Performance Enhancements for the Lattice-Boltzmann Solver in the LAVA Framework

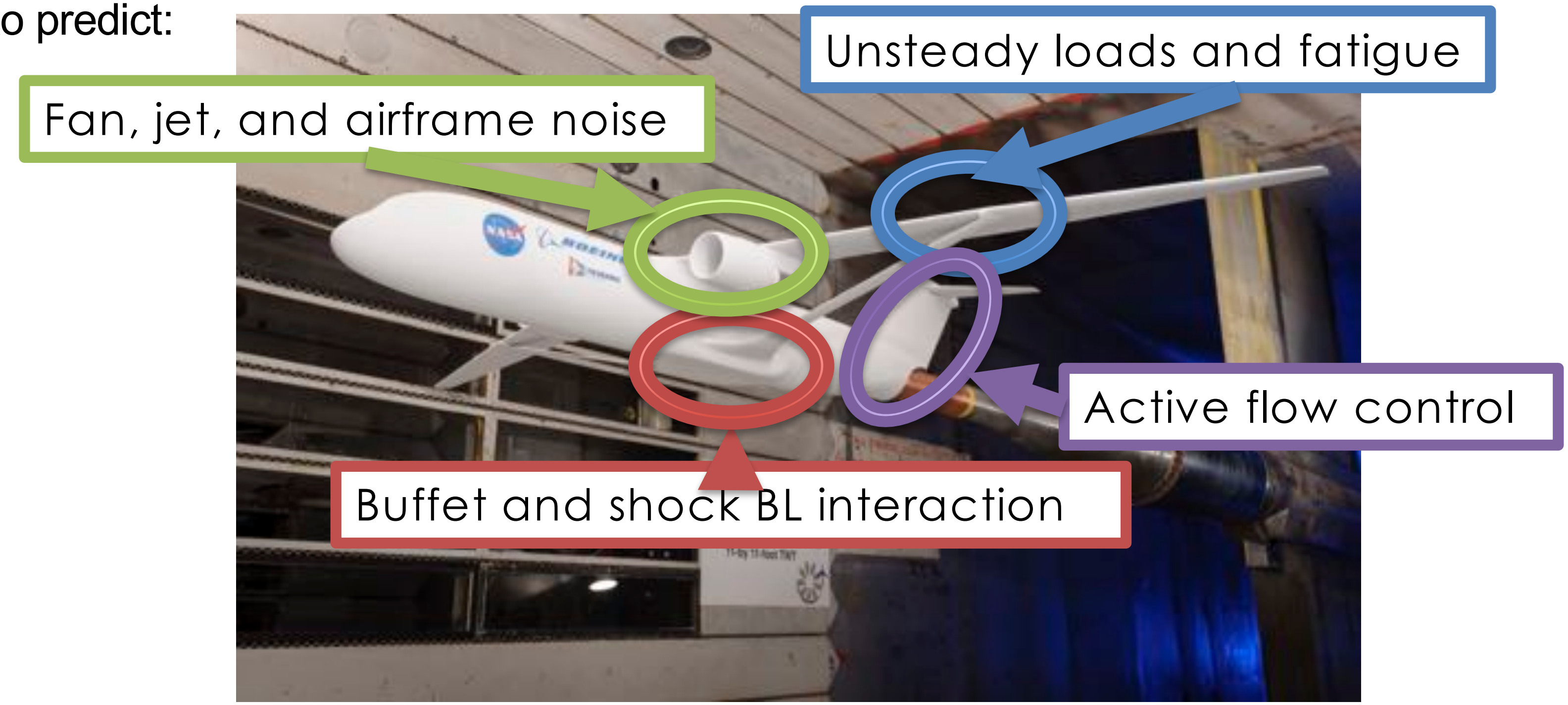
ICCFD10-2018-101

Michael Barad, Joseph Kocheemoolayil,
Gerrit Stich, and Cetin Kiris
Computational Aerosciences Branch
NASA Ames Research Center

ICCFD10 2018
July 9-13, Barcelona, Spain

Motivation

- ✓ **Increase predictive use of computational aerosciences capabilities for next generation aviation and space vehicle concepts.**
  - The next frontier is to use wall modeled and/or wall resolved large-eddy simulation (LES) to predict:



Unsteady loads and fatigue

Fan, jet, and airframe noise

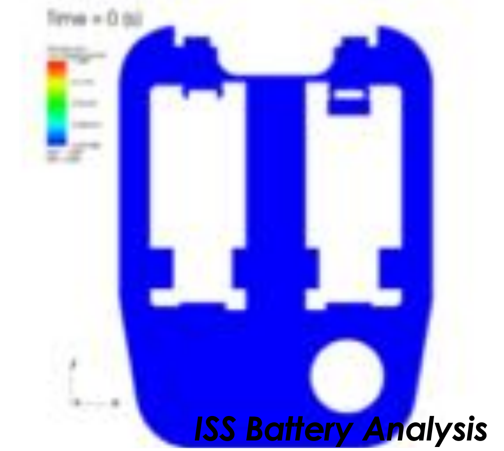
Active flow control

Buffet and shock BL interaction

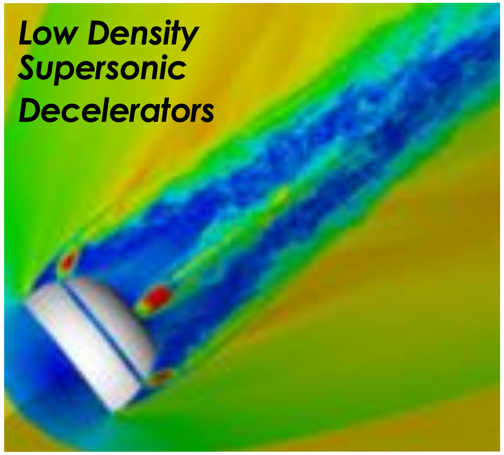
- ✓ **Need novel techniques for reducing the computational resources consumed by current high-fidelity CAA**
  - Need routine acoustic analysis of aircraft components at full-scale Reynolds number from first principles
  - **Need an order of magnitude or more reduction in wall time to solution!**

Many successful applications of High-Performance High-Fidelity Cartesian methods to NASA Mission critical applications

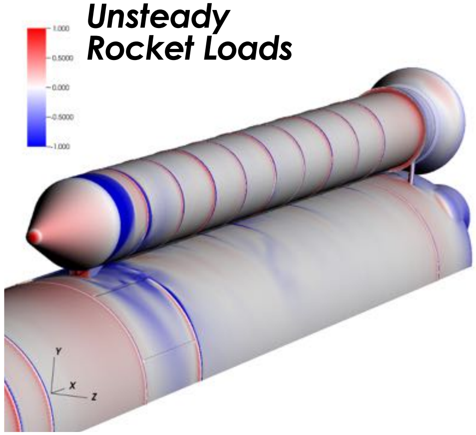
Launch Abort System Analysis for Orion



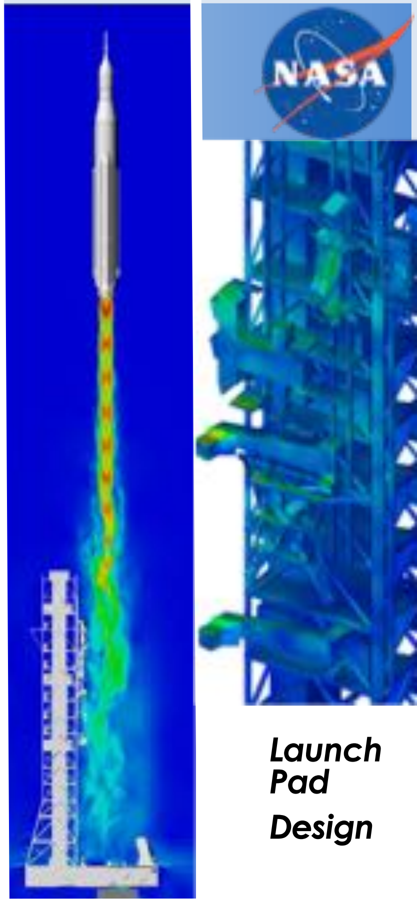
ISS Battery Analysis



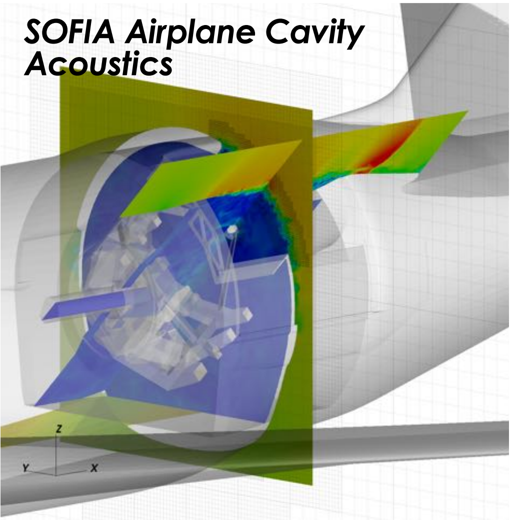
Low Density Supersonic Decelerators



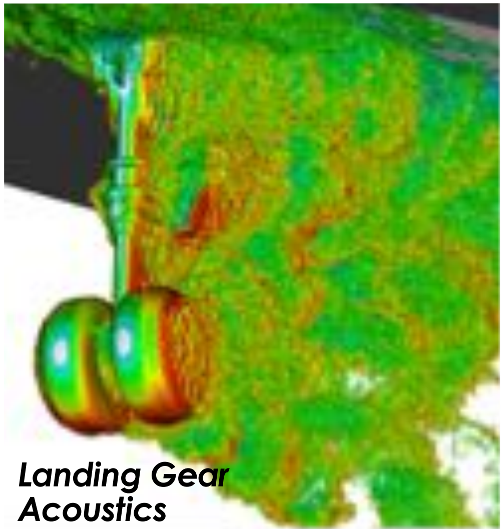
Unsteady Rocket Loads



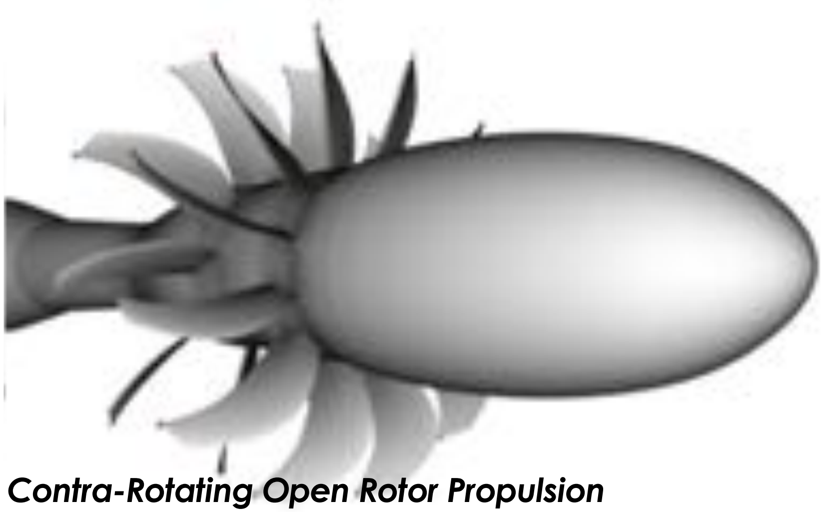
Launch Pad Design



SOFIA Airplane Cavity Acoustics



Landing Gear Acoustics

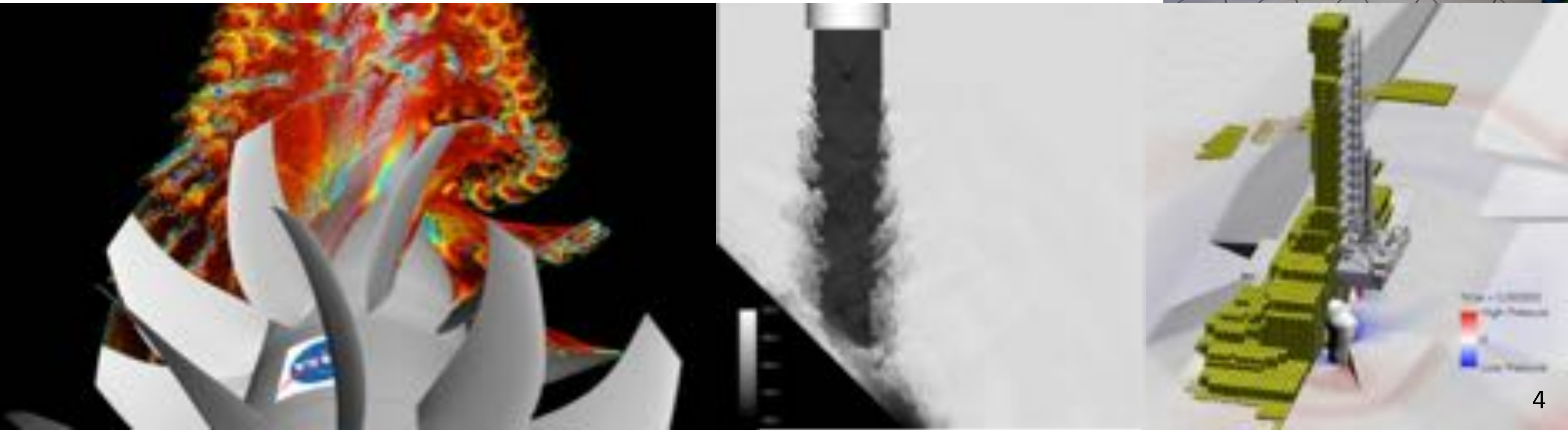


Contra-Rotating Open Rotor Propulsion

Challenges in Computational Aero-Acoustics

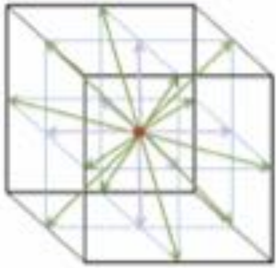
✓ Computational Requirements

- Space-time resolution requirements for acoustics problems are demanding.
- Resources used for recent Cartesian Navier-Stokes simulations:
  - Launch Environment: ~200 million cells, ~7 days (1000 cores)
  - Parachute: 200 million cells, 3 days (2000 cores)
  - Contra-Rotating Open Rotor: 360 million cells, 14 days (1400 cores)
  - Launch Abort System: 400 million cells, 28 days (2000 cores)
  - Landing Gear: 298 million cells, 20 days (3000 cores)



Lattice-Boltzmann Method (LBM)

$$\underbrace{f_i(\vec{x}+c\vec{e}_i\Delta t, t+\Delta t) - f_i(\vec{x},t)}_{\text{Streaming}} = \underbrace{\frac{1}{\tau}(f_i(\vec{x},t) - f_i^{eq}(\vec{x},t))}_{\text{Collision}}$$



✓ Physics:

- Governs space time evolution of Density Distribution Functions
- Equilibrium distribution functions are truncated Maxwell-Boltzmann distributions
- Relaxation time related to kinematic viscosity
- Pressure related to density through the isothermal ideal gas law
- Lattice Boltzmann Equations (LBE) recover the Navier-Stokes equations in the low Mach number limit

✓ Numerics:

- Extremely efficient 'collide at nodes and stream along links' discrete analog to the Boltzmann equation
- Particles bound to a regularly spaced lattice collide at nodes relaxing towards the local equilibrium (RHS)
- Post-collision distribution functions hop on to neighboring nodes along the lattice links (LHS) – Exact, dissipation-free advection from simple 'copy' operation
- Macroscopic quantities such as density and momentum are moments of the density distribution functions in the discrete velocity space

Lattice-Boltzmann Method (LBM)

✓ LBM Benefits:

- Ultra high performance: excellent data locality, vectorizable, scalable.
- Minimal numerical dissipation that is critical for computational aeroacoustics, and ideal for Large Eddy Simulations.
- Simulation of arbitrarily complex geometry with high performance structured adaptive mesh refinement is straight forward, bypassing manual and/or expensive meshing bottlenecks.

✓ NASA's LAVA-LBM

- **Progress to Date:**
    - LAVA Cartesian infrastructure has been re-factored into Navier-Stokes (NS) and LBM. Existing LAVA Cartesian data structures and algorithms are utilized.
    - Parallel Structured Adaptive Mesh Refinement (SAMR) meshing, robust collision models, second-order boundary conditions, all implemented.
    - Verification & validation: Taylor-Green vortex, flow past a cylinder, and nose landing gear.
    - A 12 to 15 times speedup compared to LAVA-Cart-NS was demonstrated for landing gear.
- **Current Efforts:**
    - Performance
        - Enhanced Accuracy at Coarse/Fine interface
        - Parallel Efficiency and Scaling
    - Moving Geometry
    - Wall Modeling
    - High Mach formulation

Focus on these for this paper

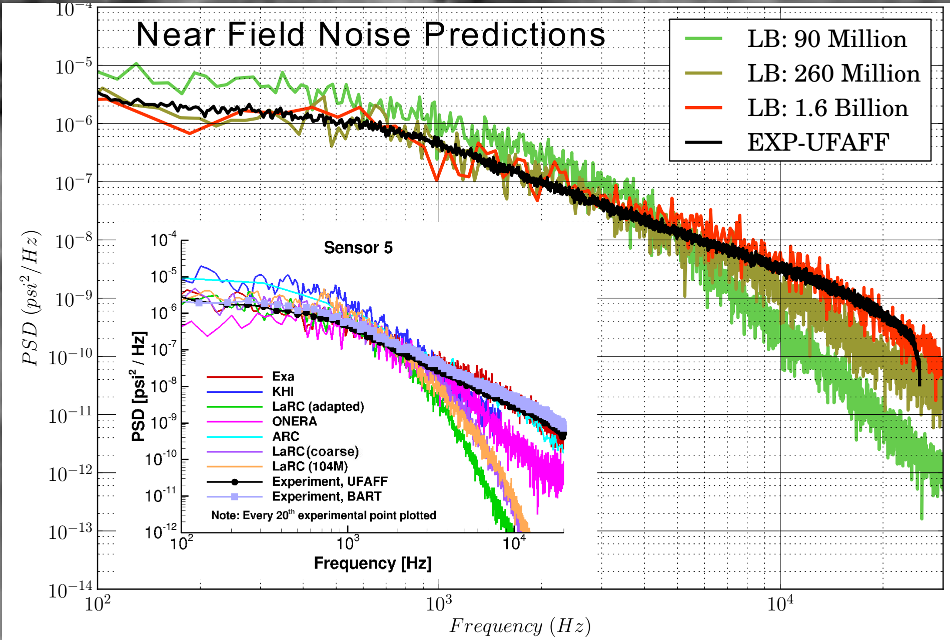
In testing phase

Initial stages of development

Recent LAVA-LBM Success for Landing Gear:

LBM @ 1.6 billion – Velocity Magnitude at Centerline

“Lattice Boltzmann and Navier-Stokes Cartesian CFD Approaches for Airframe Noise Predictions”, Barad, Kocheemoolayil, Kiris, AIAA 2017-4404

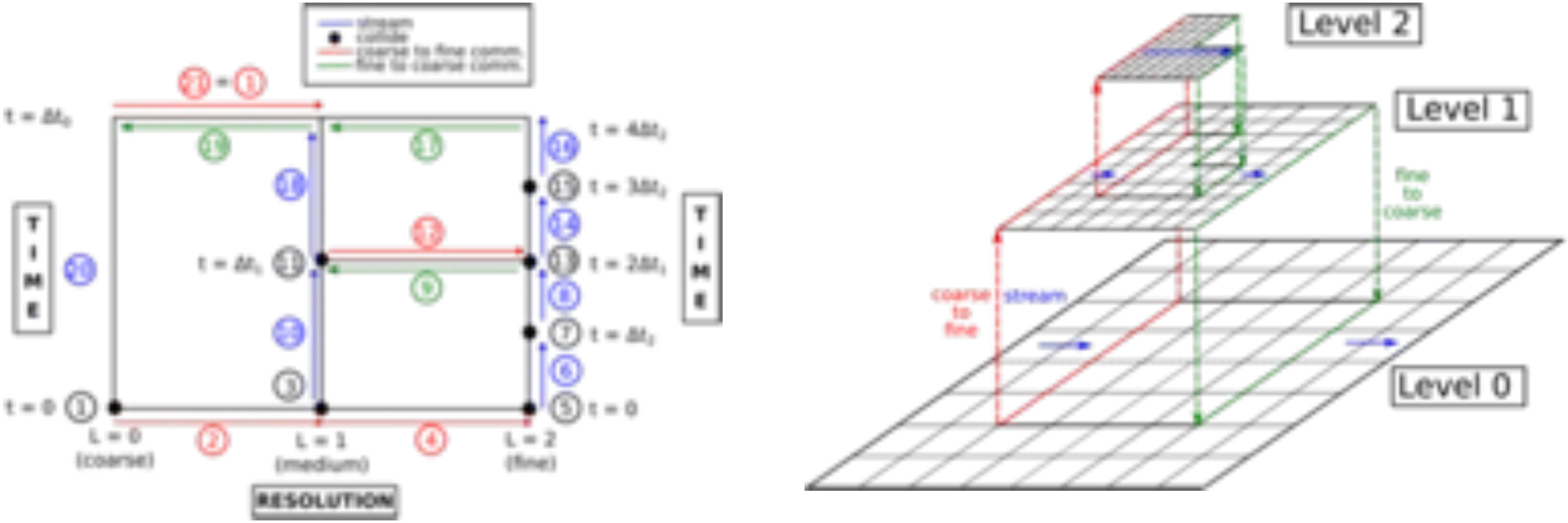


Lessons from LAVA LBM Landing Gear Simulations

- Previously demonstrated the LBM approach on the AIAA BANC III Workshop Landing Gear problem IV.
  - Computed results compare well with the experimental data
  - 12-15 times speed-up was observed between LBM and NS calculations.

- After completing the LG study, we knew that the code can be even faster!
  - Node usage not optimal with pure MPI programming model → go to **hybrid MPI/OpenMP**
  - Not enough parallelism for modern hardware → add concurrency with **tiling**
  - Moving geometry applications introduce many complexities:
    - Load balancing, points to **bigger boxes, fewer MPI ranks per node, and dynamic thread scheduling within boxes**
    - Geometry kernels are expensive, CPU vendor supplied ray-tracing libraries work best with **hybrid MPI/OpenMP**
  - Exciting **new hardware is coming to HPC**…codes need to be:
    - Ready for extremely **high concurrency**,
    - Using **memory bandwidth** efficiently

Conservative Coarse/Fine Interface



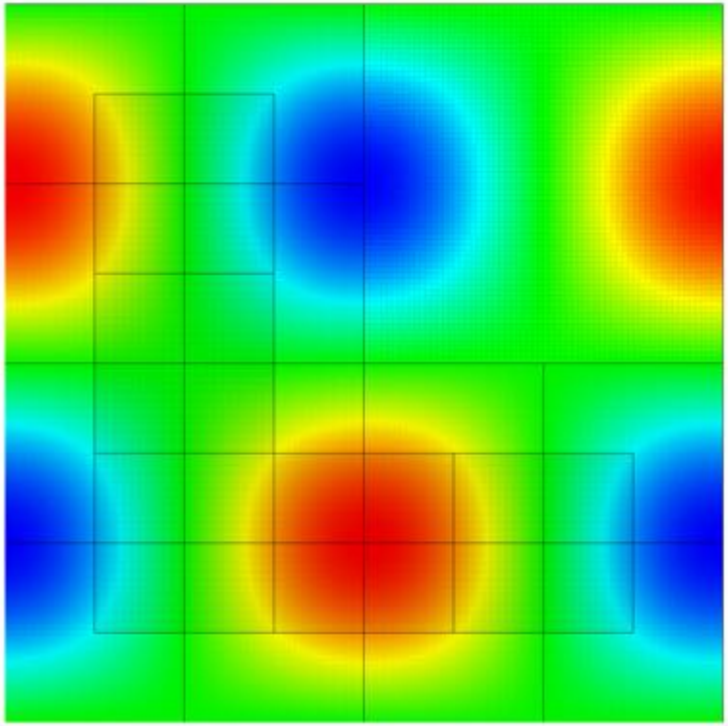
Sketch of conservative recursive sub-cycling algorithm.

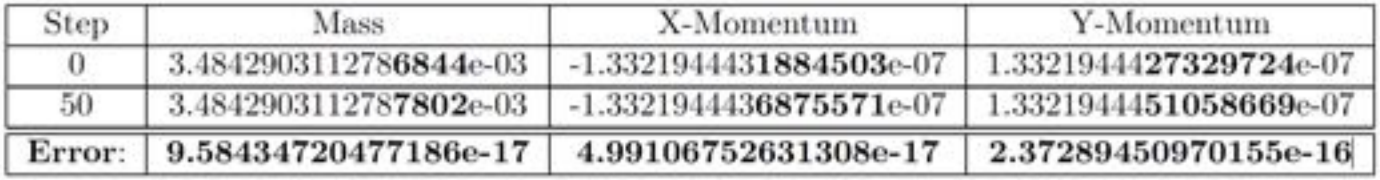
- Block structured AMR showing 3 levels of refinement by factor 2.
- Arrows indicate direction of information propagation:
  - streaming (blue),
  - coarse-to-fine communication (red),
  - fine-to-coarse communication (green).

Refs: Schornbaum and Rude 2016, Rohde et al 2006, Chen et al 2006.

Conservative Coarse/Fine Interface

2D Conservation Test:

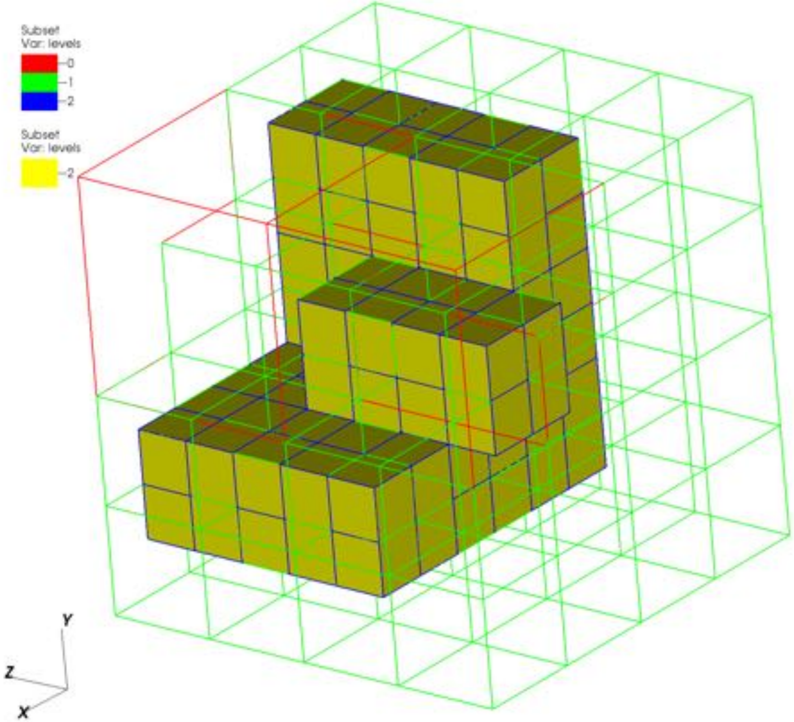


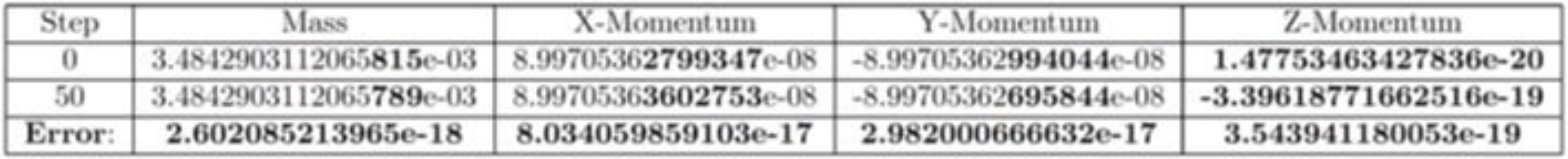


Step	Mass	X-Momentum	Y-Momentum
0	3.48429031127**86844**e-03	-1.33219444**31884503**e-07	1.33219444**27329724**e-07
50	3.48429031127**87802**e-03	-1.33219444**36875571**e-07	1.33219444**51058669**e-07
Error:	**9.58434720477186e-17**	**4.99106752631308e-17**	**2.37289450970155e-16**

Conservative Coarse/Fine Interface

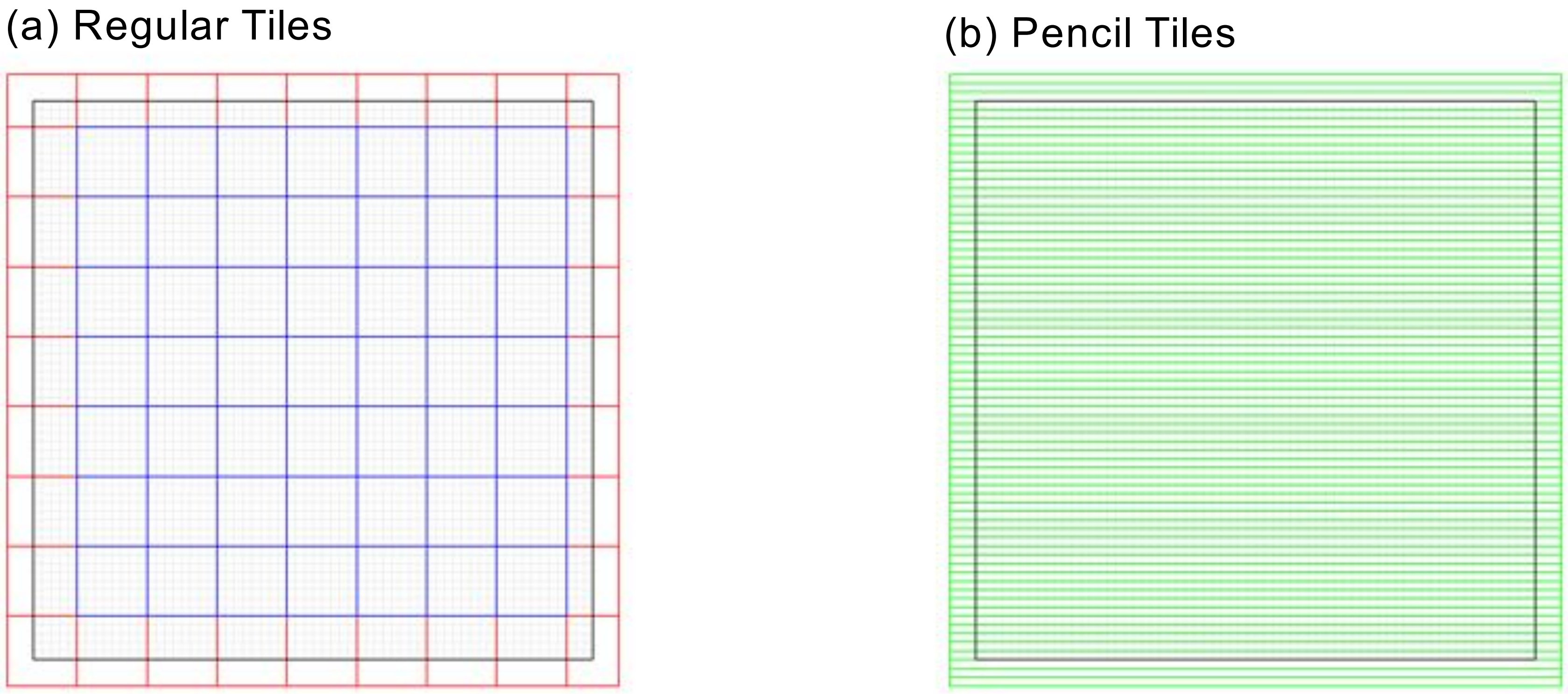
3D Conservation Test:





Step	Mass	X-Momentum	Y-Momentum	Z-Momentum
0	3.4842903112065**815**e-03	8.9970536**2799347**e-08	-8.9970536**2994044**e-08	**1.47753463427836e-20**
50	3.4842903112065**789**e-03	8.9970536**3602753**e-08	-8.99705362**695844**e-08	**-3.39618771662516e-19**
Error:	**2.602085213965e-18**	**8.034059859103e-17**	**2.982000666632e-17**	**3.543941180053e-19**

More Parallelism: Tiling


(a) Regular Tiles

(b) Pencil Tiles

Different tile types for a single box:

(a) regular tiles (8^D), including inner (blue) and outer (red); and

(b) pencil tiles (green) for contiguous memory accesses

The box shown has 64^D cells, plus 3 ghost layers. 3D tiles are conceptually similar.

More Parallelism: Tiling + OpenMP

Adding another level of parallelism has many benefits:

- **Loop collapse:** OpenMP over boxes on a proc & tiles in each box

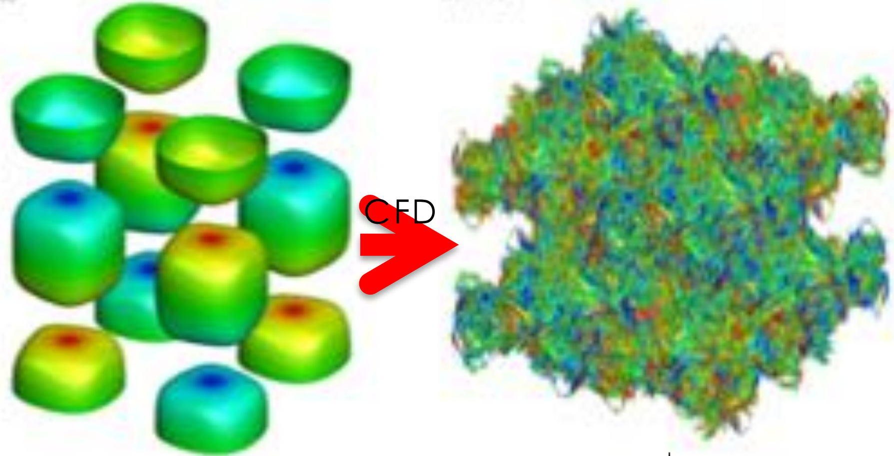
```
#pragma omp for schedule(dynamic) collapse(2)
for (int ibox = 0; ibox < nbox; ++ibox)
  {
    for (int itile = 0;itile < ntile; ++itile)
      {
        work(ibox,itile);
      }
  }
```

- **Improved load balancing** for irregularities:
  - Complex geometry
  - AMR
- **Bigger boxes** are possible which improves surface/volume ratios and reduces MPI expense
- **Asynchronous communication** is enabled:
  - Outer tiles are computed first, then non-blocking MPI sends
  - Inner tiles then computed
  - Finish MPI comms

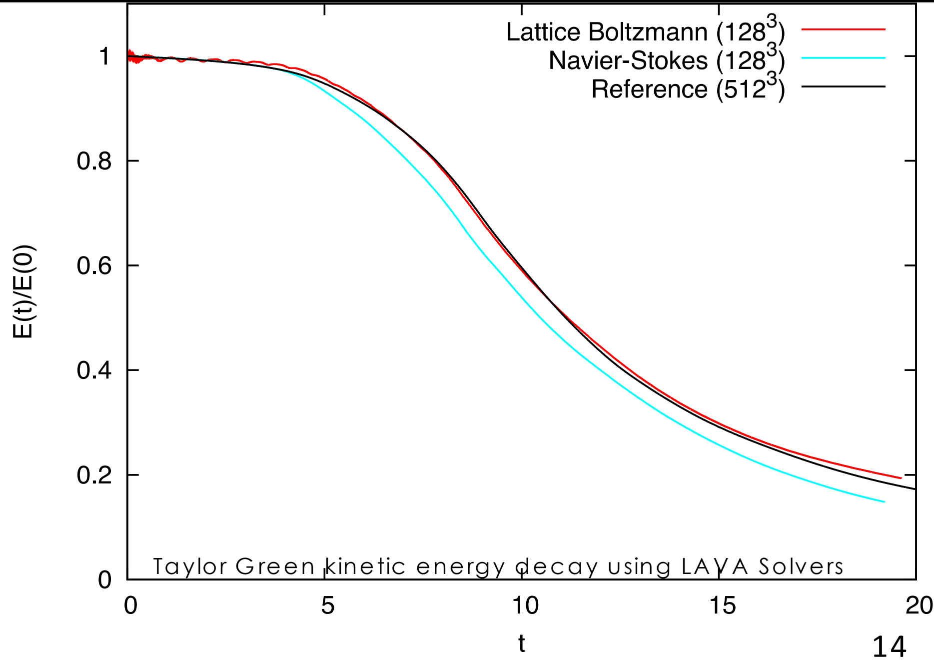
LAVA LBM: Verification and Validation

TURBULENT TAYLOR GREEN VORTEX BREAKDOWN TEST CASE:

- **Motivation**:
  - Simple low speed workshop case for testing high-order solvers
  - Illustrates ability of solver to simulate turbulent energy cascade
  - Periodic boundary conditions
- **Setup**:
  - Analytic initial condition
    - Mach = 0.1
    - Reynolds Number = 1600
  - Triply periodic flow in a box
- **Comparisons:**
  - LAVA's Lattice Boltzmann (LB) solver captures the turbulent kinetic energy cascade from large scales to small scales extremely well.
  - Performance compared to LAVA's Cartesian grid Navier-Stokes WENO solver showed a factor of 50 speedup.



Taylor Green vorticity breakdown. Image credit: 3rd International Workshop on High-Order CFD Methods (Beck et al)



Taylor Green kinetic energy decay using LAVA Solvers

TGV Profiling: Setup

Taylor-Green Vortex (TGV) test case:

- 256^3 cells per node problem size, unless noted otherwise
- Single static level (i.e. no AMR issues)
- No geometry
- 64 time-steps performed, time to solution measured
- All simulations conducted on Skylake nodes on NASA's Pleiades supercomputer (1 node has 2 sockets, 20 physical cores per socket)
- Focused on 3 versions of the code:
  - Baseline (no tiling)
  - Tiling with data copies to tiles
  - Tiling without data copies to tiles

See paper for these results

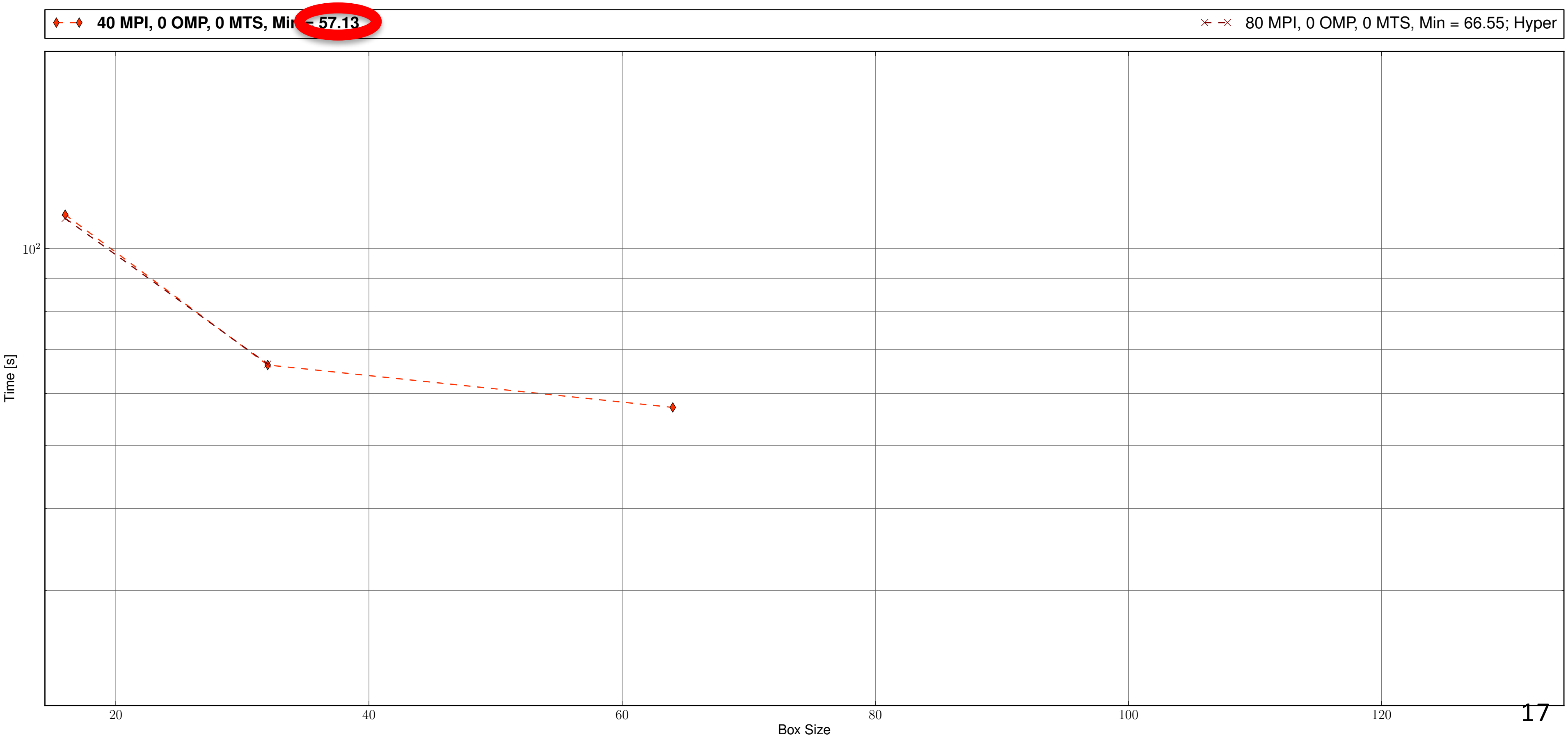
TGV Profiling: Setup

- **Parameter Space and Terminology:**
  - MBS: Max Box Size (i.e. box size) [16,32,64,128,256]
  - MTS: Max Tile Size (i.e. tile size) [0,4,6,8,10,12,16,32,64,128]
  - MPI: Number of ranks / node [1,2,4,8,12,16,20,24,28,32,36,40,50,60,70,80]
  - OMP: Number of OpenMP threads [0,1,2,4,8,12,16,20,24,28,32,36,40,50,60,70,80]
  - Hyper: Hyperthreading (i.e. over-subscribing cores) [no/yes]
  - Nodes: [1,8,64,512]

- **Three profiling analyses were performed:**
  1. <u>Single packed-node parameters study</u>
     → investigate MBS vs MTS vs MPI vs OMP parameter space
  2. <u>Single-node strong scaling study</u>
     → investigate parallel scaling on a single node
  3. <u>Multi-node weak scaling study</u>
     → investigate parallel scaling across nodes, keeping work per node fixed

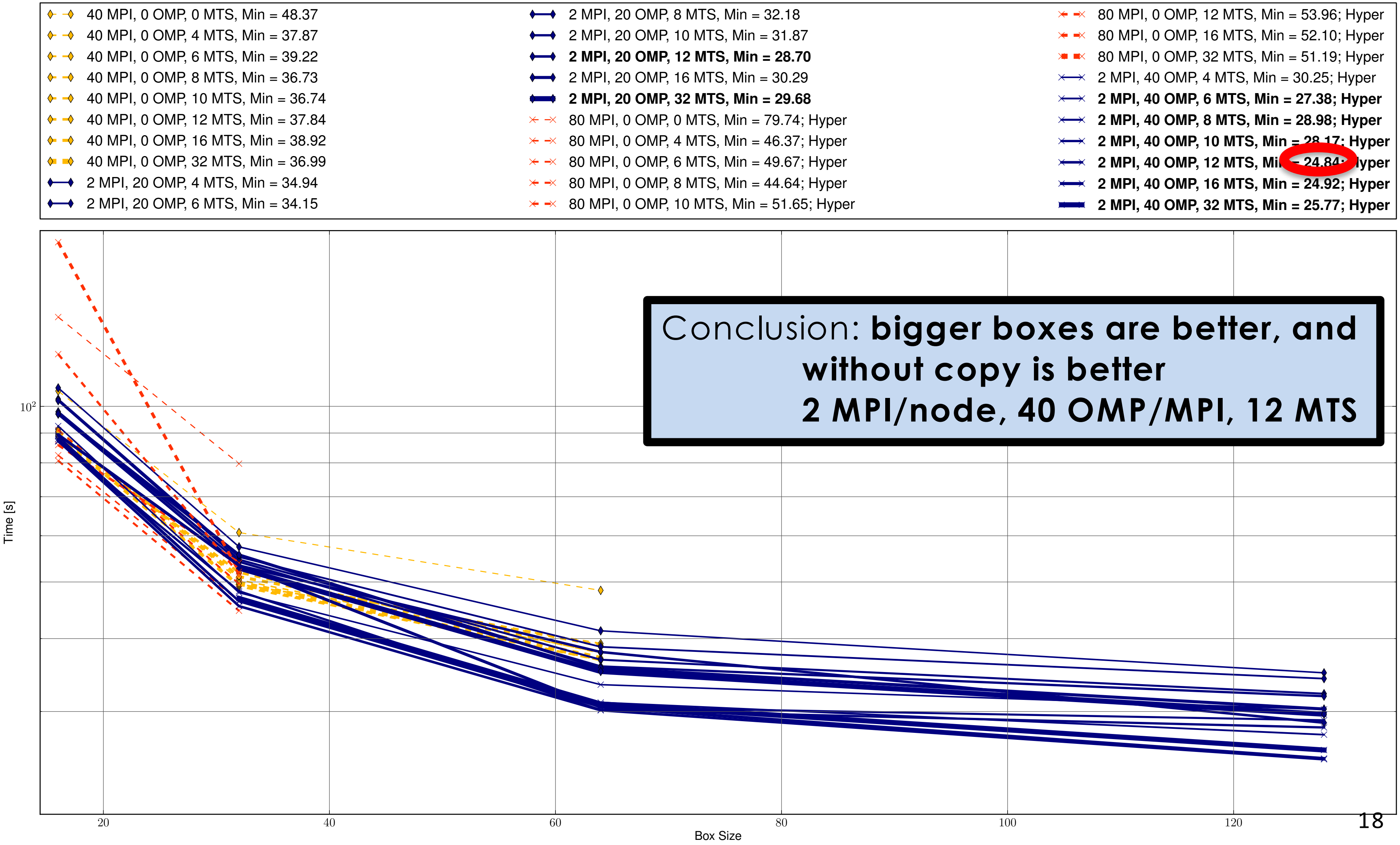
TGV Profiling: Single Packed-Node

Baseline (no tiling), sensitivity to box size (MBS):



TGV Profiling: Single Packed-Node

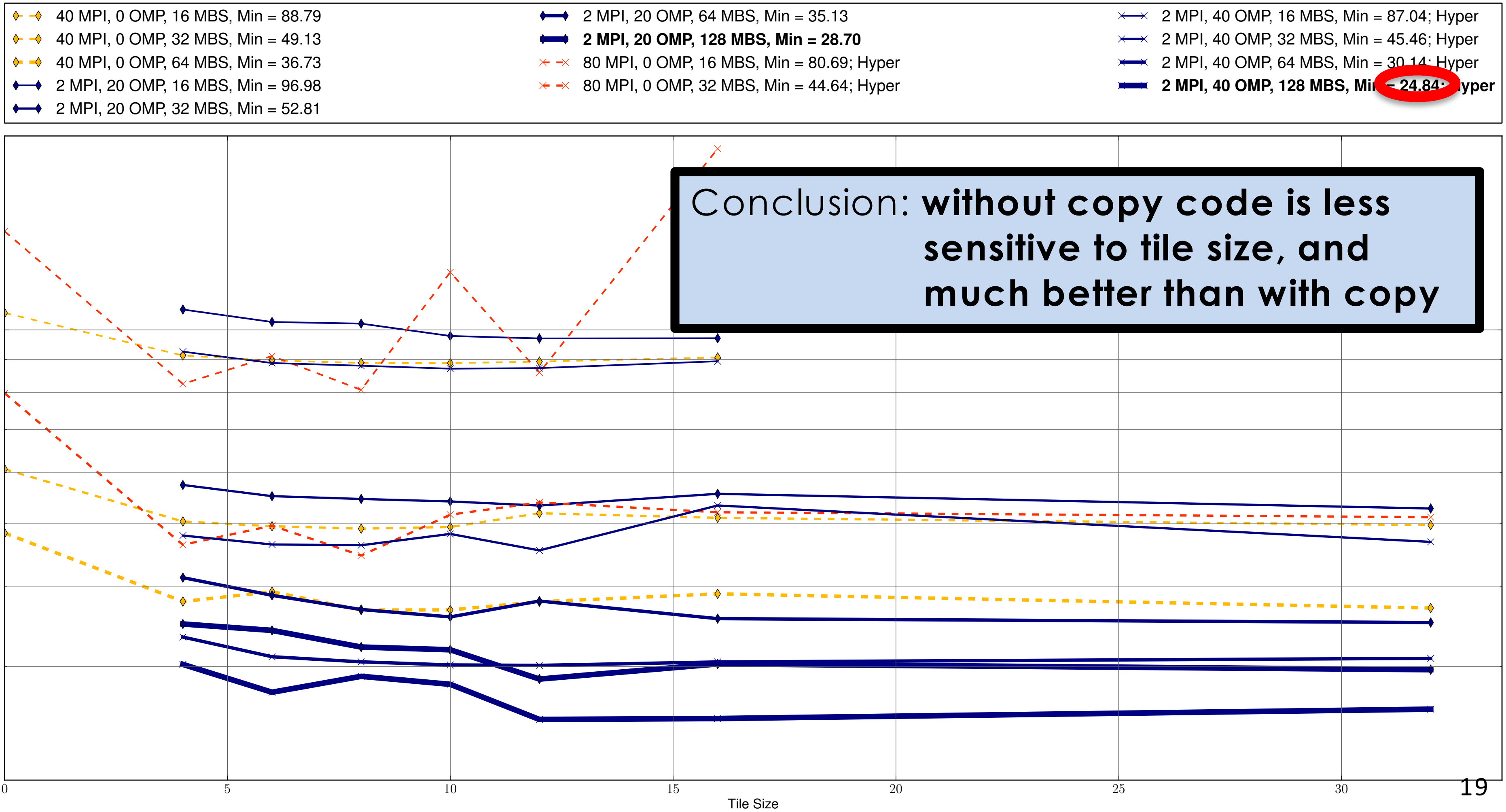
Optimized, **without copy** into tiles, sensitivity to box size (MBS):



Conclusion: **bigger boxes are better, and**
without copy is better
2 MPI/node, 40 OMP/MPI, 12 MTS

TGV Profiling: Single Packed-Node

Optimized, **without copy** into tiles, sensitivity to tile size (MTS):

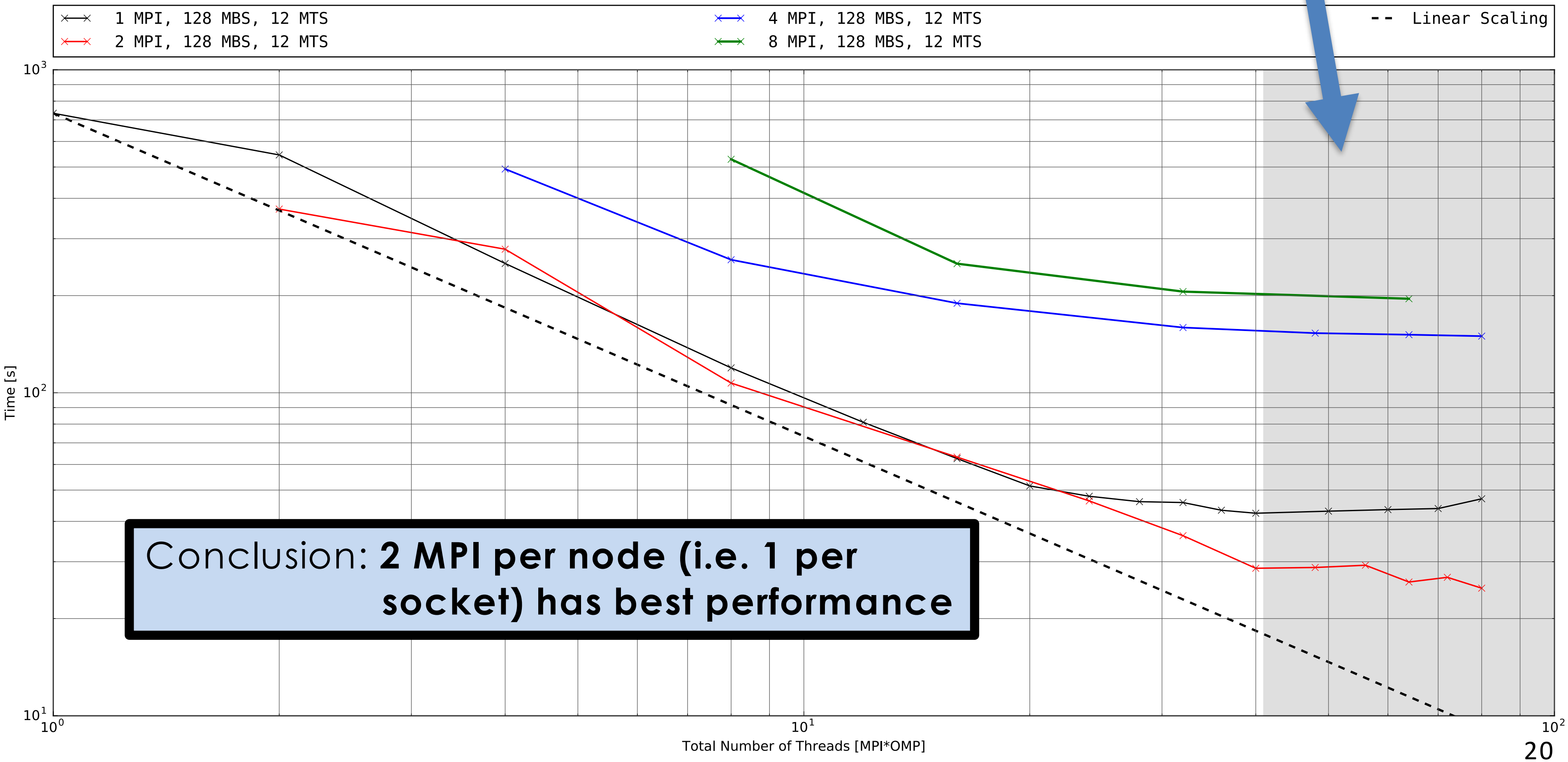


Conclusion: **without copy code is less sensitive to tile size, and much better than with copy**

TGV Profiling: Single-Node Strong Scaling

Optimized, **without copy** into tiles:

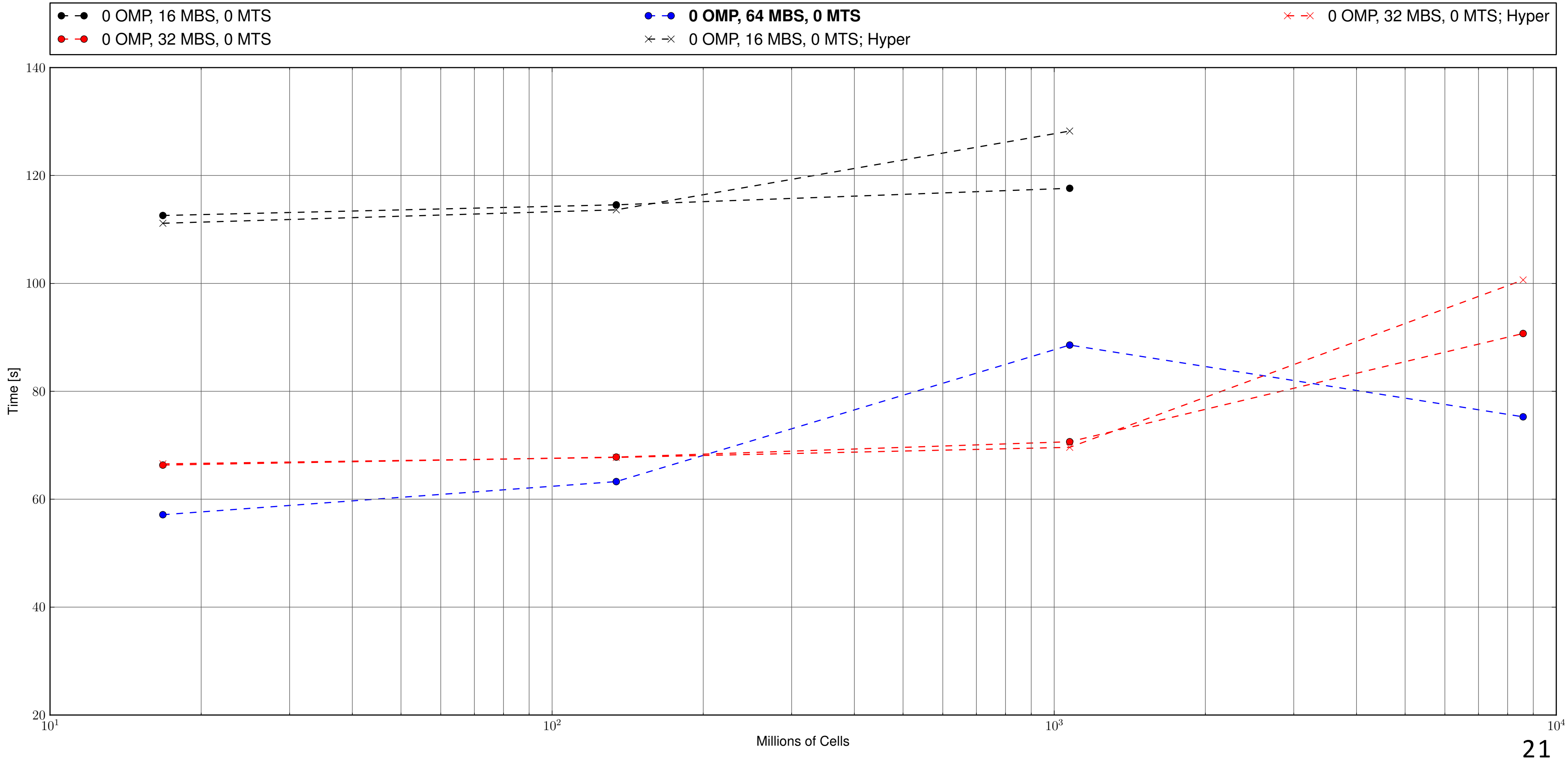
Hyperthreading region is marked with gray shading



Conclusion: **2 MPI per node (i.e. 1 per socket) has best performance**

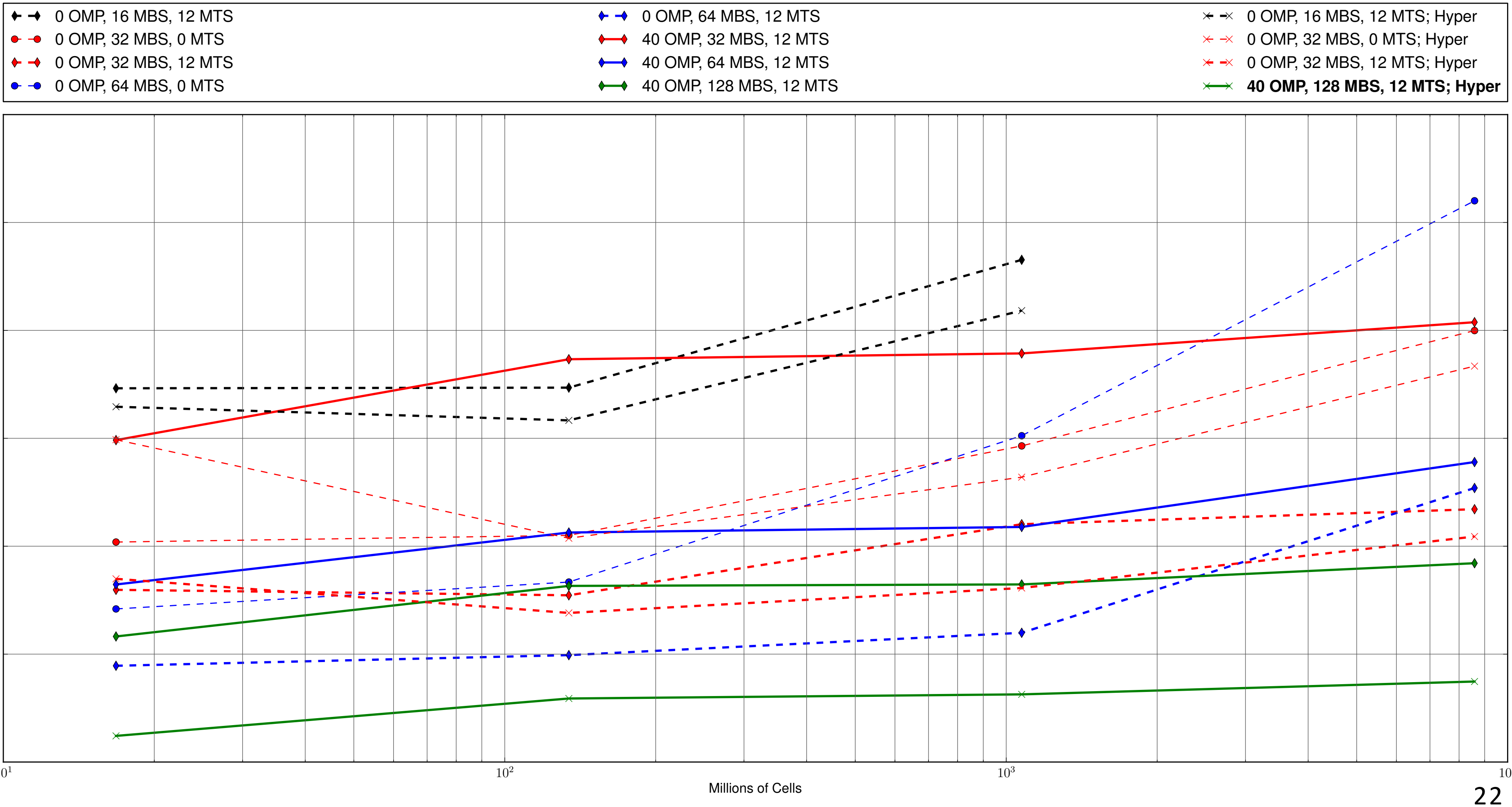
TGV Profiling: Multi-Node Weak Scaling

Baseline (no tiling):



TGV Profiling: Multi-Node Weak Scaling

Optimized, **without copy** into tiles:



TGV Profiling: Multi-Node Weak Scaling

Optimized, **without copy** into tiles:

MPI per Node	OMP	MBS	MTS	HT	Nodes=1 N=256 [s]	Nodes=8 N=512 [s]	Nodes=64 N=1024 [s]	Nodes=512 N=2048 [s]
40	0	16	12	No	89.27	89.39	113.08	-
40	0	32	0	No	60.77	62.01	78.59	99.97
40	0	32	12	No	51.92	50.9	64.08	66.86
40	0	64	0	No	48.37	53.36	80.48	124.03
40	0	64	12	No	37.84	39.81	43.95	70.79
1	40	32	12	No	79.67	94.66	95.73	101.52
1	40	64	12	No	52.91	62.54	63.57	75.6
1	40	128	12	No	43.27	52.64	52.92	56.85
80	0	16	12	Yes	85.85	83.32	103.67	-
80	0	32	0	Yes	79.74	61.48	72.8	93.38
80	0	32	12	Yes	53.96	47.63	52.25	61.79
2	40	128	12	Yes	**24.84**	**31.77**	**32.54**	**34.92**

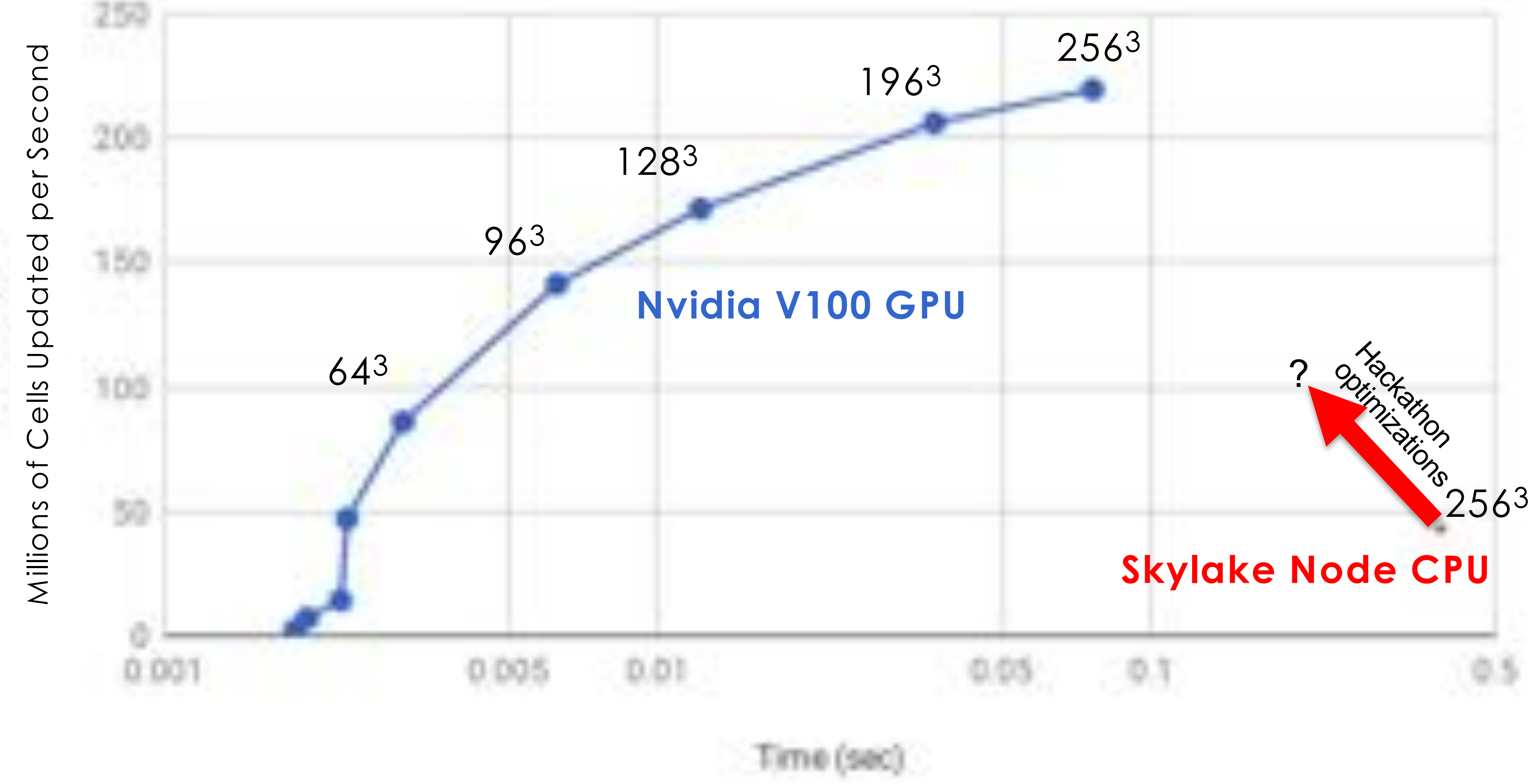
Best practice

~10 billion cells

Bonus: GPU Hackathon 2018

The LAVA team participated in a “GPU Hackathon” in Boulder, CO (06/2018)
Focused on a highly simplified LBM-mini app (single level TGV)

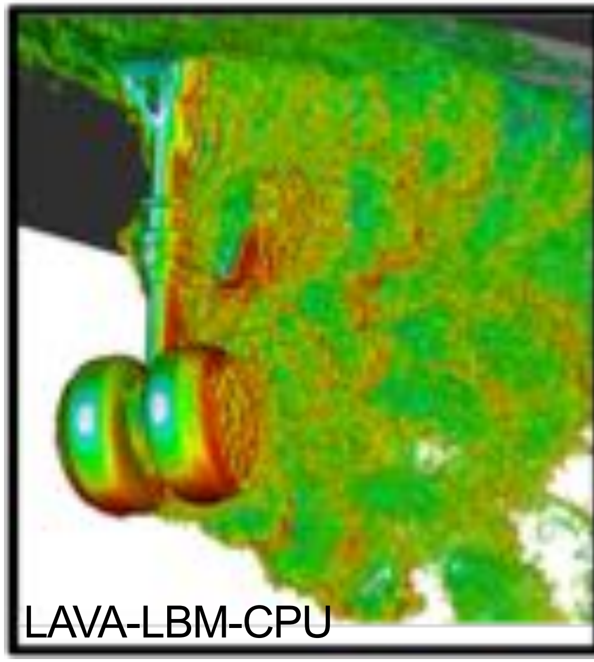
Strong scaling with varying problem size:



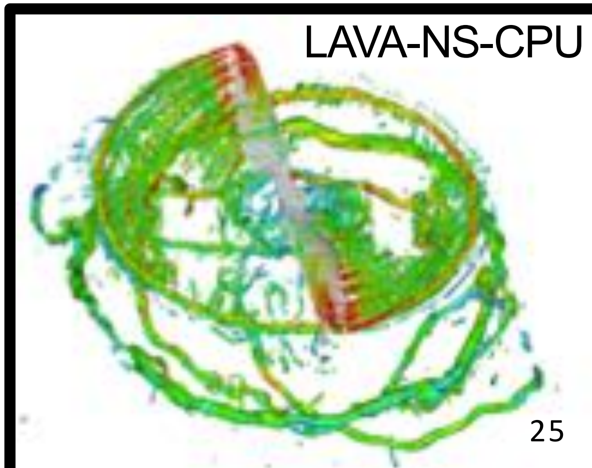
Remaining Challenges for LAVA-LBM-GPU

The following key operations are implemented efficiently on the CPU, but not yet addressed during the hackathon for GPU:

- MPI parallel
- AMR operators
- Immersed boundaries
  - Fixed geometry
    - Introduces load imbalances at both simulation startup and during time-stepping
    - Treated using structured looping in LAVA -> should map to GPU with some effort
  - Moving geometry
    - Major cost / load imbalances are introduced at every timestep (re-computing geometry intersections, etc).
    - Expense on CPU treated using highly optimized vendor supplied ray-tracing kernels (Embree). Enabling technology for CPU calcs.
    - On CPU this is currently roughly a 1.2-1.5x hit in performance, not sure how this will be addressed on the GPU. Try using NVIDIA OptiX.



LAVA-LBM-CPU



LAVA-NS-CPU

TGV Profiling: Summary

For the simple Taylor-Green Vortex problem:

- Found that copying into small tile sized memory is slower than just using the box based memory layout. Not enough re-use in LBM for cache-blocking.

- **Developed best practices**:
    - Larger boxes are better
    - Tile sizes of 8-12 are superior than smaller or larger
    - Hyperthreading yields a small improvement (~1.16x speedup)
    - 1 MPI per socket, 40 OMP threads per socket (i.e. hyperthreaded)

- Achieved a **2.3x speedup over the baseline** code for a single Skylake-SP CPU node containing 40 physical cores,

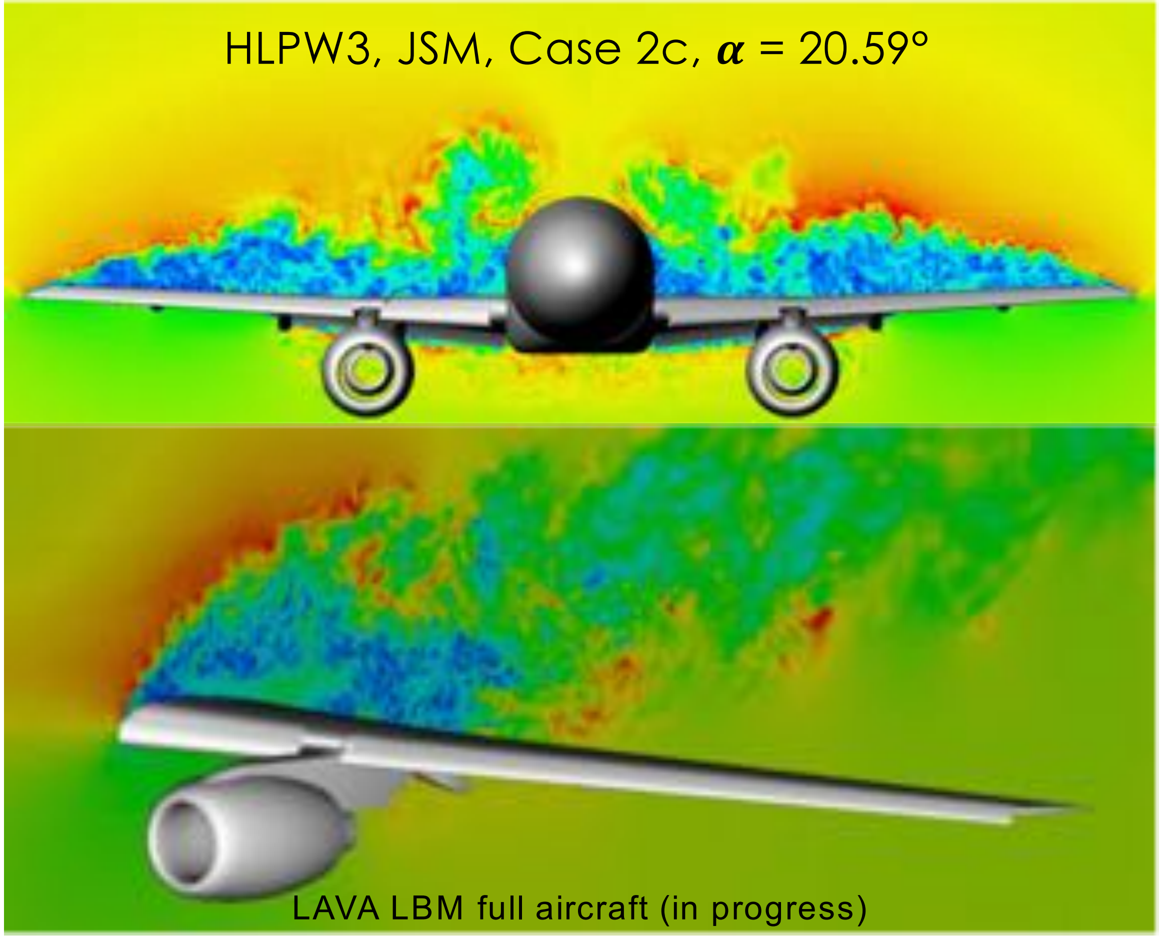
- Achieved a 2.14x speedup over the baseline code for 64 Skylake-SP nodes containing 2560 cores

- **Scaled the code almost perfectly to 20480 physical cores** where the problem size was **~10 billion cells**

- LAVA-LBM-GPU mini-app on Nvidia V100 yielded **11.5x speedup vs CPU baseline**. Could result in O(100)x speedup for full-app vs LAVA-NS-CPU.

Next Steps

- Further code optimizations for:
  - Moving geometry and
  - Adaptive meshing
- Improve wall modeling for arbitrarily complex geometry at high Reynolds numbers
- Extend Mach number range to transonic and high speed flows



HLPW3, JSM, Case 2c, $\alpha = 20.59°$

LAVA LBM full aircraft (in progress)

Acknowledgments

- This work was partially supported by the NASA ARMD's Transformational Tools and Technologies (TTT) project and Revolutionary Computational Aerosciences (RCA) sub-project.

- LAVA team members in the Computational Aerosciences Branch at NASA Ames Research Center for many fruitful discussions

- Computer time provided by NASA Advanced Supercomputing (NAS) facility at NASA Ames Research Center

Questions ?