Ristra, FleCSI, and Legion: multi-physics on unstructured meshes

Jonathan Pietarila Graham, Ben Bergen, Davis Herring, Kevin Larkin,
Chris Malone, Maxim Moraru, Scott Pakin, Nathan Vaughn-Kukura

Dec. 4, 2024

LA-UR-24-32213

Ristra

Research Project for Lagrangian Hydrodynamics codes

Moya – Ristra's low energy-density multiphysics code
Lagrangian staggered grid hydro
Features (May 2023)

Hydro

- Conservative staggered grid hydro
  - Kinematic variables at vertices
  - Material state variables at cell centers
  - 2^{nd} order Runge-Kutta
  - Unstructured mesh
  - General polyhedral cells
- Artificial/shock viscosity
  - Von Neumann – Richtmyer

Mesh Management

- Mesh stiffener
  - Temporary Quadrilateral Subzoning (TQS)
- Mesh relaxation
  - Feasible set method
  - Corner-angle controller
  - Per-material controller

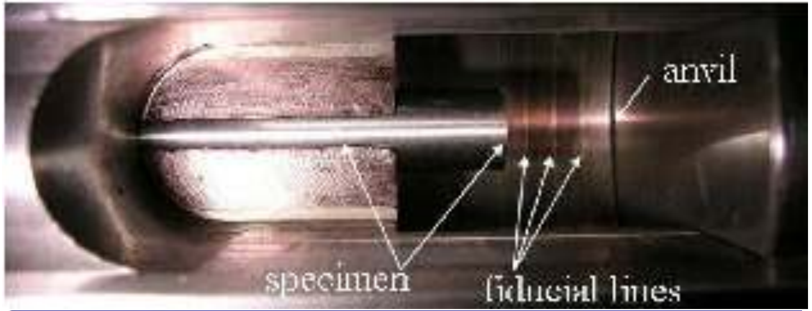
Material Models

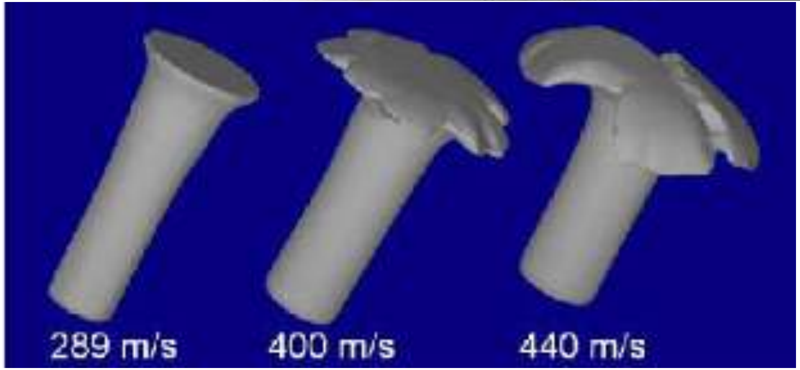
- Gases
  - EOS: gamma law gas
- Solids
  - Strength: Steinberg-Guinan, linear hardening and Preston-Tonks-Wallace (PTW)
  - EOS: Gruneisen
  - Melt: Lindemann
- High Explosives
  - EOS: Jones-Wilkins-Lee (JWL)
  - Point-detonated
  - Direct lighting

Moya capabilities

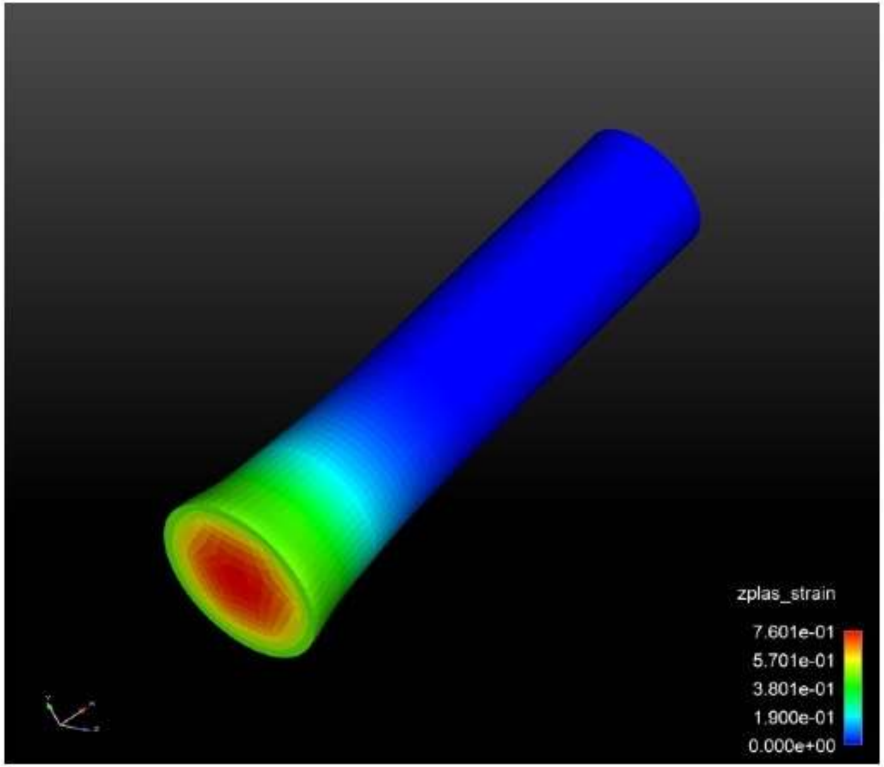
strength modeling – Taylor Anvil Impact

Impact of a projectile against a rigid boundary measures dynamic material properties





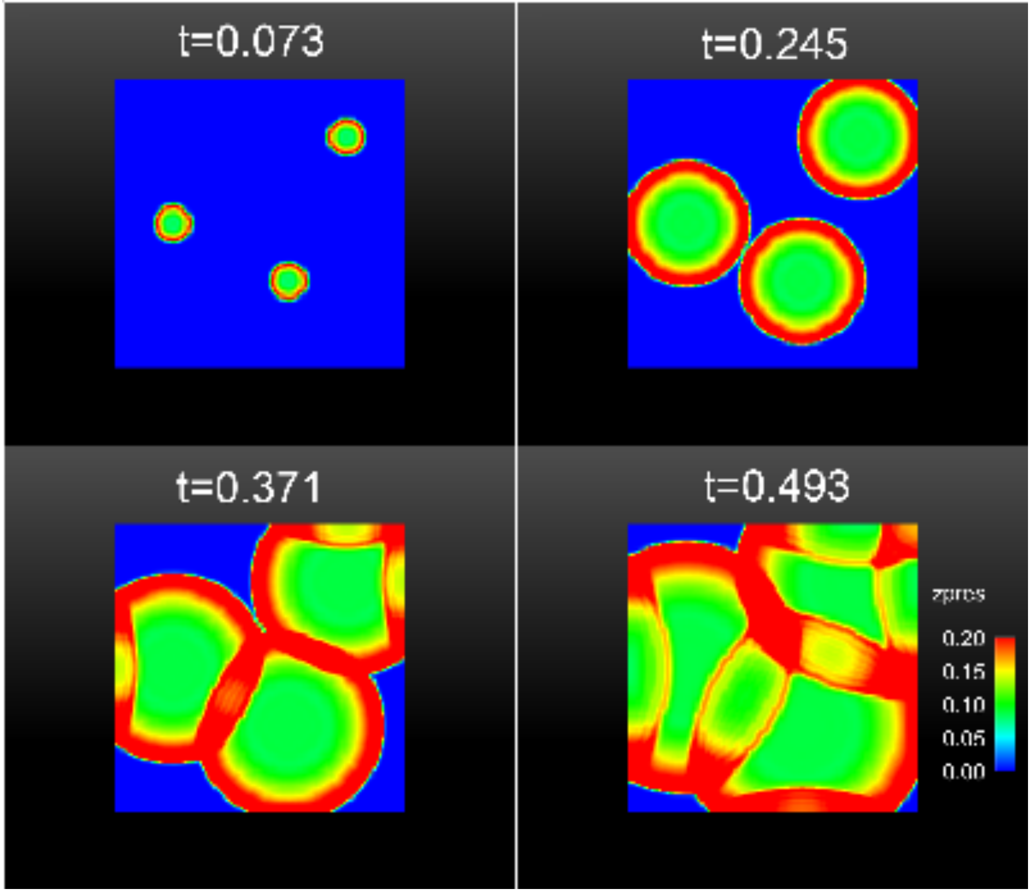
Anderson et al., CP845, Shock Compression of Condensed Matter - 2005



Moya: plastic strain in a Taylor anvil impact simulation

Los Alamos

Moya capabilities
high explosives



Simulation of a high explosive material

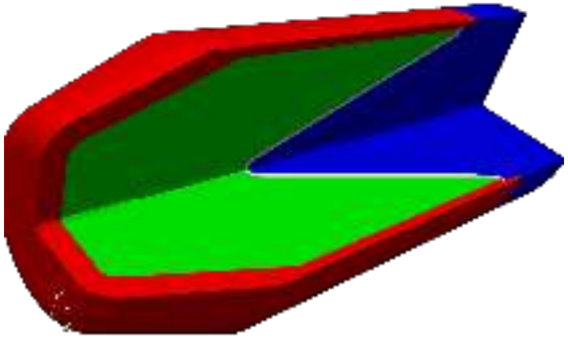
- Three point-detonators

Demonstration Problem
shaped charge

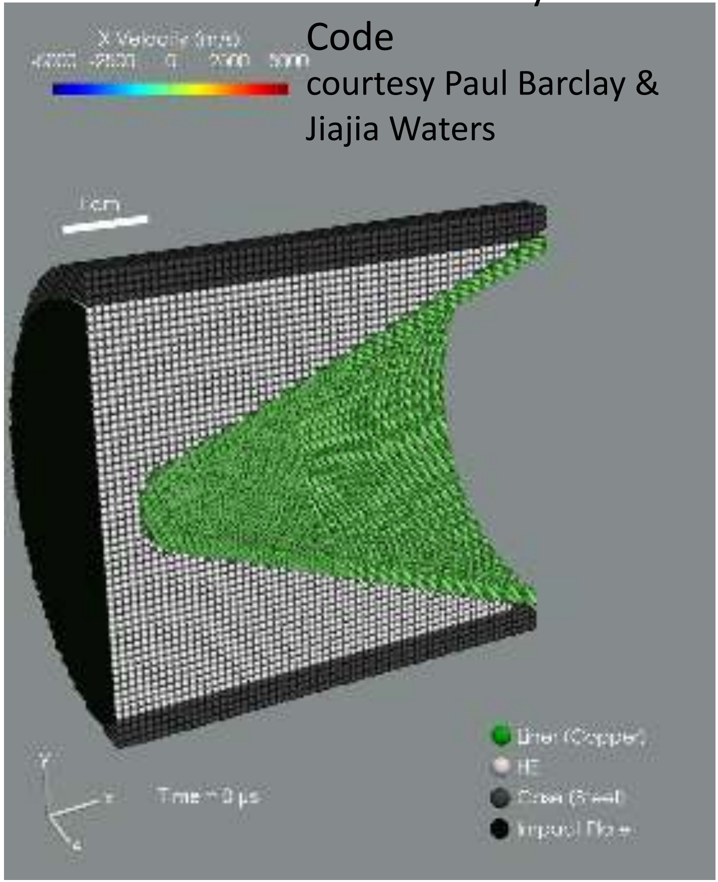
CartaBlanca
Multi-Velocity Particle Code
courtesy Paul Barclay & Jiajia Waters

Focused explosive charge

- cutting and forming metal
- penetrating armor
- perforating oil & gas wells



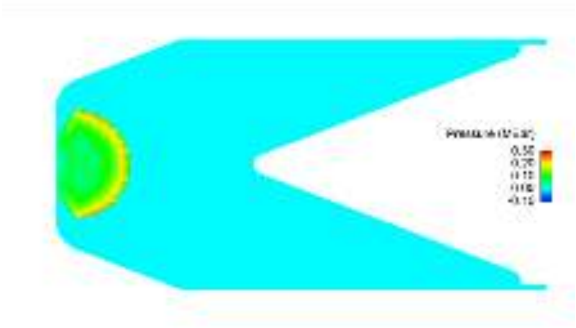
shell (steel)
explosive
liner (copper)
air



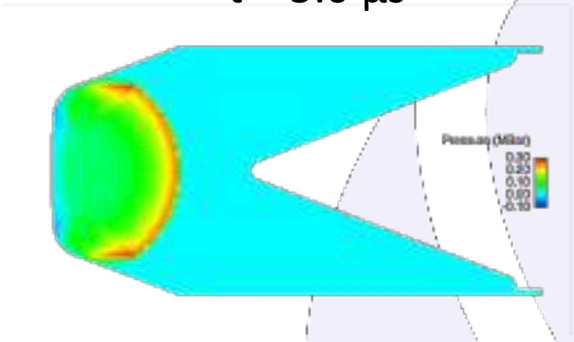
Moya results
Jet begins forming

propagating shock
(pressure)

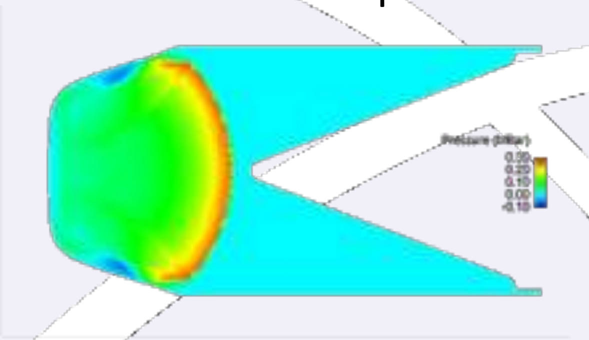
t = 2.5 μs



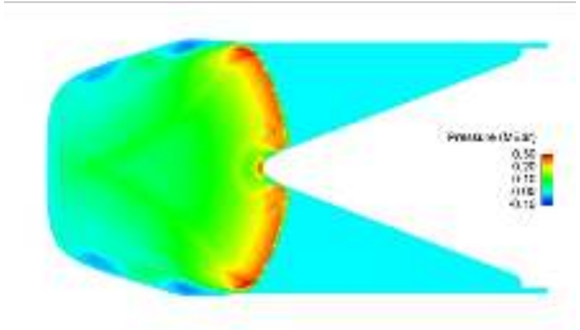
t = 5.0 μs



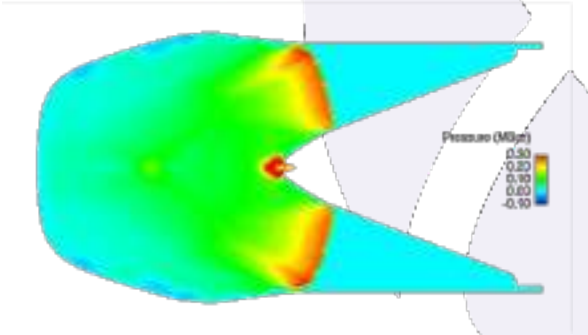
t = 7.5 μs



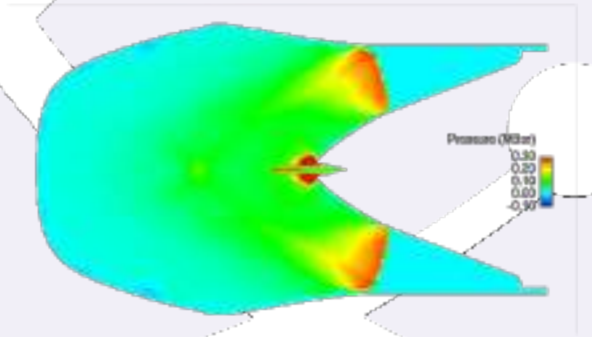
t = 10.0 μs



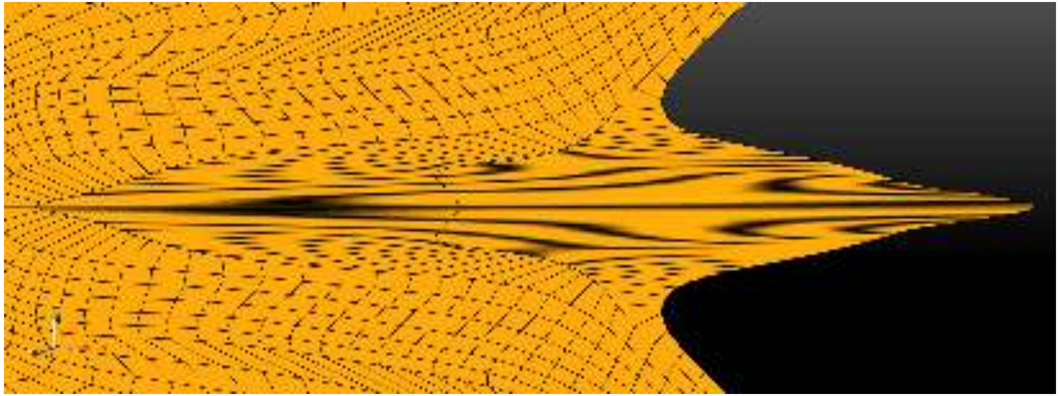
t = 12.5 μs

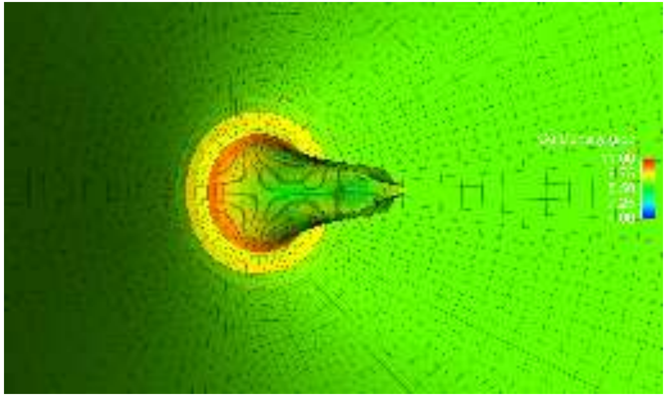


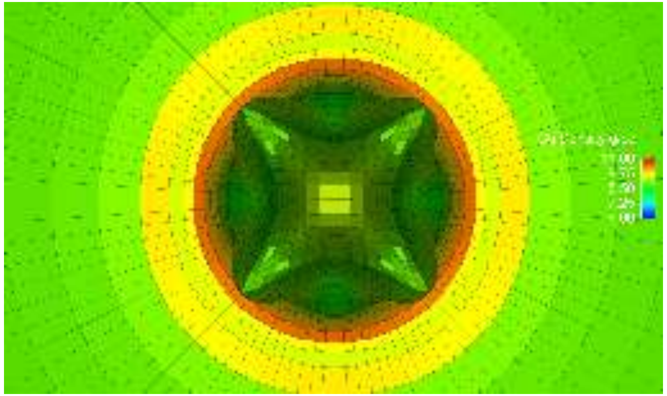
t = 15.0 μs



Moya results







Top left: 2D slice through the jet

Right: 3D views of the jet.

FleCSI

Flexible Computer Science Infrastructure

FleCSI 2.0: The Flexible Computational Science Infrastructure Project

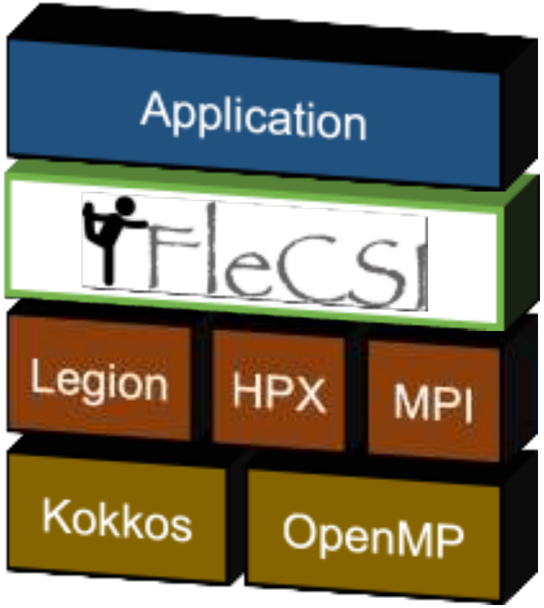
Bergen, Ben ; Demeshko, Irina ; Ferenbaugh, Charles ; Herring, Davis ; Lo, Li-Ta ; Loiseau, Julien ; Ray, Navamita ; Reisner, Andrew, Euro-Par 2021: Parallel Processing Workshops, 2022, p.480-495

https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-21-25604

FleCSI
provides functionality atop other run-times

- Multiple communication backends provide risk mitigation
- Type-safety added to Legion's task-based execution model
- Kokkos wrapped to support simple use cases for ranges of values or indices
- Overheads measured as negligible

➔ Application code can be oblivious to everything lying beneath the FleCSI layer



FleCSI provides a separation of concerns

Without FleCSI

- Physics code is cryptic to computer-science contributors and vice versa
- Dual expertise required for modification

Separating the two,

FleCSI lets physics codes…

- …be written by physicists
- …benefit from advanced computing infrastructure
- ...quickly & easily be ported to new architectures

FleCSI
code example

Task declaration

```
void red(mesh::accessor<ro> mesh,
         field<double>::accessor<rw,rw,ro> u,
         field<double>::accessor<ro,ro,ro> f) {

    forall(vertex, mesh.vertices<mesh::owned>(),"red") {
        u[vertex] = ...
    };

 }
```

Task execution

```
flecsi::execute<red>(mesh, u(mesh), f(mesh));
```

FleCSI
application developers' viewpoint

Physics developers need to:

- understand the parallel communication requirements of their algorithm
- declare their tasks' read and write permissions to FleCSI (details to follow)

Physics developers do *not* need to:

- explicitly request communication
- keep track of whether ghost data is valid or stale
- know exactly *when* tasks occur, including parallel communication
- know how colors are bound to hardware
- work hard to get new physics modules onto GPU
- change code to explore MPI, Legion, or HPX backends

FleCSI
implicitly schedules ghost copies

- Categories of data:
  - *owned = exlusive + shared*
  - *exclusive*
  - *shared*
  - *ghost*

- `FleCSI` has four types of permission
  - `ro` – read only
  - `wo` – write only
  - `rw` – read and write
  - `na` – no access

- Developer declares permissions for *exclusive*, shared and *ghost* data in each task, e.g. `<rw,rw,ro>`

- FleCSI deduces when ghost updates are required, and then schedules them

Challenges:

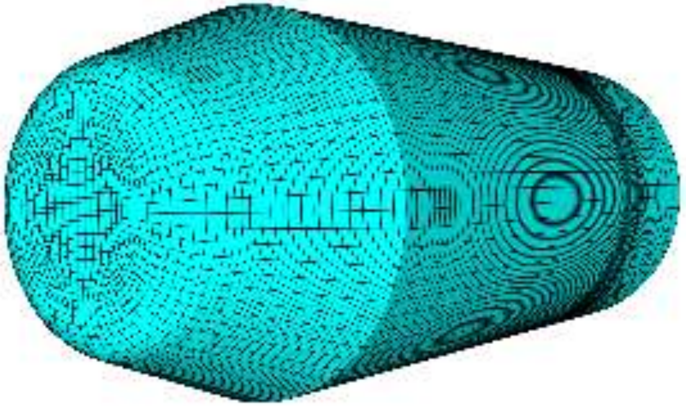
- How we're different
- History of progress
- Where are we today?

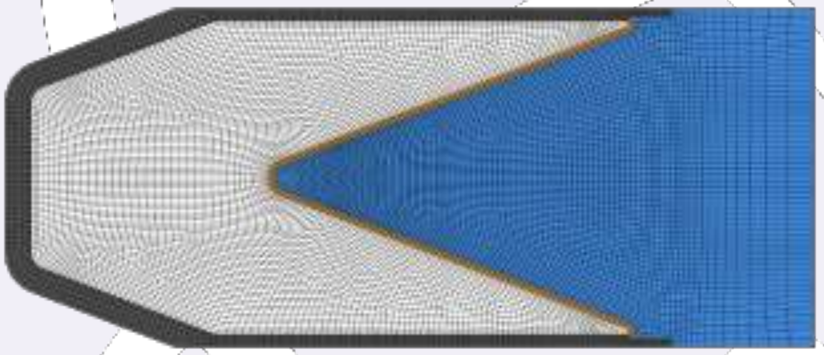
Challenges:

- How we're different (unstructured meshes)

Moya 3D shaped charge mesh (Unstructured)

- Steel (grey)
- High Explosive (white)
- Copper (orange)
- Air (blue)



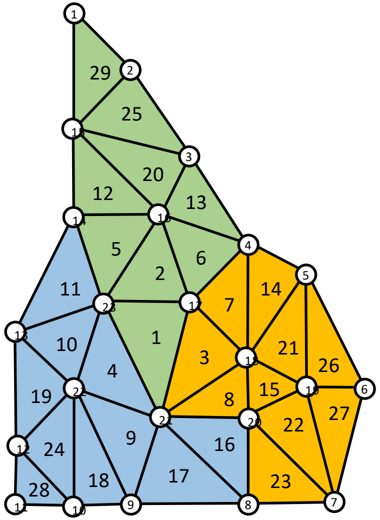


Left: Outer boundary of 3D shape charge mesh

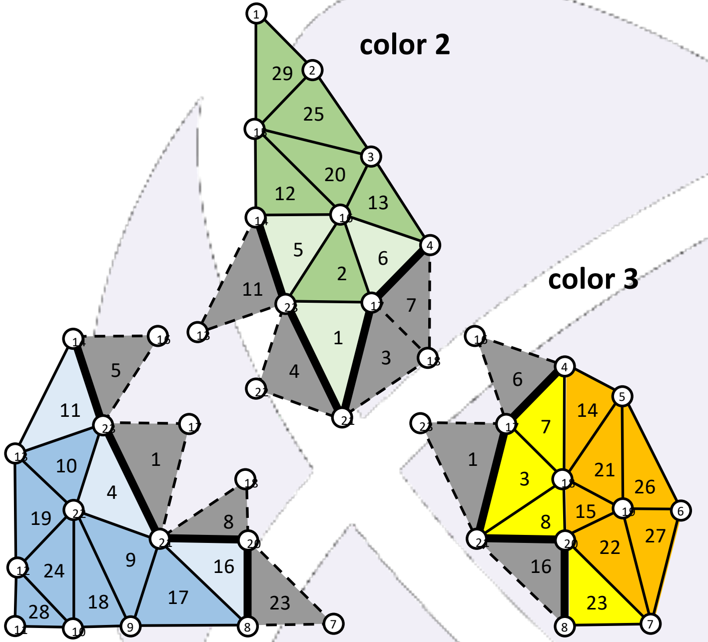
Right: 2D slice through the middle of the shape charge mesh

Unstructured meshes
lead to sparsity

Figure: colored-mesh

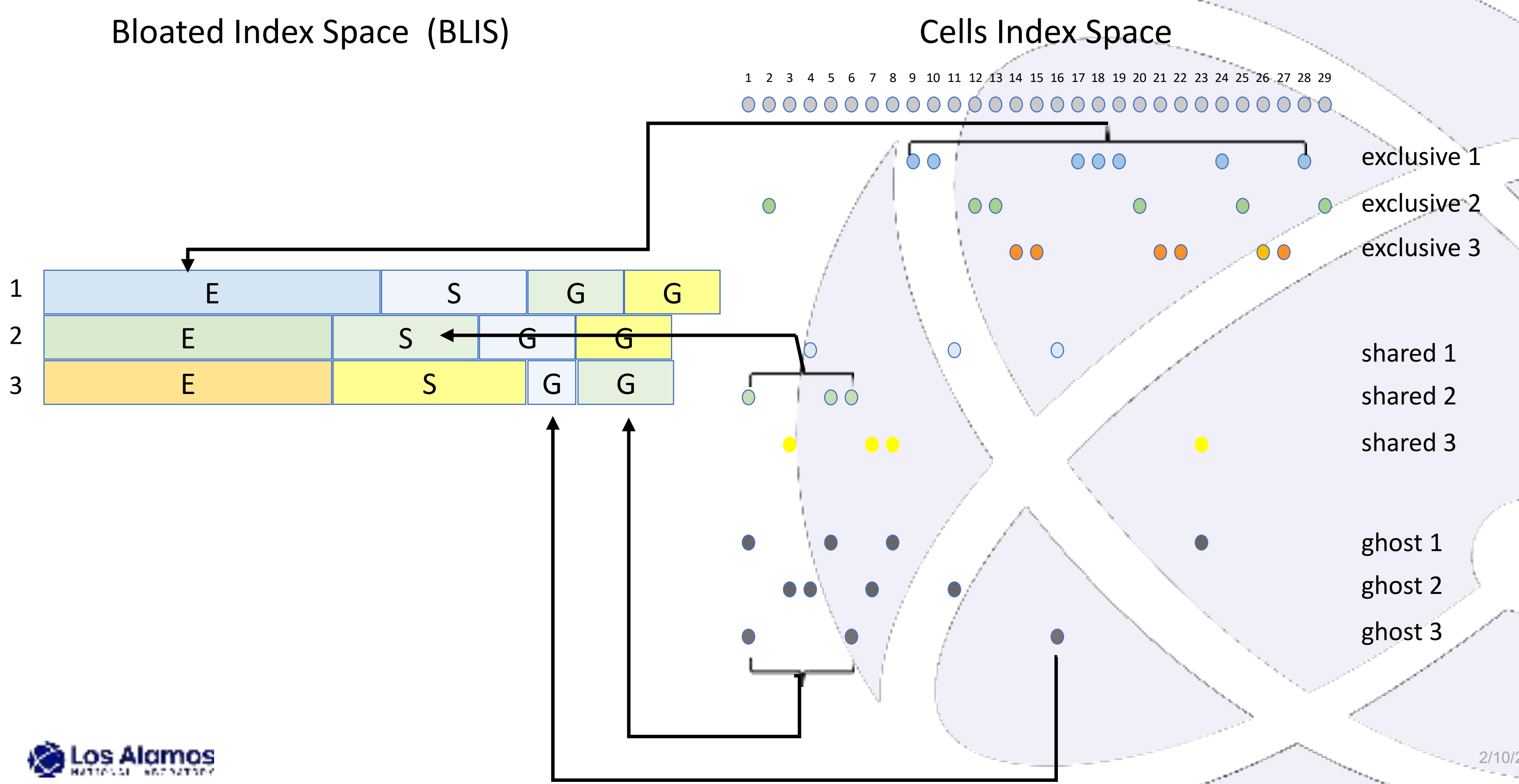


colored mesh

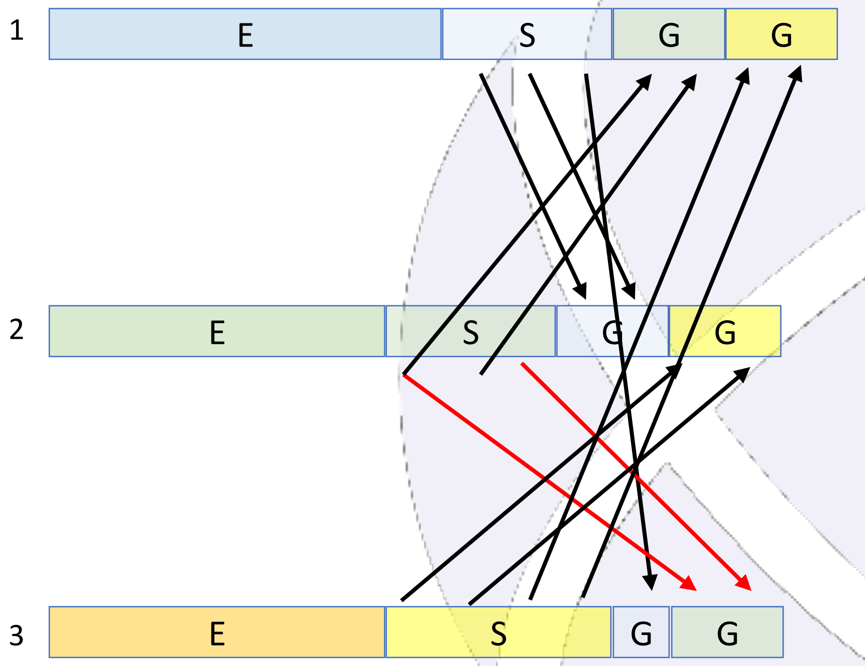


Exclusive, Shared, and Ghost

"Compact" the sparse data structures



Sparse ghost copies
1 in this simple example

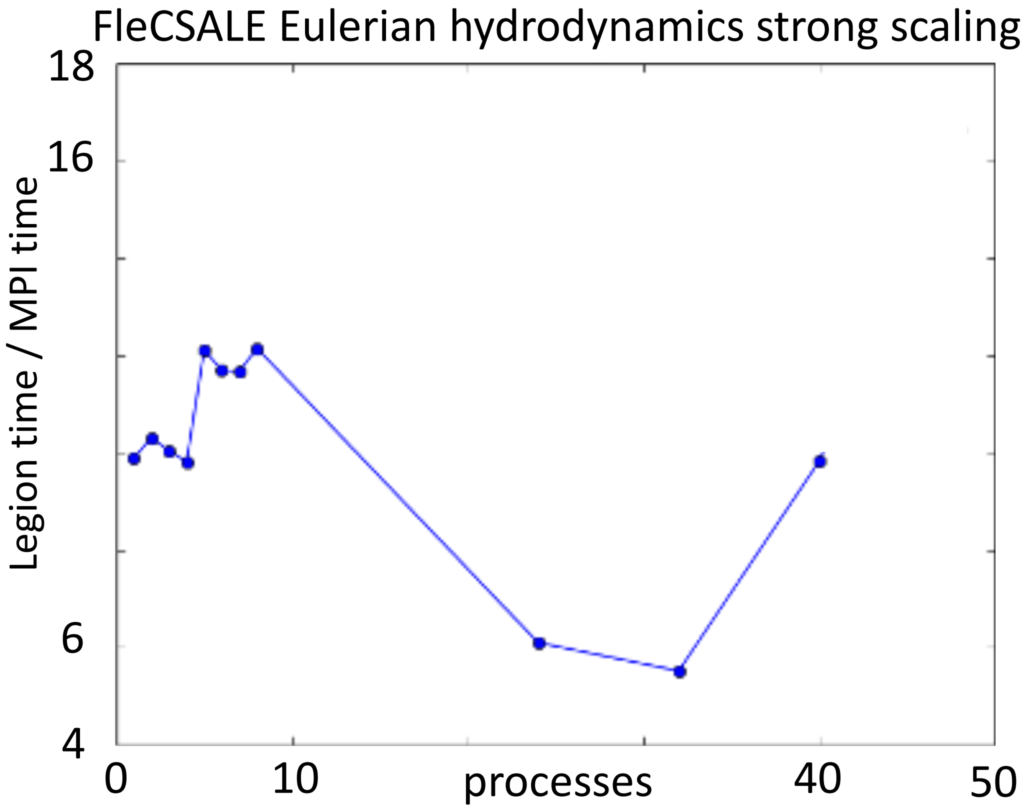


Challenges:

- History of progress

FleCSI 1.4: 1 color per process, no OpenMP

Legion is designed **to run 1 process / node**



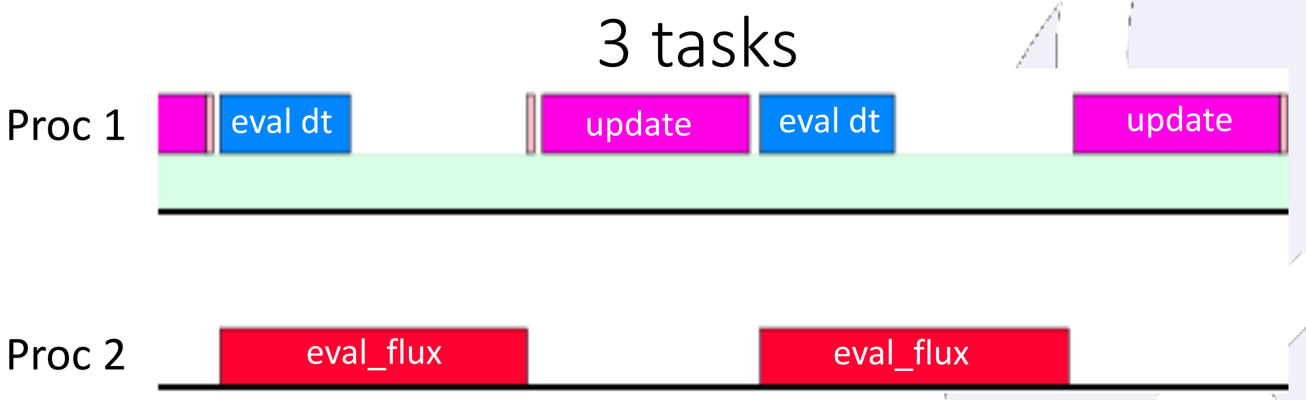
200X better than
2 years before

2D on 1M cells using
one CTS-1 node.
(Lower is better.)

FleCSALE provided both direct Eulerian and Arbitrary Lagrangian-Eulerian (ALE) hydrodynamics for gases

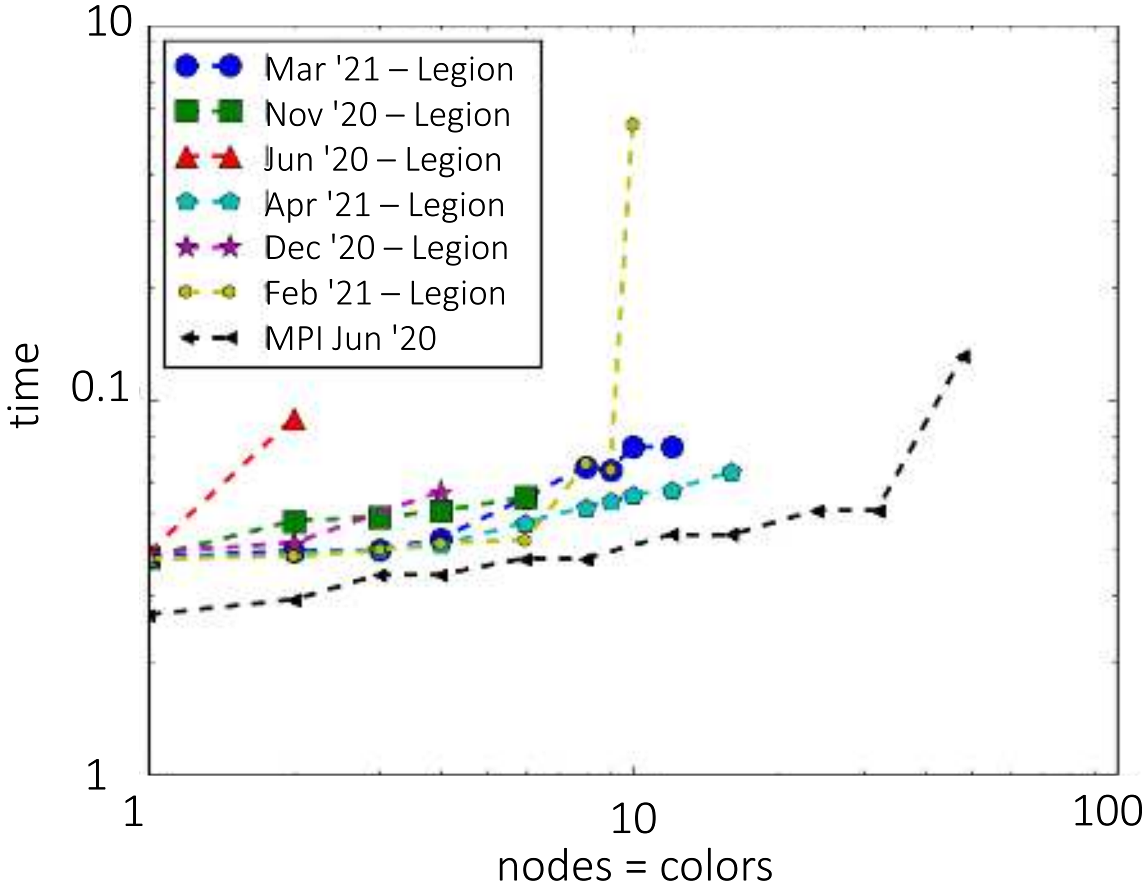
CY19 Q4

Little task parallelism in early apps



FleCSI 1.4

Long history of performance improvements



Simplified weak scaling

Time per FleCSALE hydro_3d time step versus number of Broadwell nodes

8 Mcell mesh

This study utilizes only 1 core per node for the computation.

Not a comparison: track progress

FleCSI 1.4

FleCSI 2 rewrite
~2020 - 2023

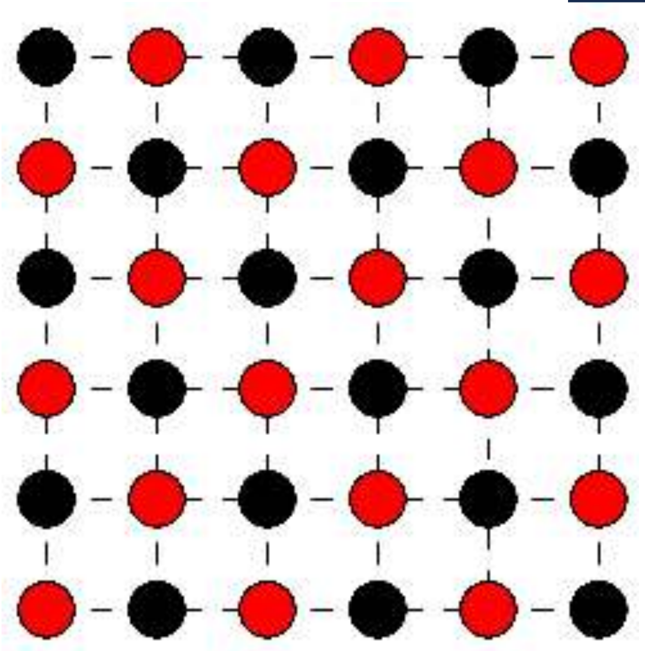
- New features
  - Multiple topology types (say, particles and a mesh)
  - Multiple instances of each (for, say, remapping)

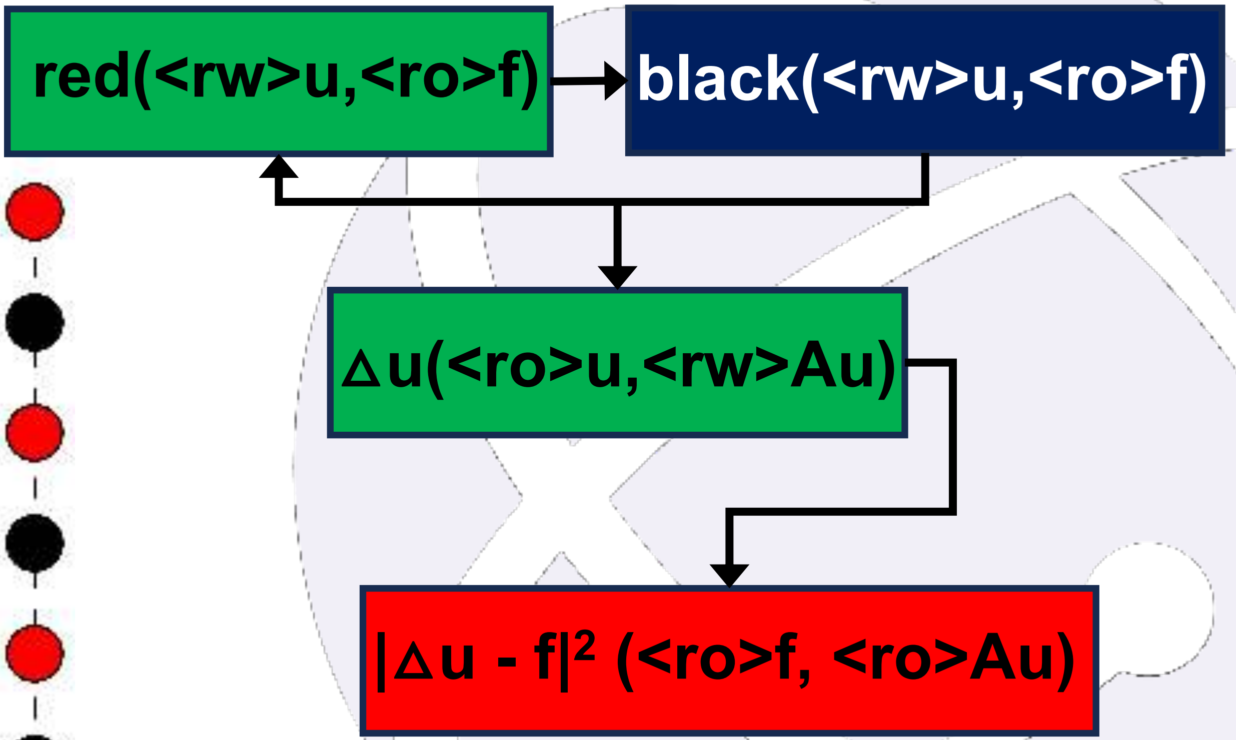
- New interface
  - Idiomatic modern C++ with templates and ranges
  - No need for registration of tasks, variables, and reductions
  - No string-based macros, which necessitated all-at-once updates for clients

Legion "overhead" measurement

- 2D Poisson with Red-Black Gauss Seidel

$\nabla^2 u = f$





FleCSI 2.0

On node optimization

not a strong scaling comparison

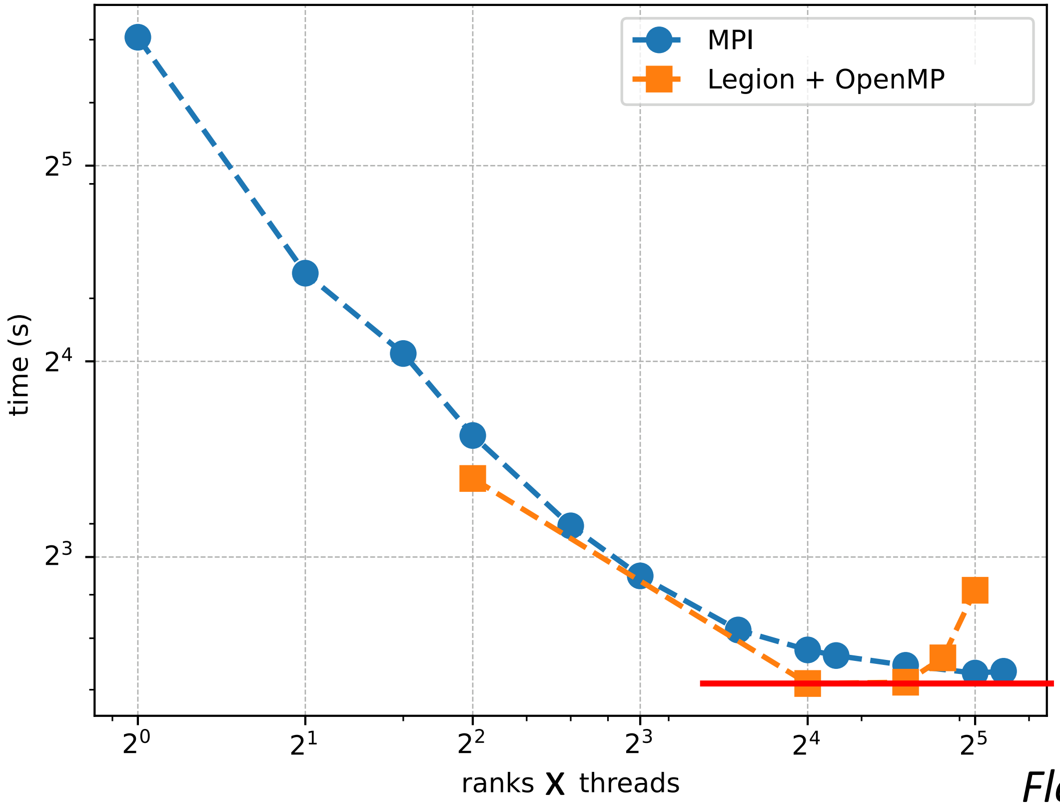
2D Poisson with Red-Black Gauss Seidel

$\nabla^2 u = f$

8192^2 cells

one Broadwell node (2 sockets)

4 Legion processes with 1 OpenMP processor each

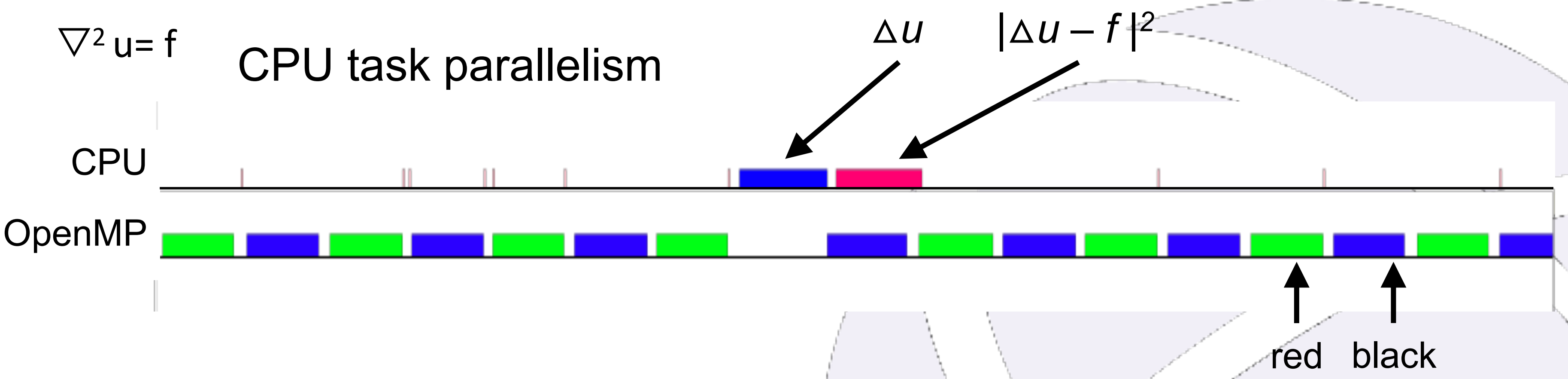


FleCSI 2.0

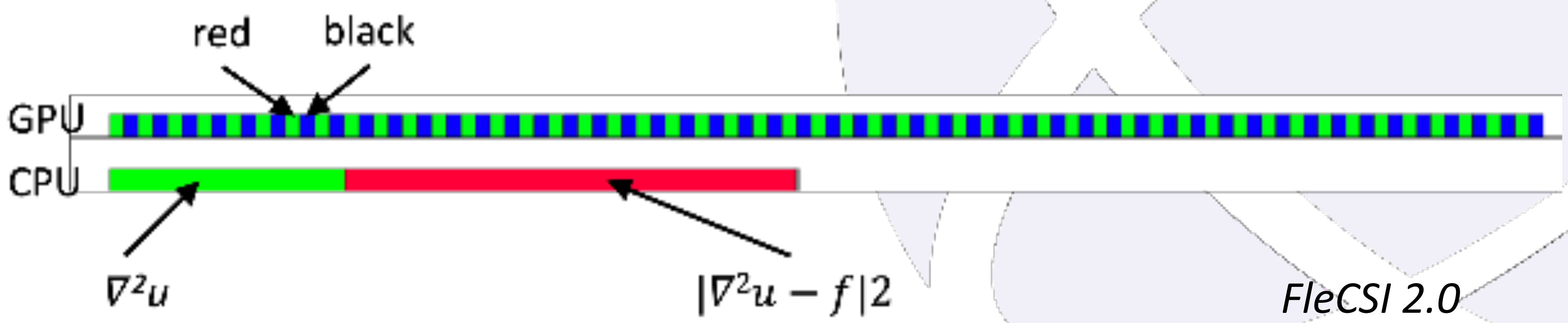
Small amount of tasking parallelism

$\nabla^2 u = f$

CPU task parallelism



CPU / GPU task parallelism



FleCSI 2.0

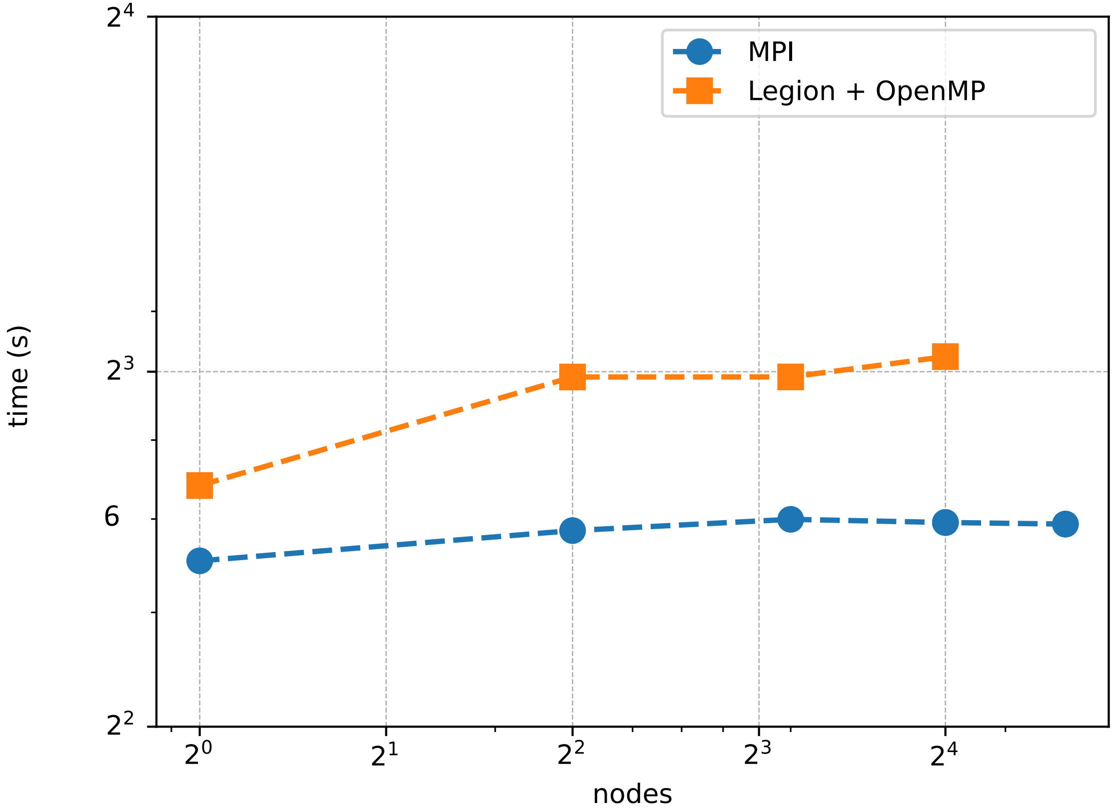
Weak Scaling

$\nabla^2 u = f$

8192^2 cells per Broadwell node

20-24 Legion threads per node

28-32 MPI ranks per node

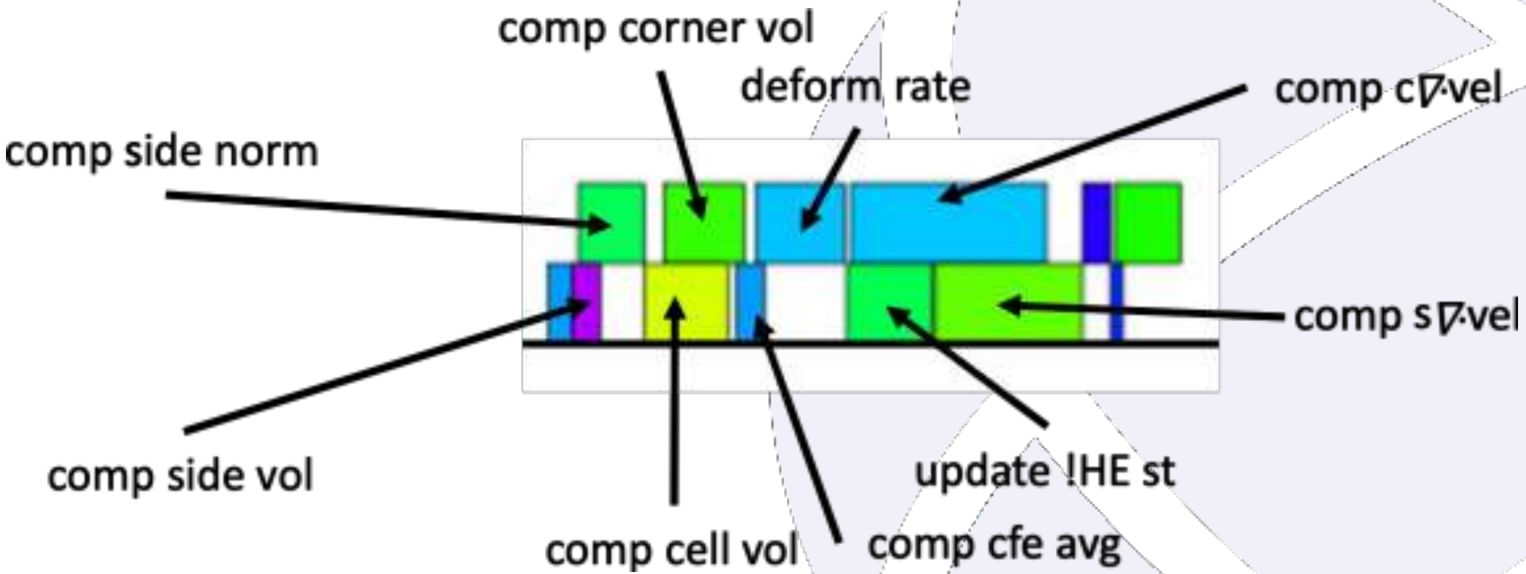


FleCSI 2.0

Challenges:

- Where are we today?

Moya has task parallelism



Los Alamos
NATIONAL LABORATORY

On node optimization

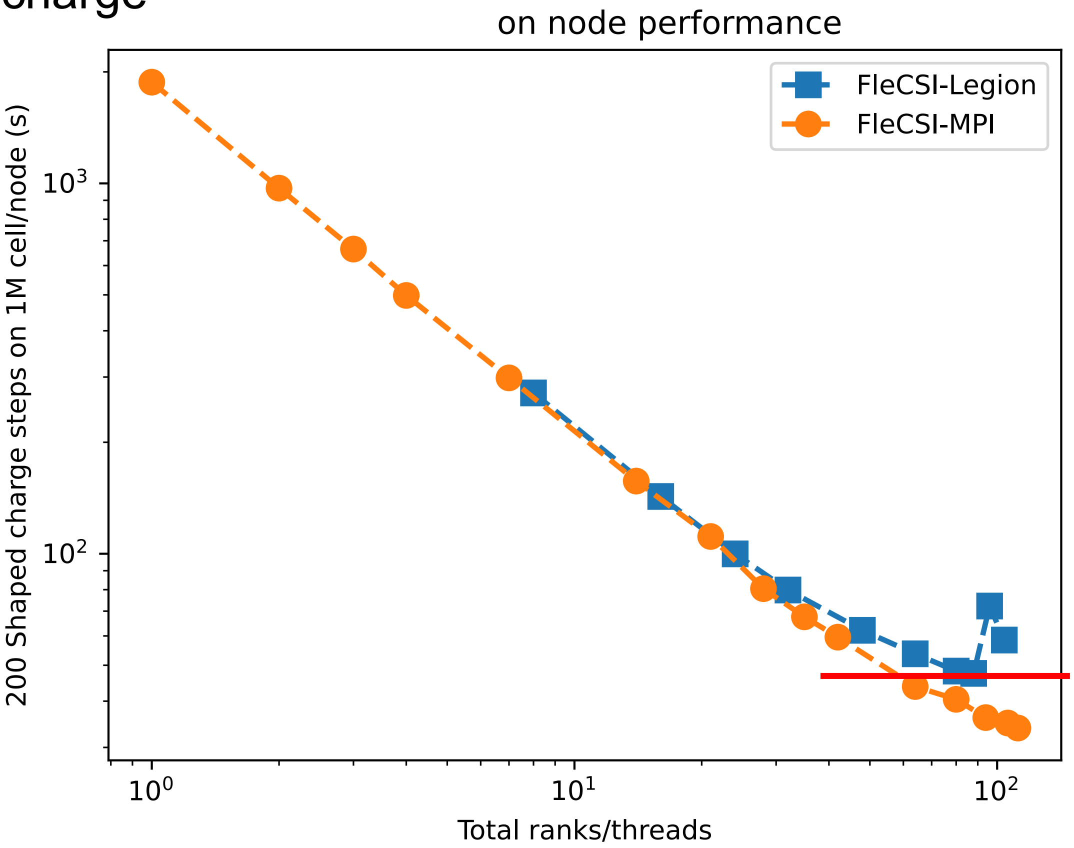
not a strong scaling comparison

Moya shaped charge

1M cells

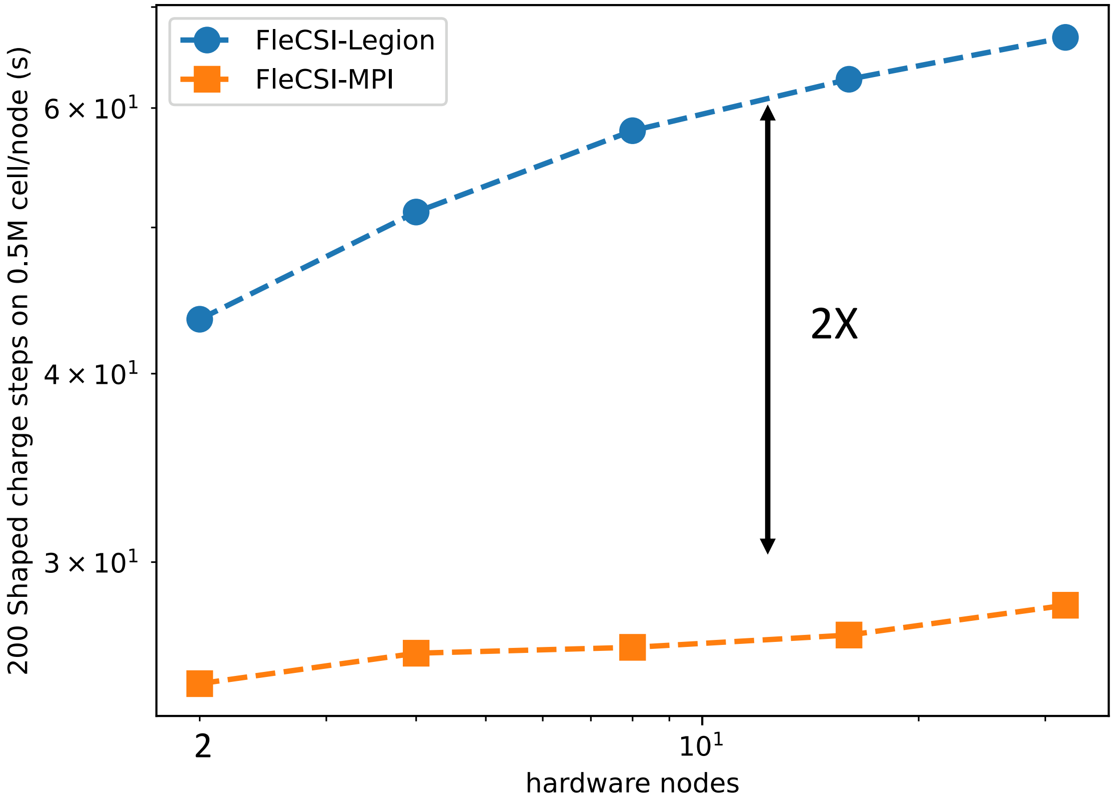
Saphire Rapids node (8 NUMA)

8 Legion processes with 1 OpenMP processor each



FleCSI 2.3

CPU weak scaling
Sapphire Rapids

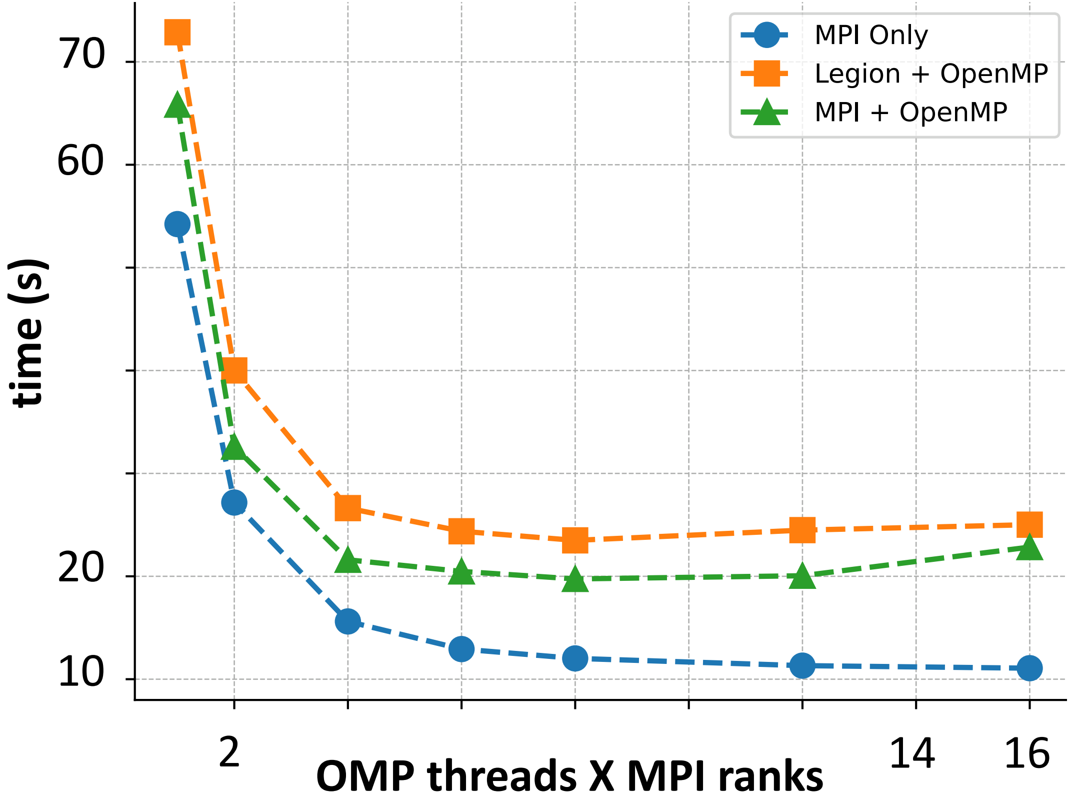


FleCSI 2.3

MPI + OpenMP
slower than MPI everywhere

$\nabla^2 u = f$

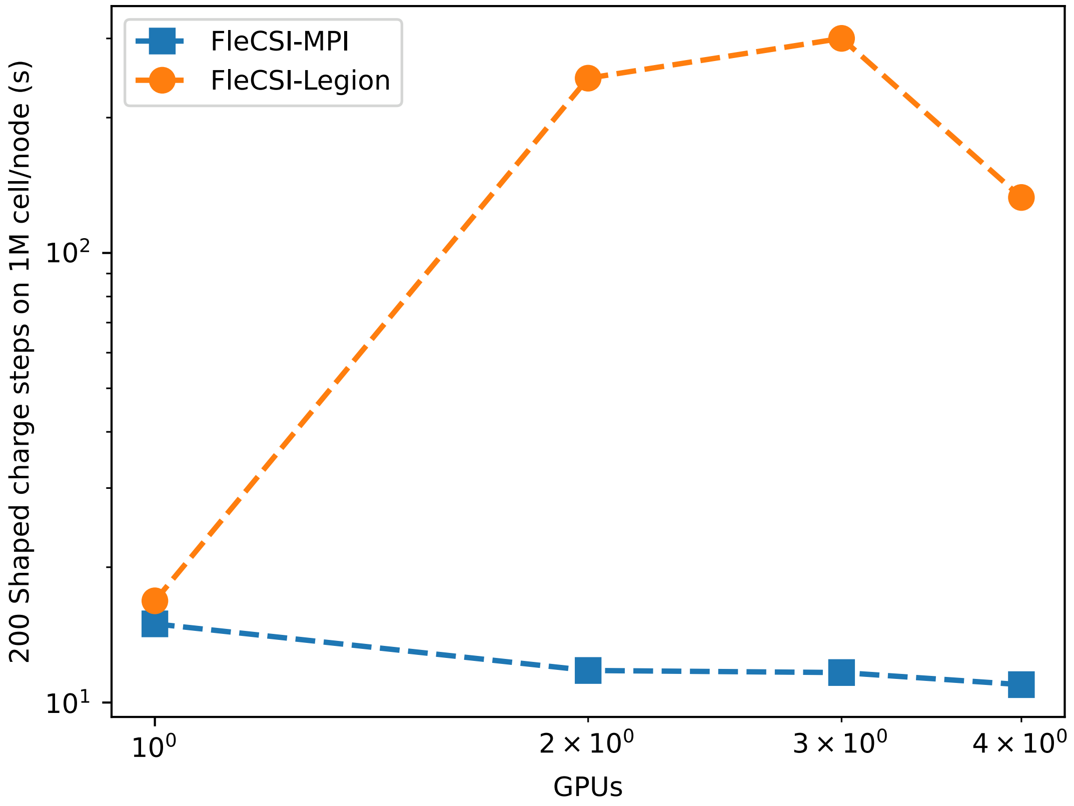
8192^2 cells per Broadwell node



FleCSI 2.0

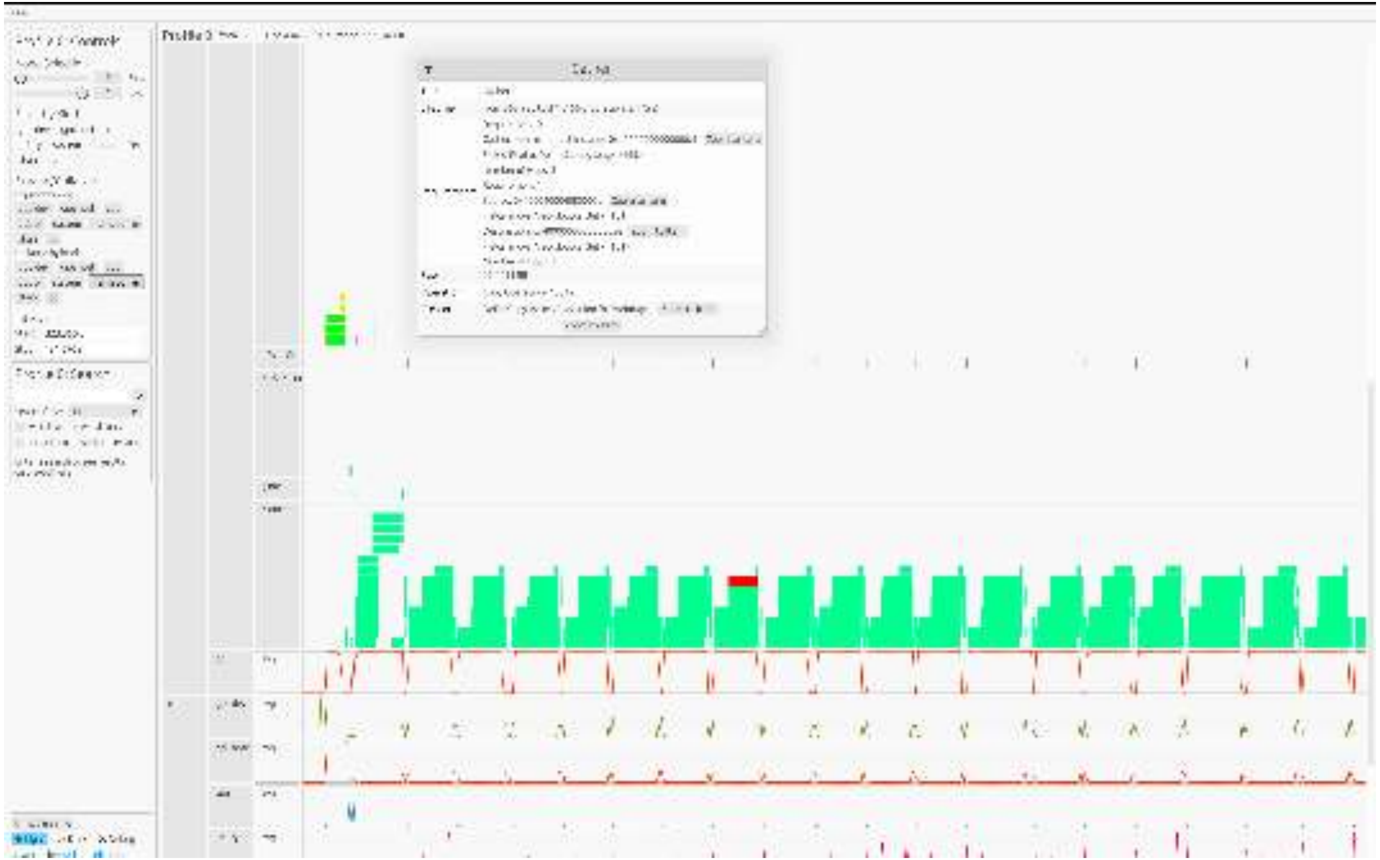
Los Alamos

GPU on-node strong scaling
H100's

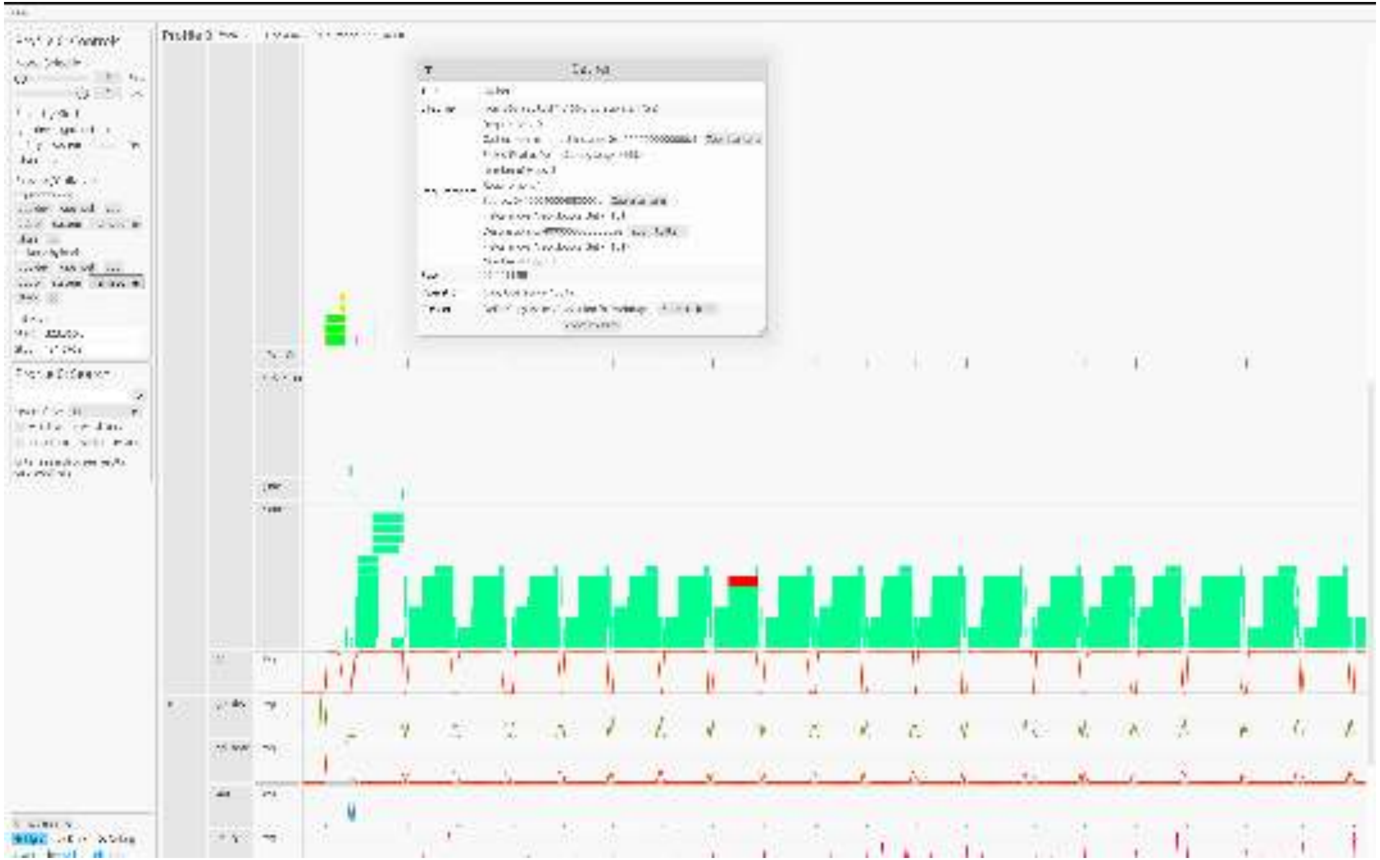


FleCSI 2.3

Profile of slow copy



Profile of slow copy



FleCSI with simple copies on A100s (single node)

GPUs	time (s)
1	22.1
2	43.6
4	76.8

Takeaways

- FleCSI mapper needs to support multiple ‘Legion OpenMP processors’
- What is happening with FleCSI mapper on multi-GPU runs?

- BLIS is maybe not the best data structure
  - Multiple structured meshes "stitched" together?

- Ristra is adopting a wait-and-see approach to Legion
- FleCSI will continue to work on Legion performance (as resources allow)

Questions?