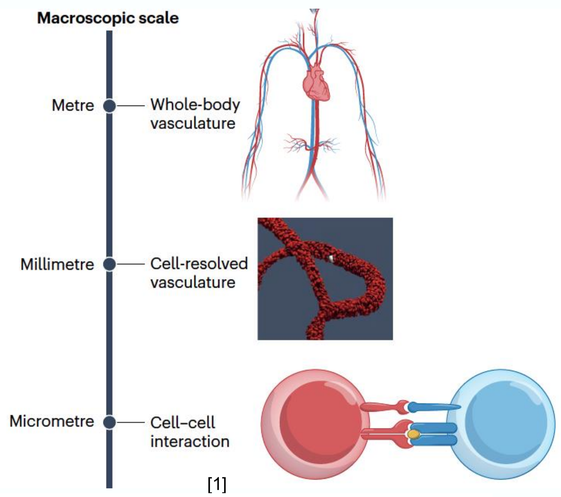
From Millions to Billions: Scaling Explicit Red Blood Cell Modeling on Exascale Systems

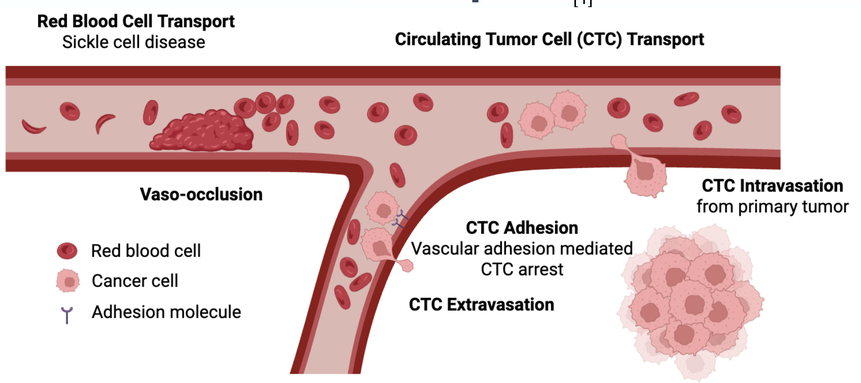
Runxin (Wendy) Wu, Wentao Ma, Jorik Stoop, Amanda Randles
Department of Biomedical Engineering, Duke University



Explicit Cell Modeling in Blood Transport

- **Explicitly resolving individual RBCs** could reveal how microscopic **cellular mechanics and interactions** give rise to macroscopic blood **physiology** and **disease progression** [1].
- While whole-body cell simulation is **computationally intractable**, we aim to **extend the practical limits** of explicit RBC modeling.



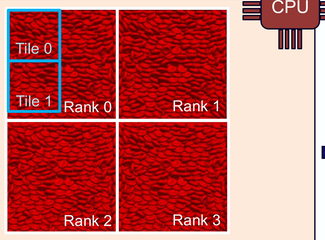


State-of-the-art Red Blood Cell Model Scale

Established Work	RBC Model	Compute System	Blood Volume	Maximum RBC Population
Peters et al., SC10 Gordon Bell finalist [2]	Simplified rigid particle model 6 unknowns per RBC	163840 cores on Blue Gene/P	~1.95 mL	300 million
Rahimian et al., SC10 Gordon Bell winner [3]	Deformable continuum membrane 336 unknowns per RBC	196608 cores on Jaguar PF	~50 uL	262.1 million
Lu et al., SC19 [4]	Deformable continuum membrane 2176 unknowns per RBC 2112 points for collision detection	12288 cores on Stampede2	Not reported	1.05 million
Ames et al., 2020 [5]	Deformable finite element membrane 5778 vertex unknowns per RBC 1280 element unknowns per RBC	1344 CPU cores and 192 NVIDIA V100 GPUs on Summit	~4uL	17 million
This work	Deformable finite element membrane 5778 vertex unknowns per RBC 1280 element unknowns per RBC	601600 CPU cores and 75200 AMD GPUs on Frontier	1mL	4.95 billion

Efficient CPU-GPU Cell Packing & Simulation

Initialization



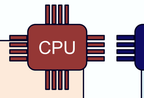
RBC packing with random cell tiles
Resolve cell overlap at task boundary

Cell type data:
- Connectivity, reference configuration, shear modulus, Skalak modulus, bending modulus

Individual cell data:
- Vertex velocity, force, position

5778 vertex unknowns per RBC
1280 element unknowns per RBC

Simulation loop

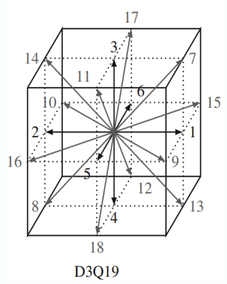


- Fluid data halo exchange
- Cell ownership management
- Cell data halo exchange
- Inlet cell repopulation
- Outlet cell deletion
- Output writing & Checkpointing

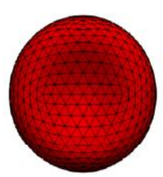
- Fluid collide stream
- Cell velocity interpolation
- Cell position update
- Cell force calculation
- Cell force spread to fluid

HARVEY – Fluid Structure Interaction Solver

Fluid Dynamics
Lattice Boltzmann Method

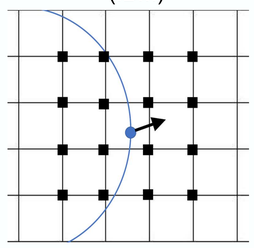


Cell Mechanics
Finite Element Method



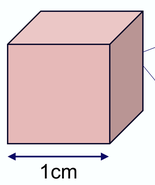
Red Blood Cell(RBC)

Fluid Structure Interaction
Immersed Boundary Method (IBM)

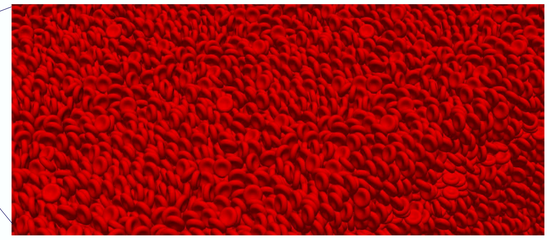


Towards Billions-Scale: 4.95 Billion Explicit Resolved RBC in 1mL Blood

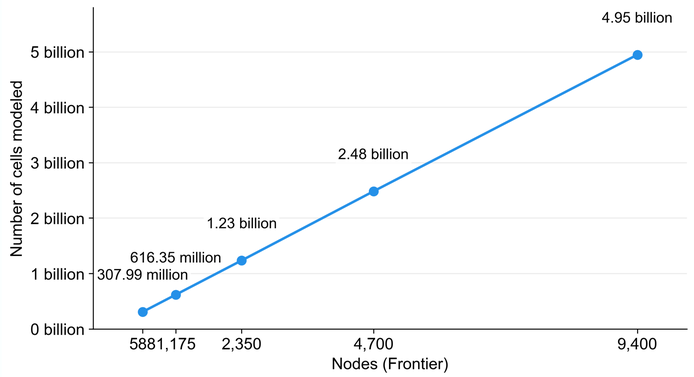
Near full system run on Frontier – 9400 nodes
601600 CPU cores and 75200 AMD GPUs



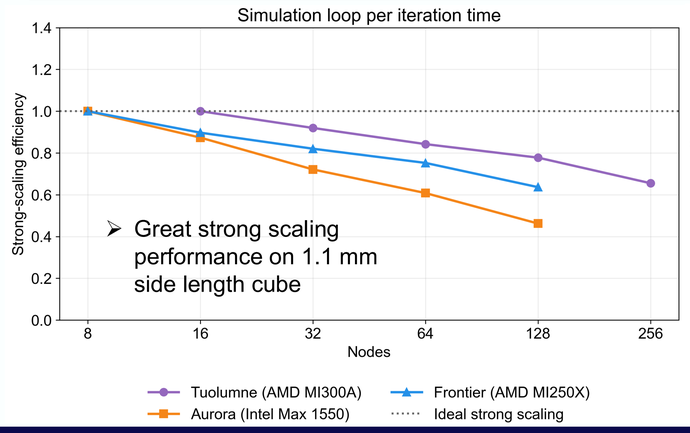
Blood volume: 1mL
Resolution: 1um
Fluid point: 999.4 billion
RBC population: 4.95 billion

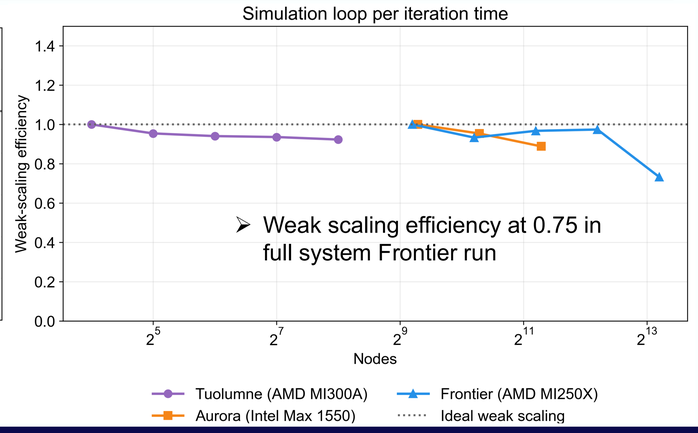


CPU aggregate memory: **3.761 PB**, ~80% usage
GPU aggregate memory: **1.444 PB**, ~30% usage

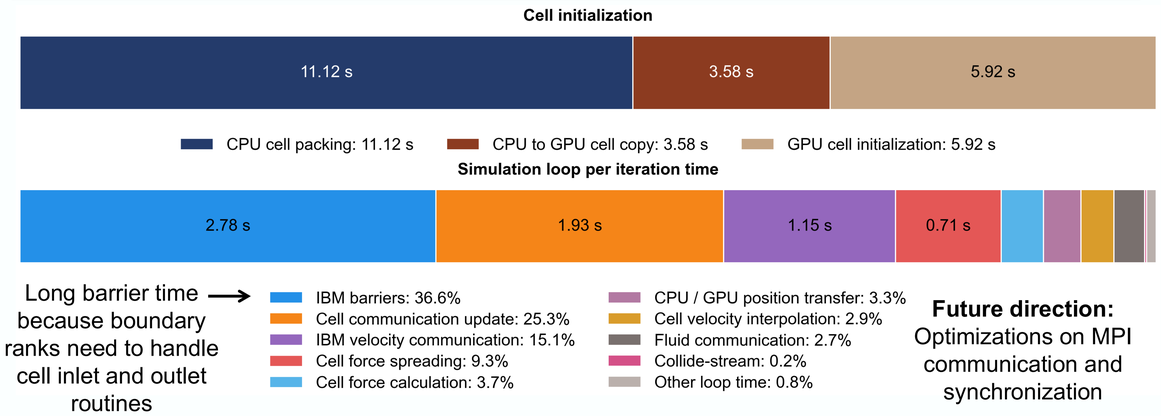


Strong Scaling & Weak Scaling Efficiency





4.95 Billion Cell Run Profiling Breakdown



Long barrier time → because boundary ranks need to handle cell inlet and outlet routines

- IBM barriers: 36.6%
- Cell communication update: 25.3%
- IBM velocity communication: 15.1%
- Cell force spreading: 9.3%
- Cell force calculation: 3.7%
- CPU / GPU position transfer: 3.3%
- Cell velocity interpolation: 2.9%
- Fluid communication: 2.7%
- Collide-stream: 0.2%
- Other loop time: 0.8%

Future direction:
Optimizations on MPI communication and synchronization

References

[1] Wu et al., *Nat. Rev. Bioeng.*, 2026. [2] Peters et al., *SC10*. [3] Rahimian et al., *SC10*. [4] Lu et al., *SC19*. [5] Ames et al., *J. Comput. Sci.*, 2020.

Acknowledgement

Computational resources: DOE INCITE Program; LLNL Grand Challenge; Frontier from Oak Ridge Leadership Computing Facility; Aurora from Argonne National Laboratory; Tuolumne from Lawrence Livermore National Laboratory. Grant support: R01EB024989, DP1AG082343, NSF GRFP DGE 2139754