MFC: ACCELERATING MULTI-PHASE FLOW SIMULATION

Spencer Bryngelson

Georgia Institute of Technology

OpenACC Summit

September 14, 2021

with J.Spratt, M. Rodriguez, B. Leback, S. Ethier

WHAT IS MFC

Multi-component flow code (MFC)

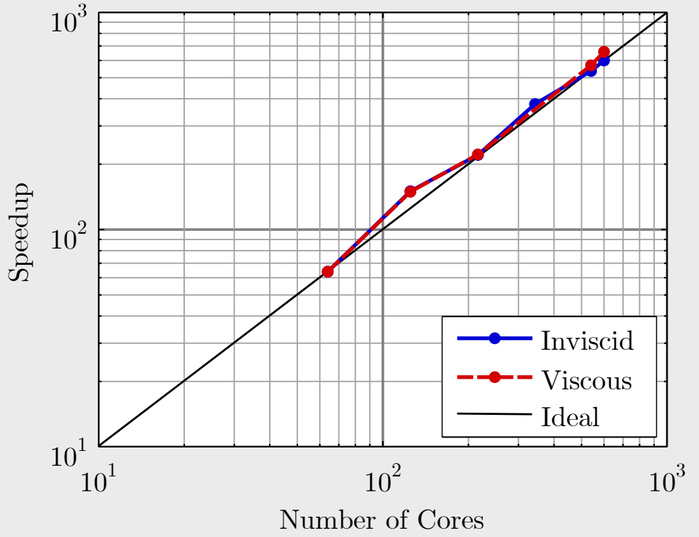
- Physics/numerics
  - Multi-phase compressible Navier–Stokes
  - Handle interfaces/shocks via “capturing”
  - Finite vol., 5th-order-accurate in space
  - Some fun features
    - Sub-grid multiphase dynamics
    - Elastic/viscoelastic models
- HPC
  - F90 w/ Python frontend
  - MPI domain decomp.
  - Parallel I/O, Lustre file system, etc.
  - Good scaling!

MFC

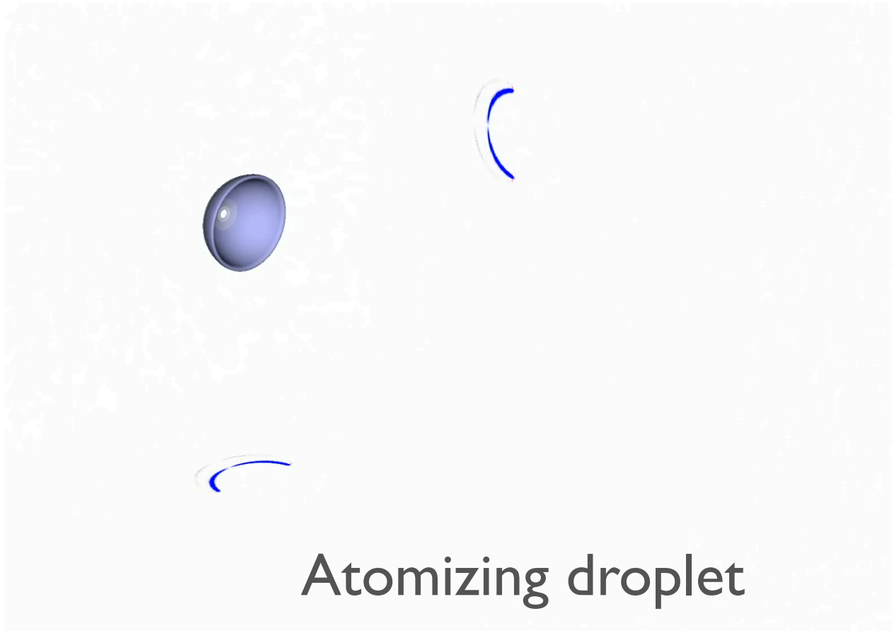
HIGH-FIDELITY MULTIPHASE FLOW SIMULATION

I’m open source!

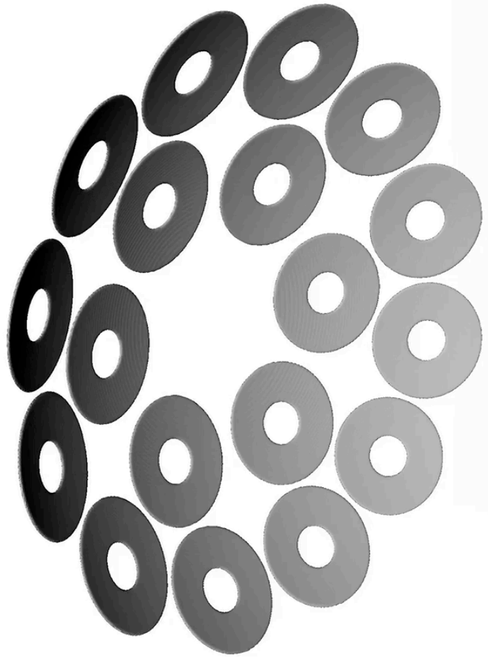
mfc-caltech.github.io



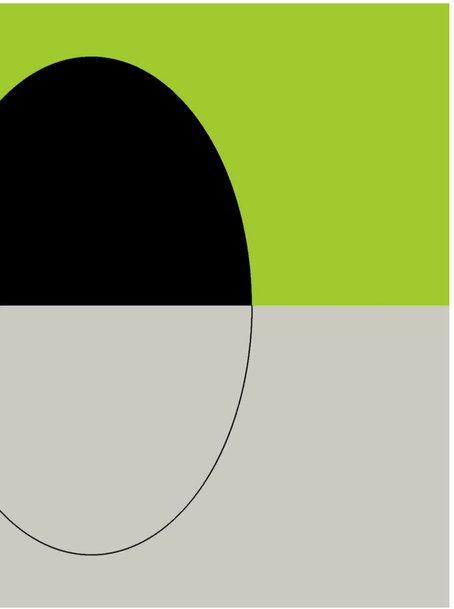
MFC IN CPU ACTION



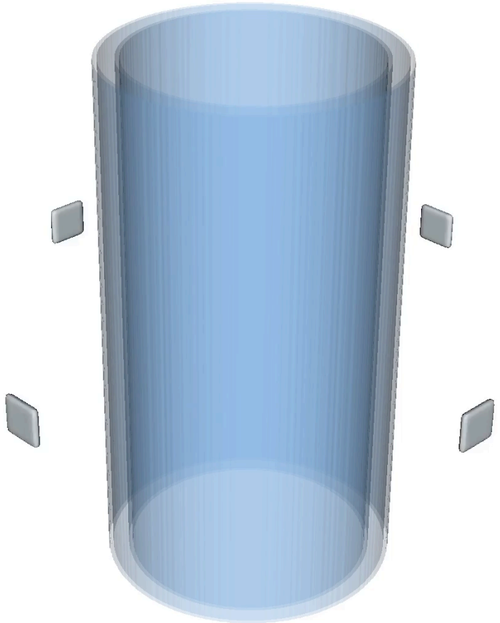
Atomizing droplet



Kidney stone ablation



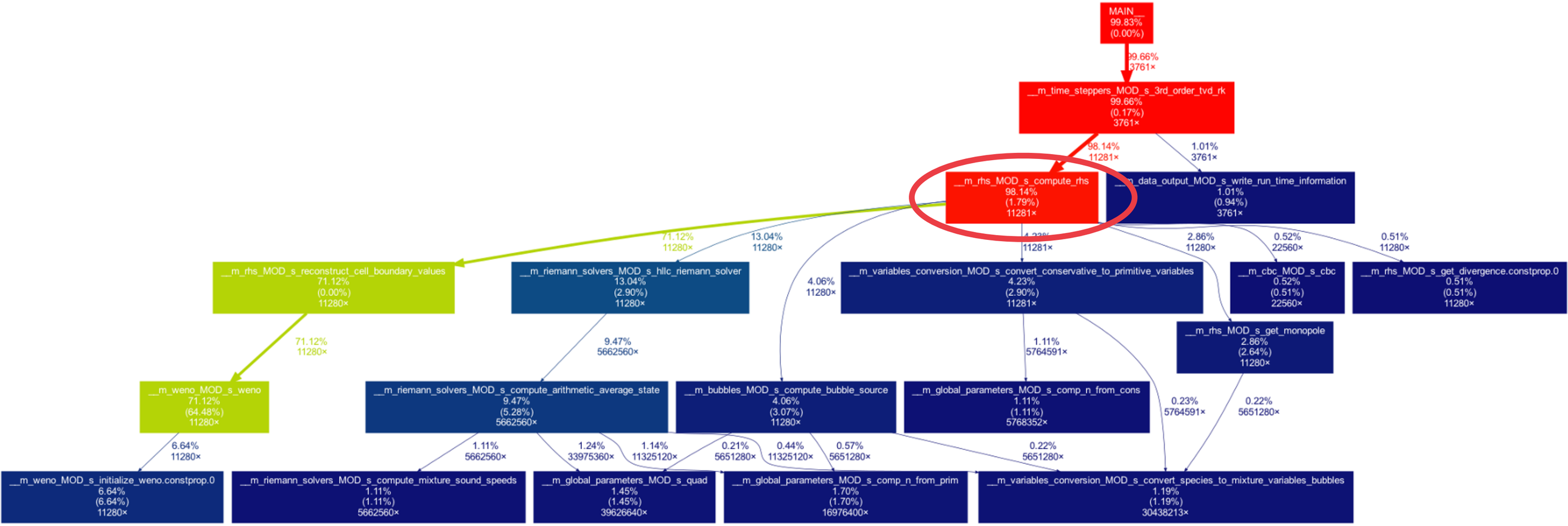
Wall-attached
bubble collapse



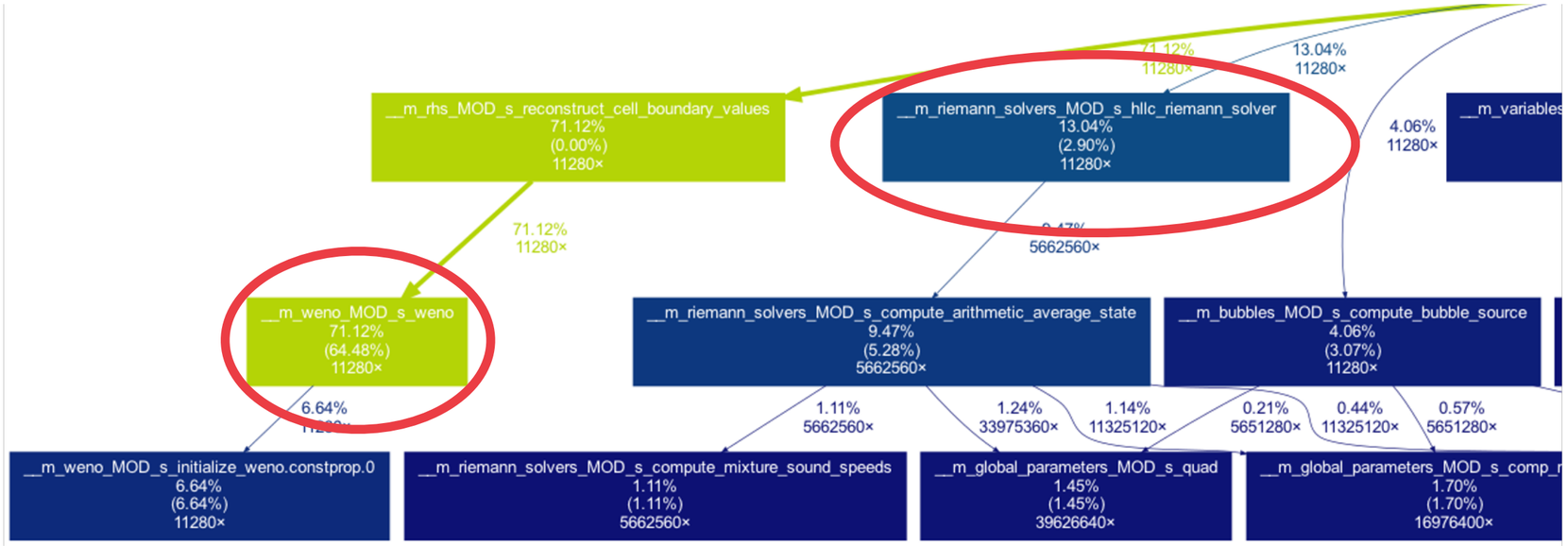
Model of feeding
humpback whales

Schmidmayer, Bryngelson et al. (2020), Bryngelson and Colonius (2020), Maeda et al. (2018), Meng (2016)

WHAT IS EXPENSIVE?



WHAT IS EXPENSIVE?



- 71% High-order-accurate WENO reconstruction
- 13% HLLC Riemann solve

These algorithms have

- High operation counts
- Low-medium memory reuse

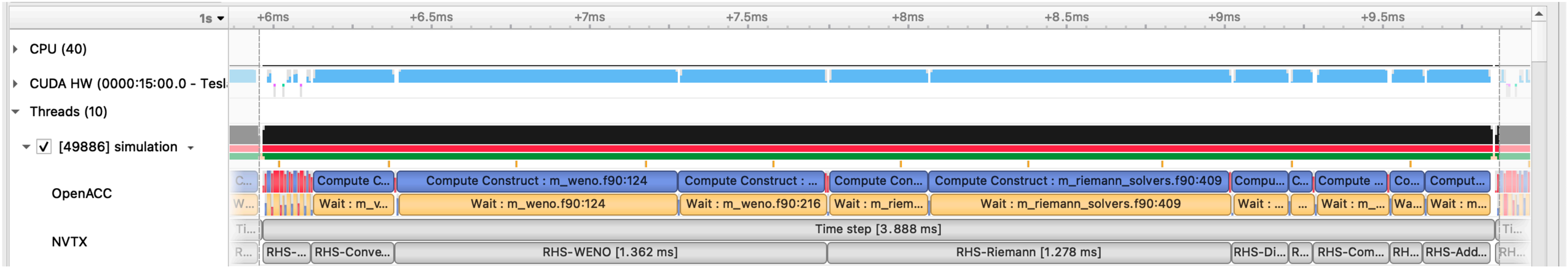
WHY OPENACC?

- Easy
  - Method complicated, don't know *a priori* if speed-up significant
  - Code long/complicated, directives helpful
- Portable
  - New supercomputers not exclusively Nvidia
  - CUDA Fortran dying
  - Only acceptable alternative is OpenCL
- Fast
  - Can be competitive with CUDA/OpenCL

KEY RESULTS

- In 3 days, team of 4 able to
  - Port core solver functionality to OpenACC kernels
    - Remove all disk-device transfers
  - Maintained MPI functionality (Nvidia MPS, CUDA-aware MPI)

- Speed-ups (on Oak Ridge Summit)
  - 300X speedup (*1x V100 vs 1 CPU core*)
  - 60X speedup node-wise (*8x V100 vs 40x CPU cores*)



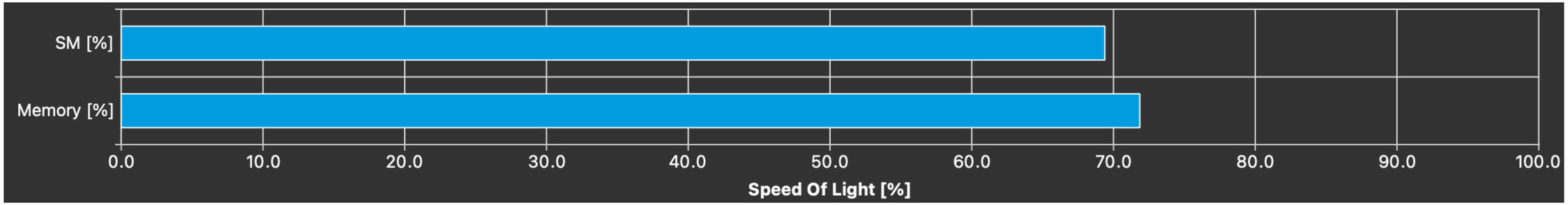
Comm.: 3%

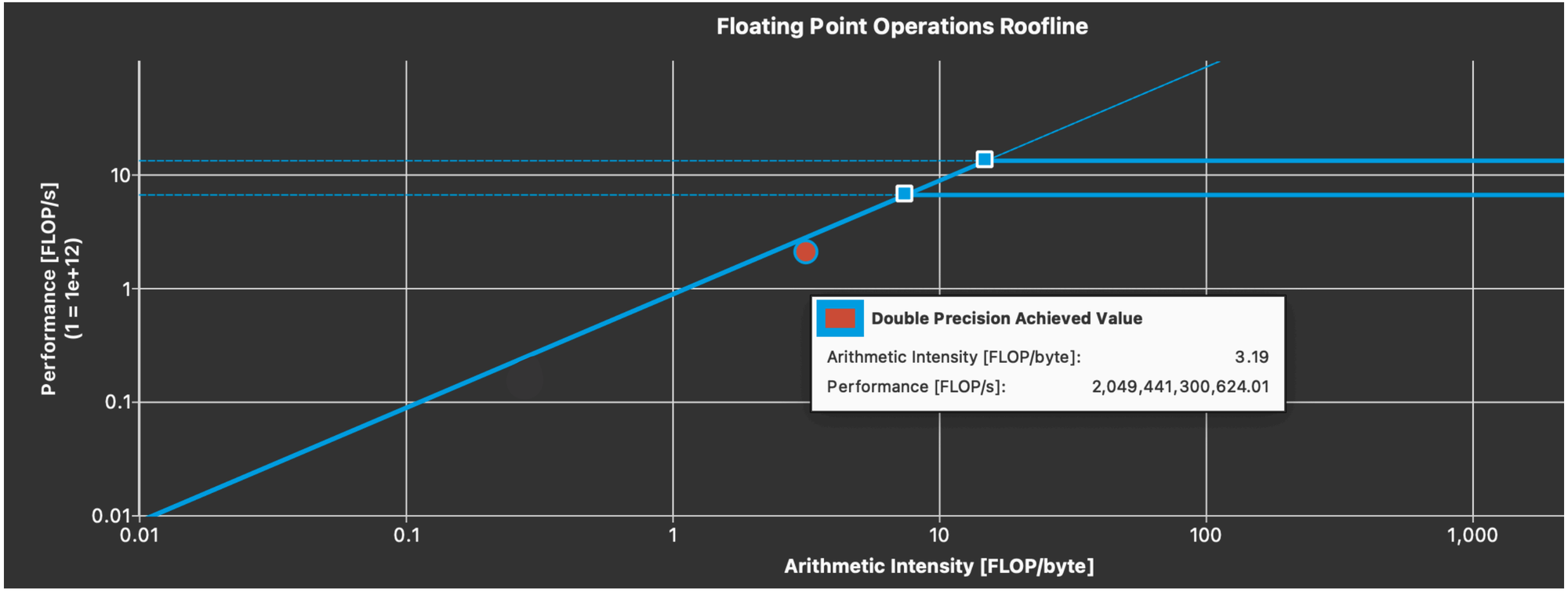
WENO: 33%

HLLC: 31%

Test on 10M grid cells

WENO KERNEL PROFILE





WHAT COULD BE BETTER?

Main

- Multi-node jobs
  - Better documentation of required MPI wrappers/env. variables
  - Lots of difficulty running multi-node MPI+OpenACC on XSEDE computers
- Profiling MPI jobs
- Fortran derived types + OpenACC data handling
  (tesla:deepcopy partial fix, Nvidia only?)

Auxiliary

- Compatibility with gcc compilers
- Support for non-Nvidia GPUs (MI-100?)
- Heuristic for when OpenACC is *not* appropriate

QUESTIONS?