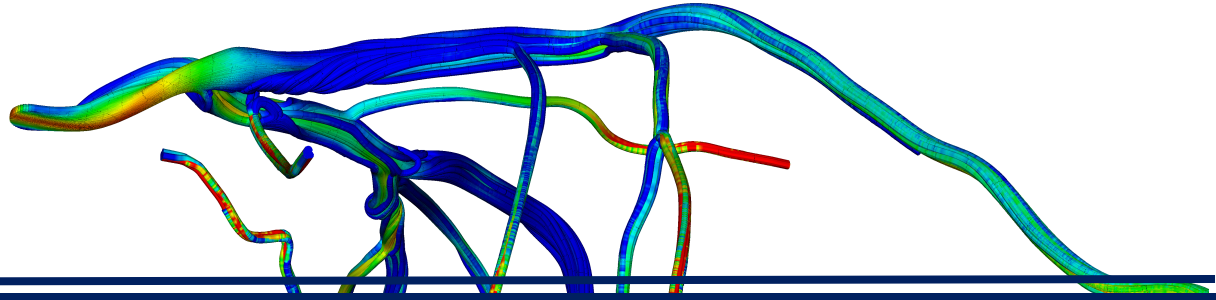


Moment representation in the lattice Boltzmann method on massively parallel hardware

Authors: Madhurima Vardhan[1], John Gounley[2], Luiz Hegele[3], Erik Draeger[4], Amanda Randles[1]

[1]Department of Biomedical Engineering, Duke University

[2]Computational Science and Engineering Division, Oak Ridge National Laboratory

[3]Department of Petroleum Engineering, Santa Catarina State University

[4]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory

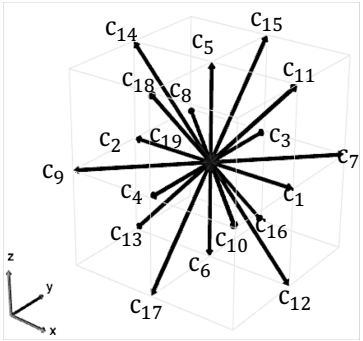
IHPCSS 2019

MOTIVATION

Patient-specific flow models are used to help determine treatment for cardiovascular disease patients but the big restriction is the detail and physiology we can capture which requires high resolution and high fidelity computational runs (~10μm)

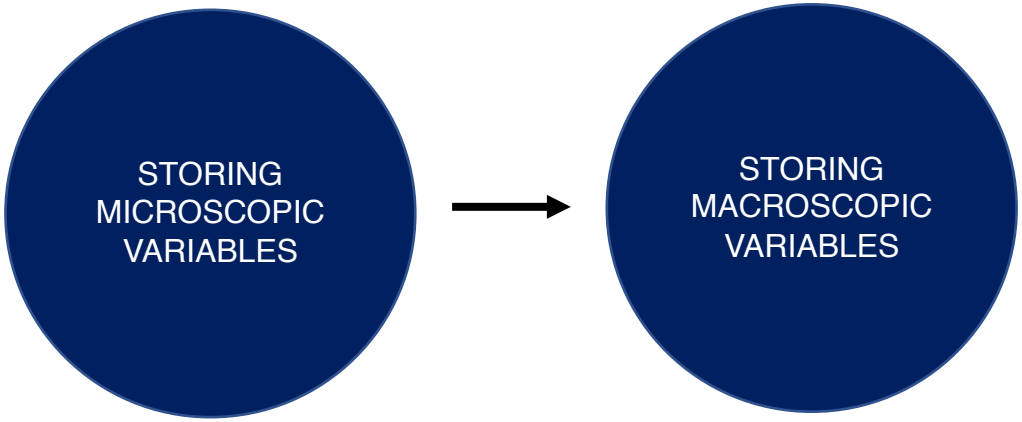
INTRODUCTION

The widely-used lattice Boltzmann method (LBM) for computational fluid dynamics is **highly scalable**, and is characterized by a distribution function $f_i\ (x, t)$.



LBM remains significantly memory bandwidth-bound on current architectures. This work:

- Introduces a new regularized LBM implementation that reduces the memory footprint by only storing macroscopic, moment-based data, reducing memory storage by up to 47% relative to state-of-art LBM implementations.
- Implements a technique for cache-aware data re-utilization to optimize cache utilization and demonstrates similar improvement in time to solution.



NAÏVE SCHEME – $38N^3$ (x8 BYTES)
CURRENT SCHEMES – $20N^3$ (x8 BYTES)
PROPOSED SCHEME – $10N^3$ (x8 BYTES)
~ 4 times memory savings

ALGORITHM

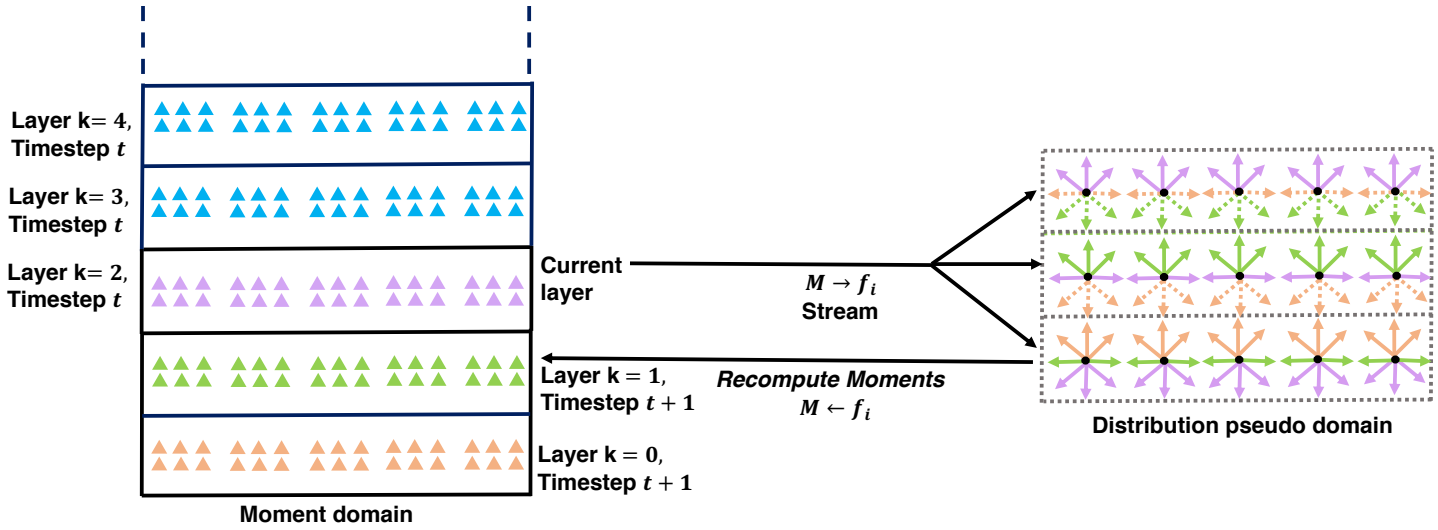
The second-order approximation of the Navier-Stokes equations only requires storing three macroscopic moments (ρ, u and stress tensor Π), which corresponds to 10 double precision values per lattice site.

$$\rho = \sum_{i}^{19} f_i(x,t) \quad \rho u = \sum_{1}^{19} c_i f_i(x,t) \quad \Pi = \sum_{1}^{19} Q_i f_i(x,t)$$

CHALLENGES: Single copy and streaming

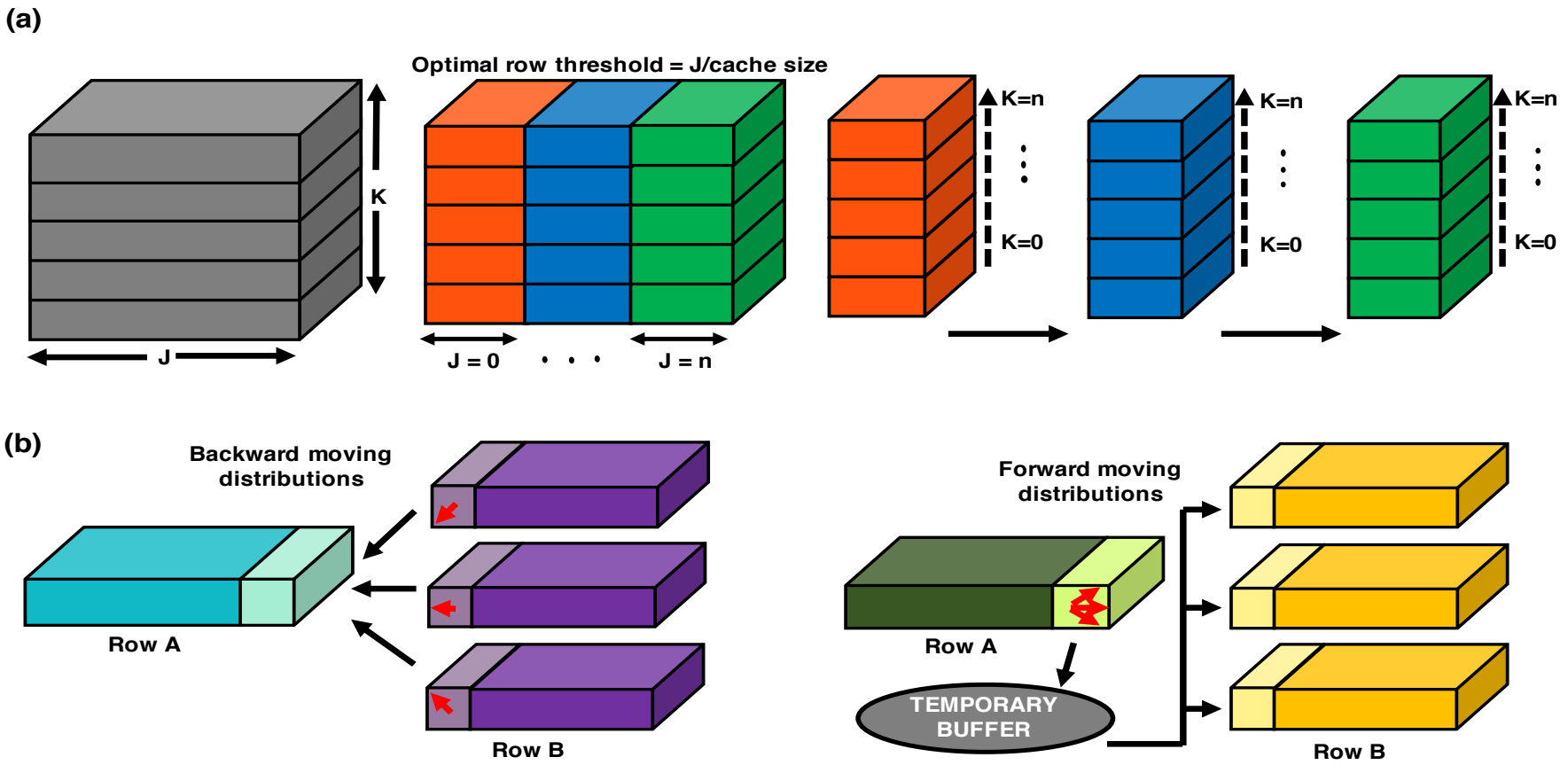
SOLUTION: Introduce a layer-based moment algorithm that stores only macroscopic moment data and uses temporary pseudo domain for streaming distribution data.

$$M = (10 * N + 3 * 19 * P)\ doubles + 19\ * N\ integers$$

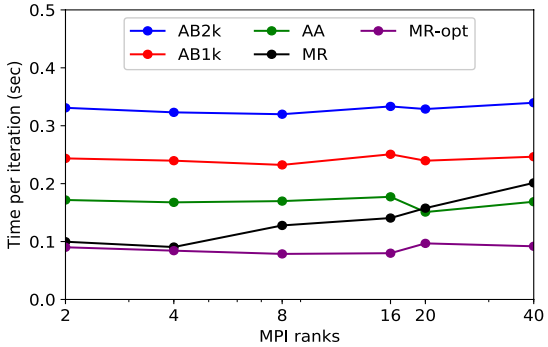


ALGORITHM

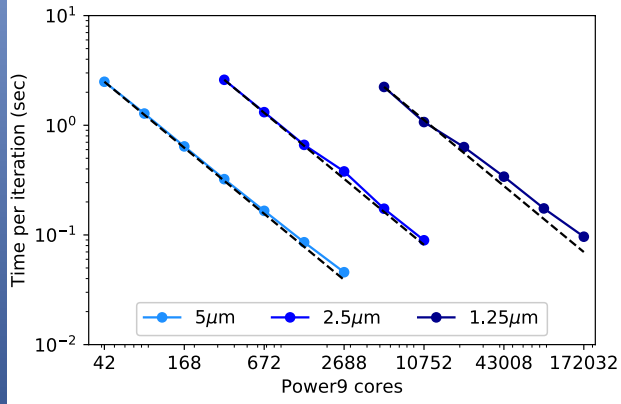
A cache-optimized moment algorithm is presented to enable cache locality and data reuse for the three temporary distribution-based layers.

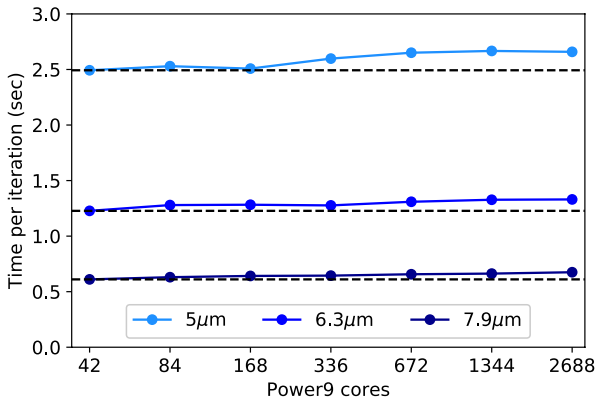


RESULTS

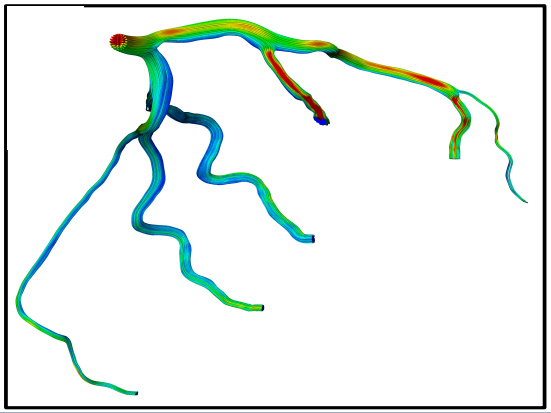


Performance comparison of distribution function vs. moment-based propagation patterns - MPI + OpenMP configurations on a Broadwell node performance.





Large-scale performance - Strong scaling and weak scaling on Summit's Power 9 CPUs for flow in cylindrical geometry at three resolutions.



Real world application - Velocity streamlines in a patient-derived left coronary arterial geometry.