MIRGE-Com

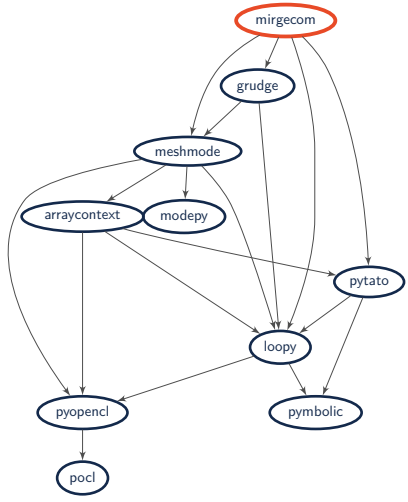
Y3 Scaling Status

Fr@CEESD - October 13, 2023

Mike Campbell & CEESD

Outline

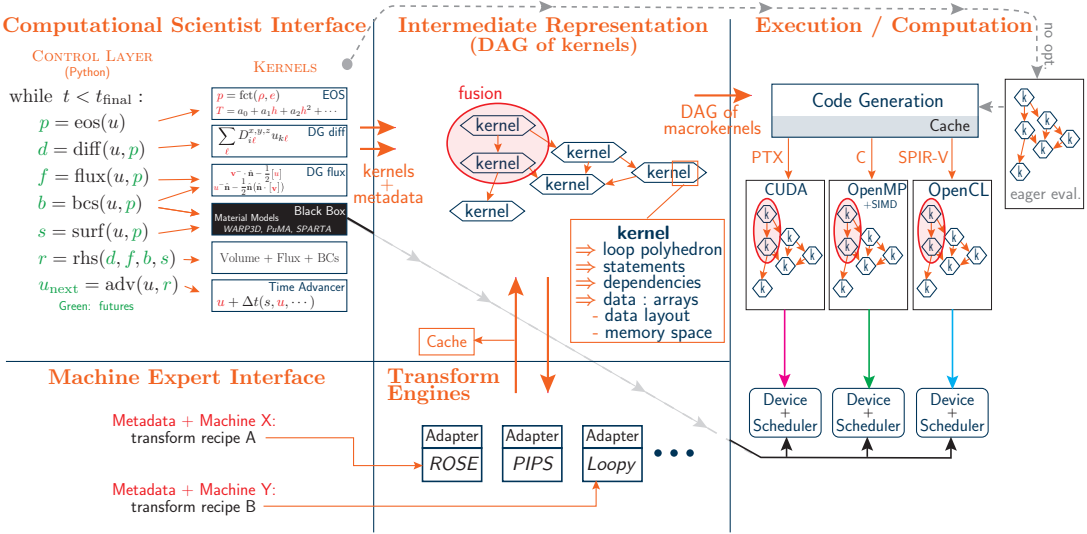
- MIRGE-Com Overview
- Performance & Scalability
- Recent Challenges
- Conclusion & Future Directions



https://github.com/illinois-ceesd/mirgecom/

MIRGE-Com in Y3

Architecture Overview

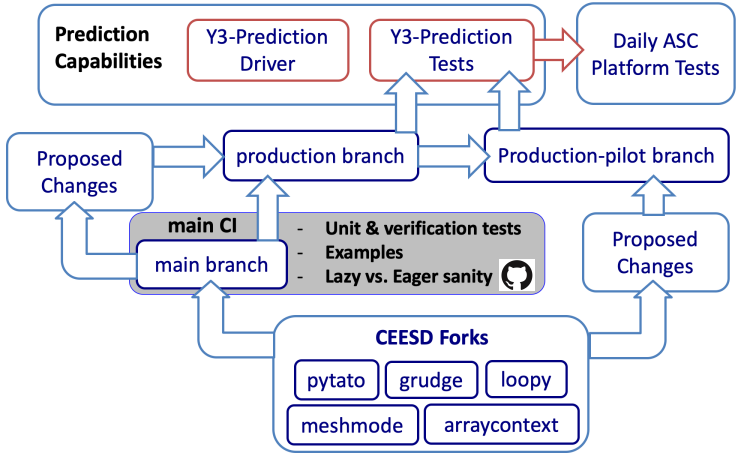


Simulation Infrastructure

- Infrastructure provides:
  - DG operators (e.g., mass, stiffness, grad, div)
  - Multiple parallel discrete geometries [M. Smith]
  - Compute device access
  - Symbolic infrastructure - great for verification
- *MIRGE-Com* library provides:
  - Conservation-law-specific data structures
  - Simulation / driver API
  - Prediction-relevant
    - RHS operators
    - Model-specific constructs (e.g., EOS, transport, reactions)
    - Boundary and numerical fluxes

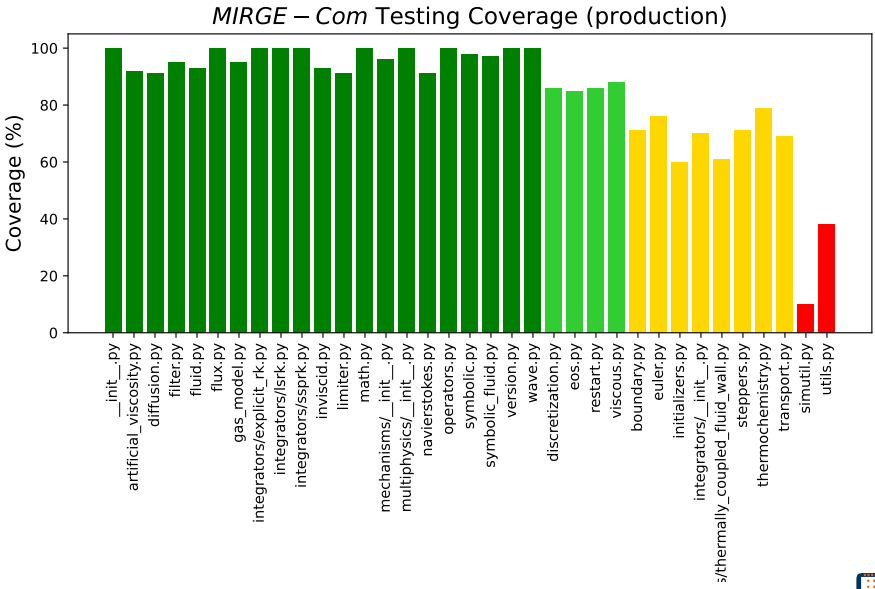
```
def euler_operator(discr, eos, boundaries, cv, time=0.0):
    inviscid_flux_vol = inviscid_flux(discr, eos, cv)
    inviscid_flux_bnd = (
        inviscid_facial_flux(discr, eos=eos, cv_tpair=interior_trace_pair(discr, cv))
        + sum(inviscid_facial_flux(discr, eos=eos,
                                   cv_tpair=cross_rank_trace_pairs(discr, cv)))
        + sum(boundaries[btag].inviscid_boundary_flux(discr, btag=btag, cv=cv,
                                                      eos=eos, time=time)
              for btag in boundaries)
    )
    q_rhs = discr.inverse_mass(weak_local_div(discr, inviscid_flux_vol)
                               - discr.face_mass(inviscid_flux_bnd))
    return make_conserved(discr.dim, q=q_rhs)
```

Prediction-supporting Development



- Prediction-supporting development process
- Prediction-targeted testing mechanism
- Continuous integration and daily ASC platform testing

Prediction supporting Development



Performance & Scalability

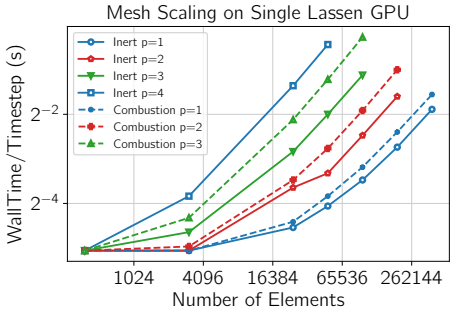
Performance & Scalability

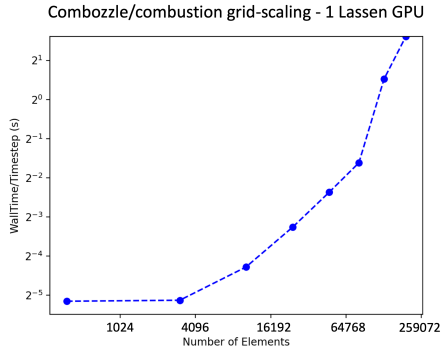
- Computational performance: Speed or FLOPS
- Parallel performance: Scalability or efficiency
  - Strong scaling: Fixed problem size, scaling resources
  - Weak scaling: Fixed work/resource, scaling problem size with resource
  - Mesh scaling: Scaling work, fixed resource
- I/O and memory performance: Bandwidth, bytes/second
- Cost performance
  - Energy efficiency
  - Total time to solution (TTS)
  - FLOPS/Fidelity
  - Usability/Productivity
- Why should we care about scalability? (enables prediction!)
  - Weak scaling out to prediction-scale
  - Versatility, portability, and resource options

Performance & Scalability

New: prediction-enabling performance

- Scaling as expected (mostly)
- Small problems are expensive
- OOM: SVM/Unified memory
- Mem growth: Garbage collection

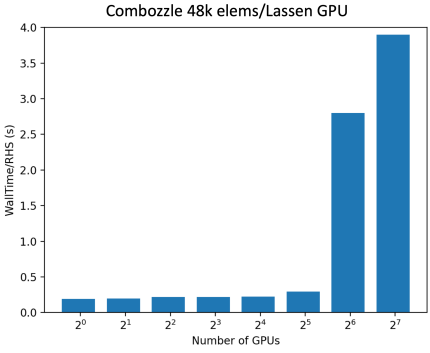




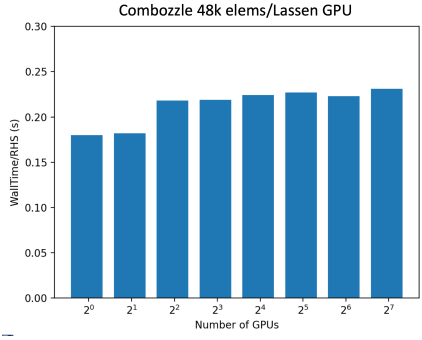
DAG Splat Effect

New: prediction-enabling performance

- DAG Splat: DAG for each boundary
- Limits weak scaling - DAG for each neighbor



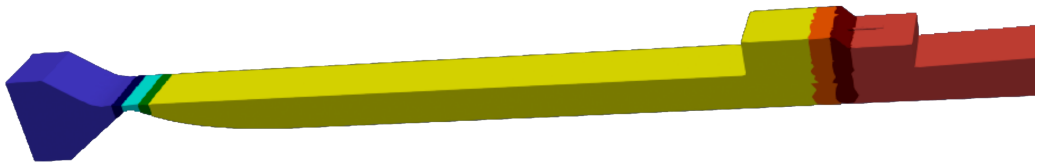
- Mitigation: Metis $\rightarrow$ 1D decomp
- Real fix: Function calls in the DAG (aka function outlining)



DAG Splat Mitigation

New: prediction-enabling performance

- DAG Splat: DAG for each boundary
- Limits weak scaling - DAG for each neighbor
- Mitigation: Metis $\rightarrow$ 1D decomp
- Real fix: Function calls in the DAG (a.k.a. outlining)

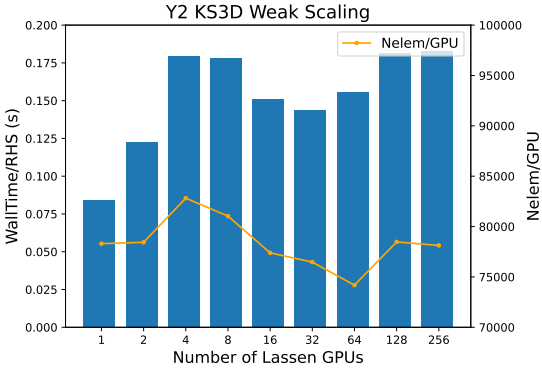


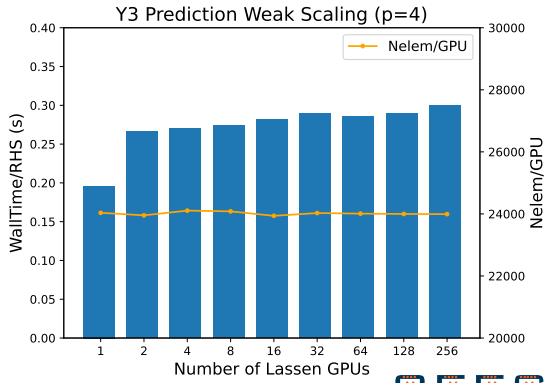
Y2 Domain Decomposition

Scaling with 1D Partitioning

New: prediction-enabling performance

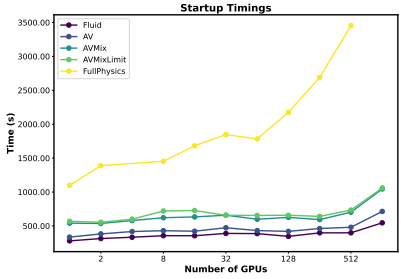
- DAG Splat: DAG for each boundary
- Limits weak scaling - DAG for each neighbor
- Mitigation: Metis vs. 1D decomp
- Real fix: Function calls in the DAG (a.k.a. outlining) [M. Smith]

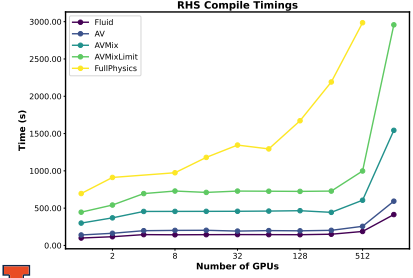


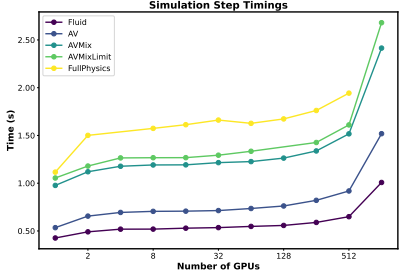


Prediction Performance Snapshots: Lassen DAT

- Weak scaling: 24k $p4$ per GPU
- Fluid: NS-only, 2 species
- AV: Fluid + Artificial Viscosity
- AVMix: AV + 7 species mixture EOS
- AVMixLimit: AVMix + species limiter
- FullPhysics: AVMixLimit + PLTransport + Wall

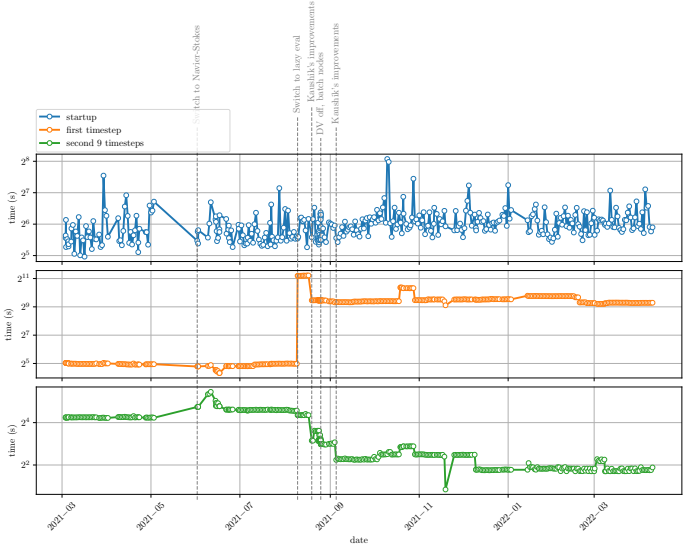






Performance Monitoring on Lassen

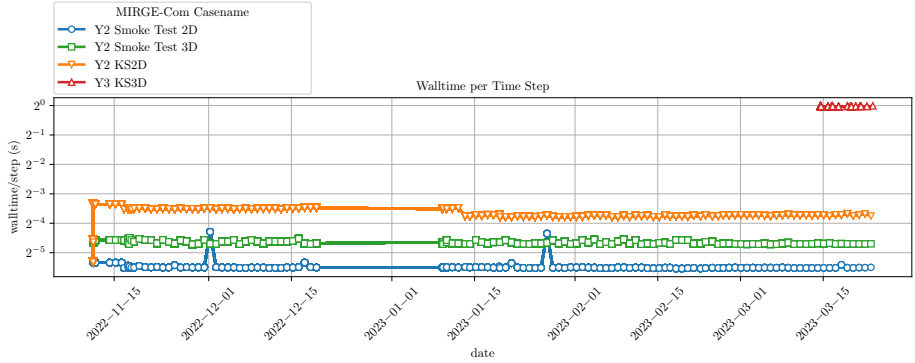
https://github.com/illinois-ceesd/timing



Performance Monitoring on Lassen

https://github.com/illinois-ceesd/timing

- Data on key capabilities collected nightly
- Intended to track performance vs. code/features
- Δ's indicate change in performance
- New this cycle:
  - Multi-case/comparitive plotting
  - Tracking parallel cases

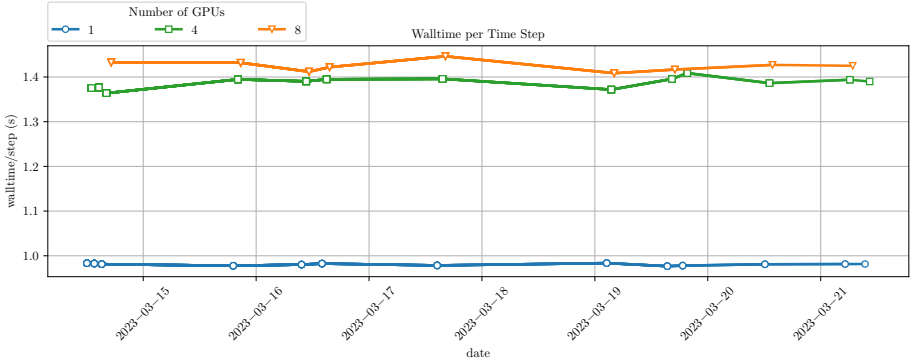


Performance Monitoring on Lassen

https://github.com/illinois-ceesd/timing

- Data on key capabilities collected nightly
- Intended to track performance vs. code/features
- Δ's indicate change in performance

- New this cycle:
  - Multi-case/comparitive plotting
  - Tracking parallel cases

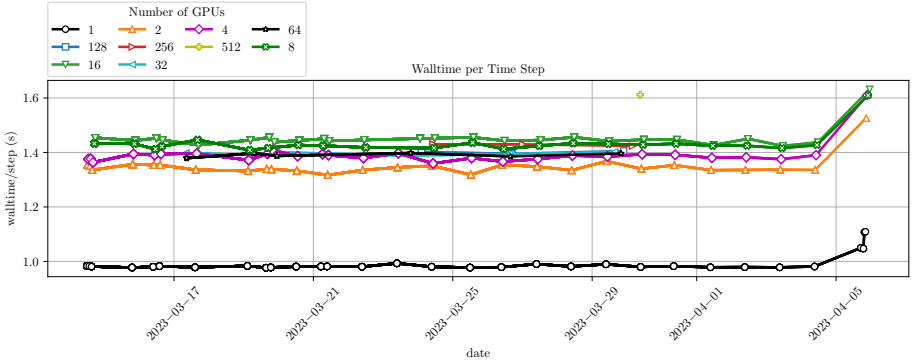


Performance Monitoring on Lassen

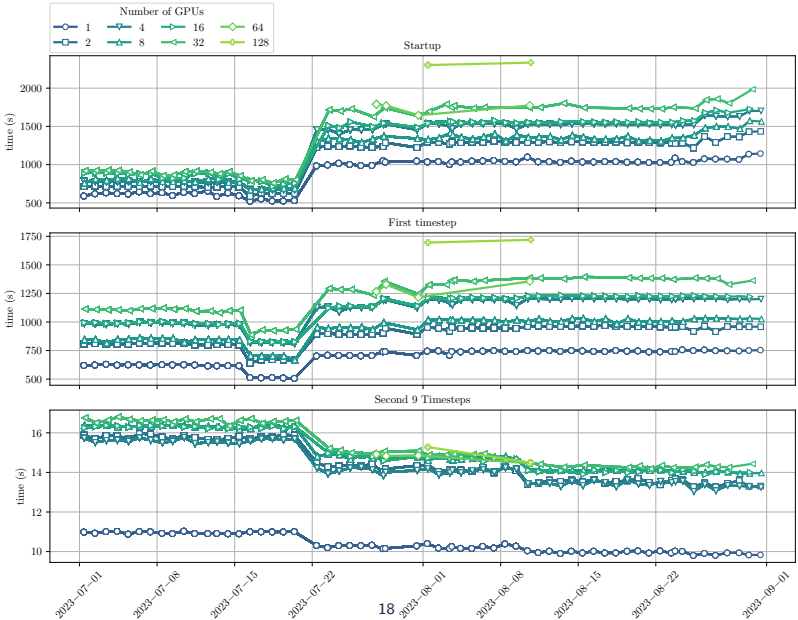
https://github.com/illinois-ceesd/timing

- Data on key capabilities collected nightly
- Intended to track performance vs. code/features
- Δ's indicate change in performance

- New this cycle:
  - Multi-case/comparitive plotting
  - Tracking parallel cases



Recent Y3 Monitoring on Lassen



Recent Challenges

Recent Challenges and Looming Threats

- Loop nest error and indeterminism [M. Diener, M. Smith]
- DAG splat - 1D part seen as looming threat
- Super long compile times for essential models
- Mesh I/O and processing
  - Mesh distribution fell down at scale (mpi4py@pkl5) [M. Diener]
  - Gmsh ingest uses far too much memory
    - Prevented prediction scaling past 512 ranks, precluded machine-scale runs
    - Pre-partitioning work-around gets us to 1024, will 2048 test today!
- M-to-N restart: needed for simulation portability (now working for prediction)

Wrapping Up & Looking Ahead

Summary and Next Steps

- *MIRGE-Com* has Y3 prediction-supporting performance (poised to deliver more)
- Understanding *MIRGE-Com* performance is the next major focus

Next steps

- Understanding and improving performance:
  - Instrumentation (Mem & Tags) [M. Diener]
  - Code-to-kernel correspondence improvements: [M. Diener]
  - Auto-tuning [Nick Christensen]
  - DAG Splat [M. Smith]
  - Performance model
- Upcoming enhancements:
  - Workflow: *Parsl* [D. Friedel]
  - Hexahedral elements [Addison Alvey-Blanco]
  - M-to-N restart (done!)

Questions?

This material is based in part upon work supported by the Department of Energy, National Nuclear Security Administration, under Award Number DE-NA0003963.