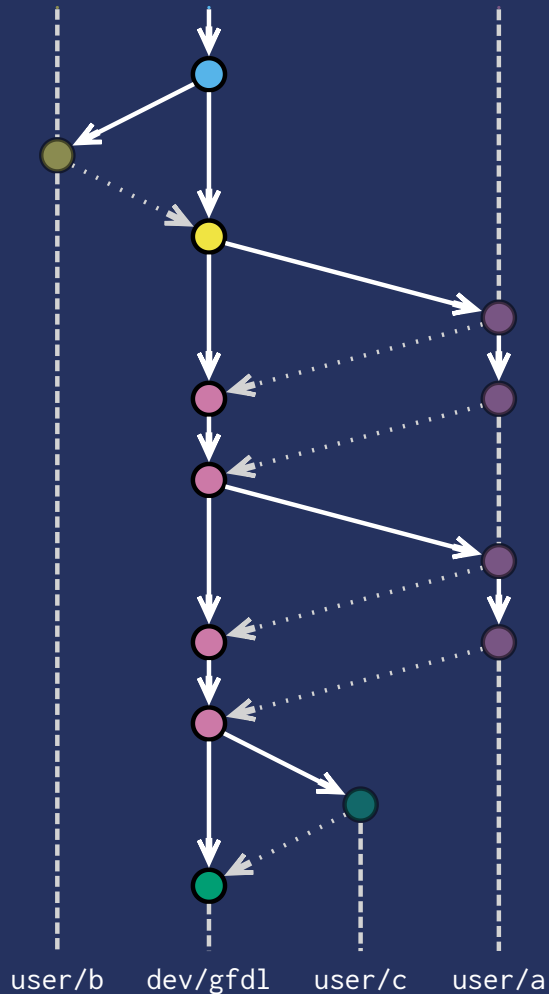
# TANGLED HISTORY



Having many active users leads to a non-sequential history

This severely complicates our ability to track bugs and regressions!

# REWRITE HISTORY!



We aggressively rebase to create a new linear history

However, note that we also clobber user history

We *cannot* clobber external research codebase!

How to preserve existing runs?

# MOM6 CONSORTIUM



# WHAT IS THE MOM6 CONSORTIUM?

Codebase is governed by a consortium of research groups.

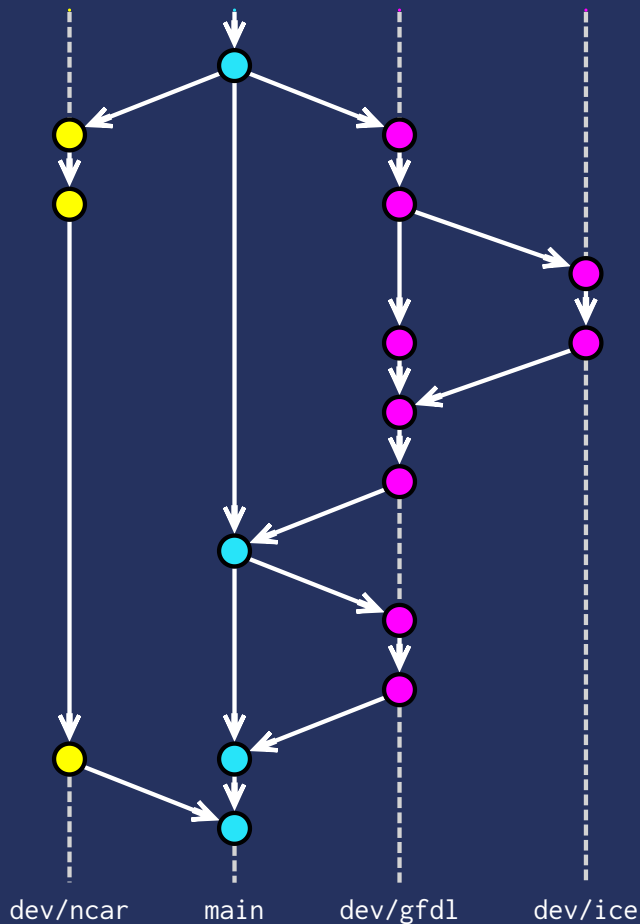
All changes to the "hub" must be tracked and preserved.

Groups manage their own branch, and contribute to main.





# CODE CONTINUITY



- Merges to main are **unanimous**
- All branches within a node are preserved
- Nodes can easily collaborate on joint projects

# SUMMARY

## Bit reproducibility?

Restrict model to *non-ambiguous* operations

## Preserve existing solutions?

*Regression* and *self-consistency* testing

## Code preservation?

Mutual governance by a *consortium* of groups