# Multiscale simulations of complex fluid rheology

Michael P. Howard, Athanassios Z. Panagiotopoulos

*Department of Chemical and Biological Engineering, Princeton University*

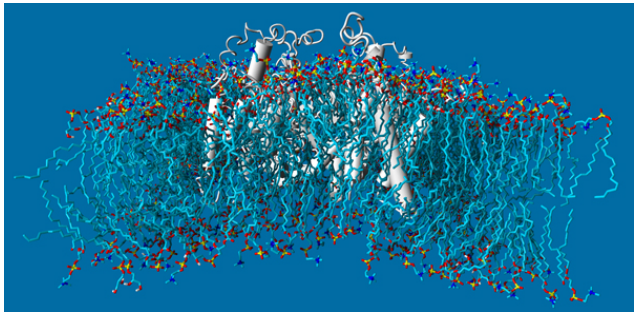
Arash Nikoubashman

*Institute of Physics, Johannes Gutenberg University Mainz*

18 May 2017

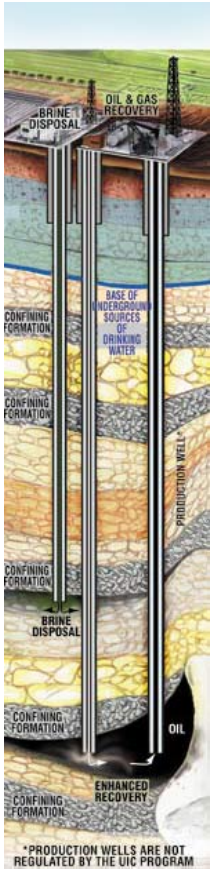
mphoward@princeton.edu

# Complex fluids are everywhere

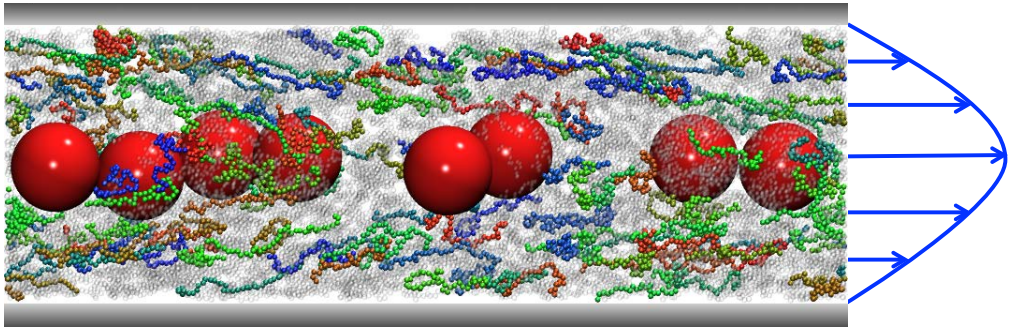




# An industrial example: enhanced oil recovery

Primary recovery: pressure drives the oil from the well (10%)
Secondary recovery: water or gas displaces oil (~30%)
Tertiary (enhanced) oil recovery by gas, water, chemical injection



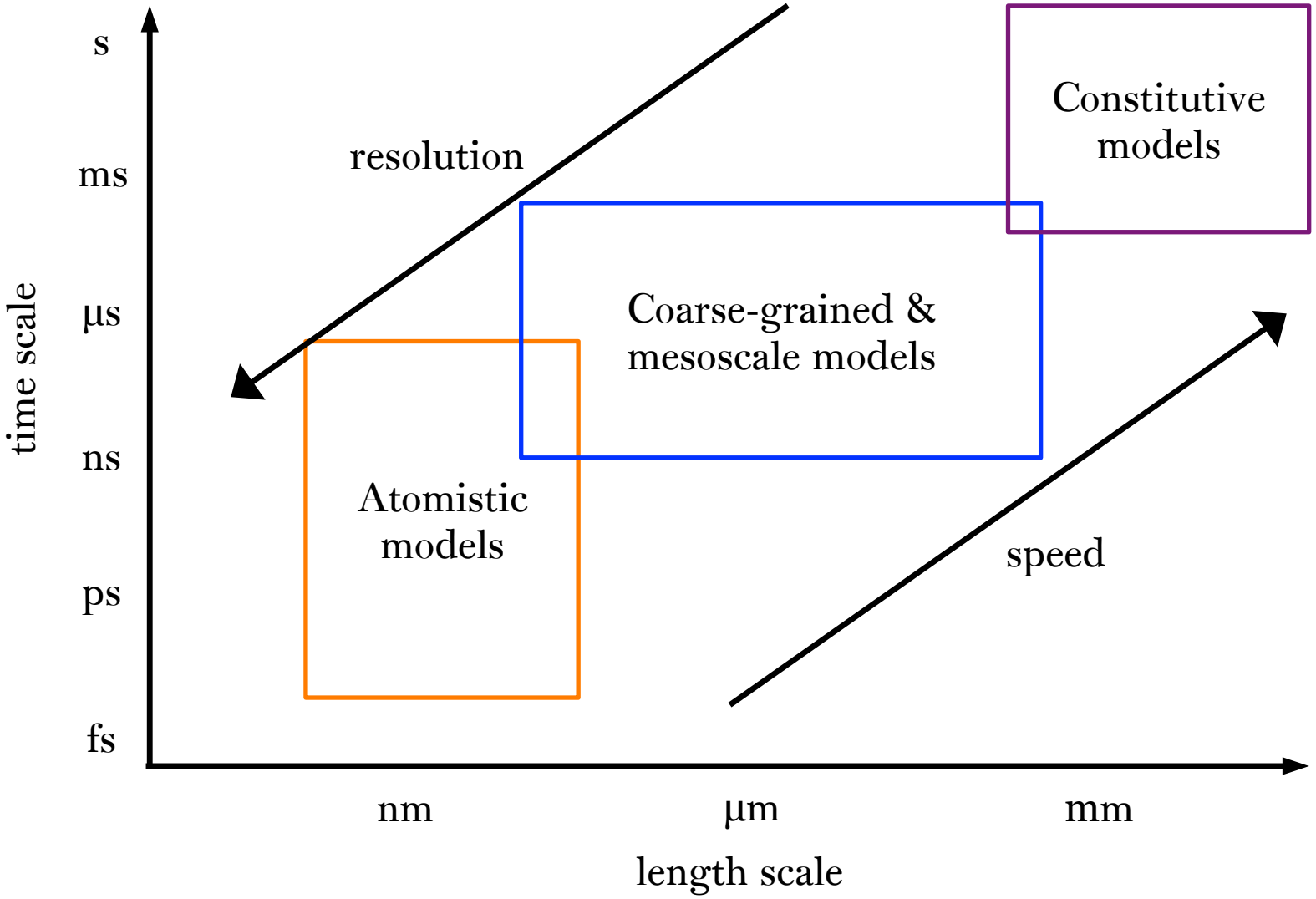

Surfactants, polymers, nanoparticles, etc.



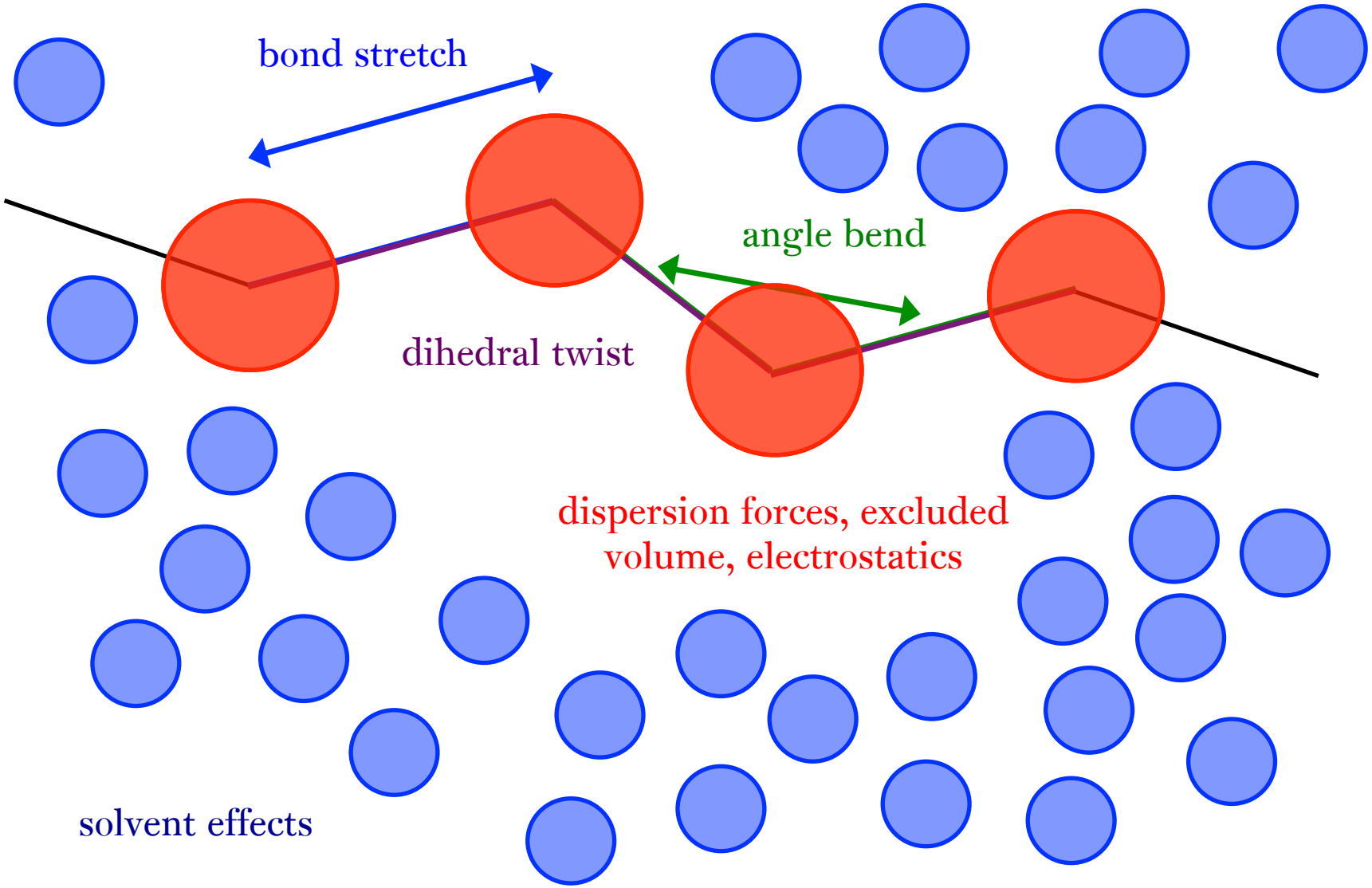
What is the "best" formulation? What fundamental physics control transport of these fluids?

Computer simulations can allow for rapid exploration of the design space for complex fluids

# Multiscale simulation model



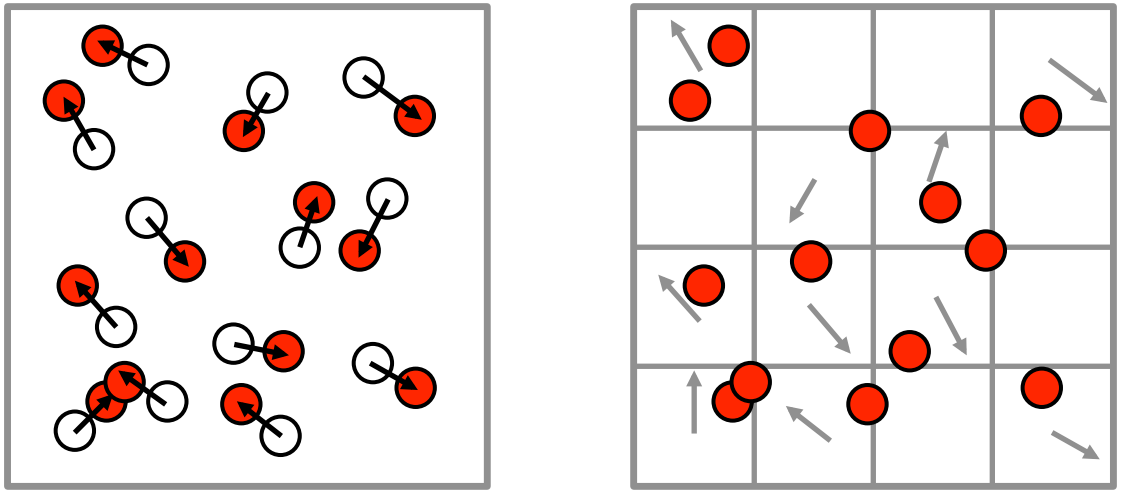
# Multiscale simulation model



# Multiparticle collision dynamics

Mesoscale simulation method that coarse-grains the solvent while preserving hydrodynamics.[1]

Alternate solvent particle *streaming* with *collisions*.



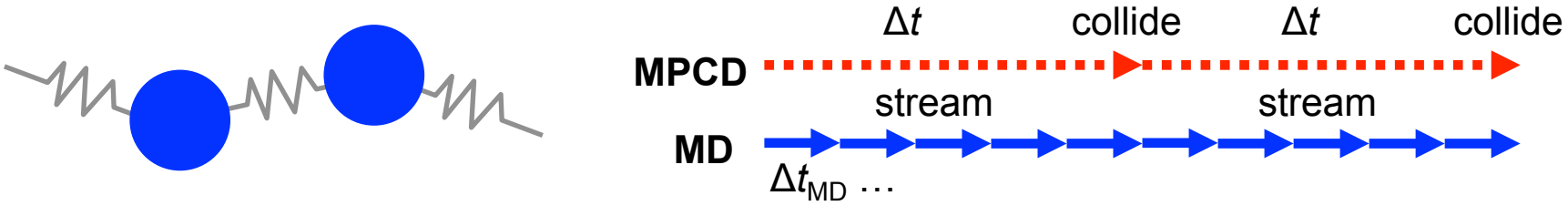
Collisions are momentum- and energy-conserving. An H-theorem exists, and the correct Maxwell-Boltzmann distribution of velocities is obtained. The Navier-Stokes momentum balance is satisfied.



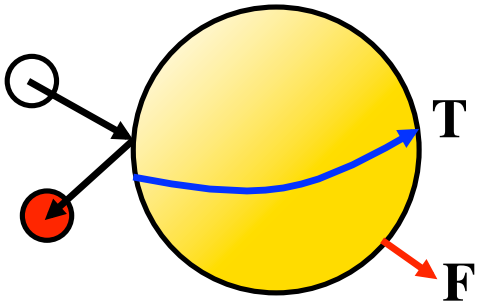
[1] A. Malevanets and R. Kapral. *J. Chem. Phys.* **110**, 8605-8613 (1999).

# Multiparticle collision dynamics

Polymers are coupled to the solvent during the collision step.[2]



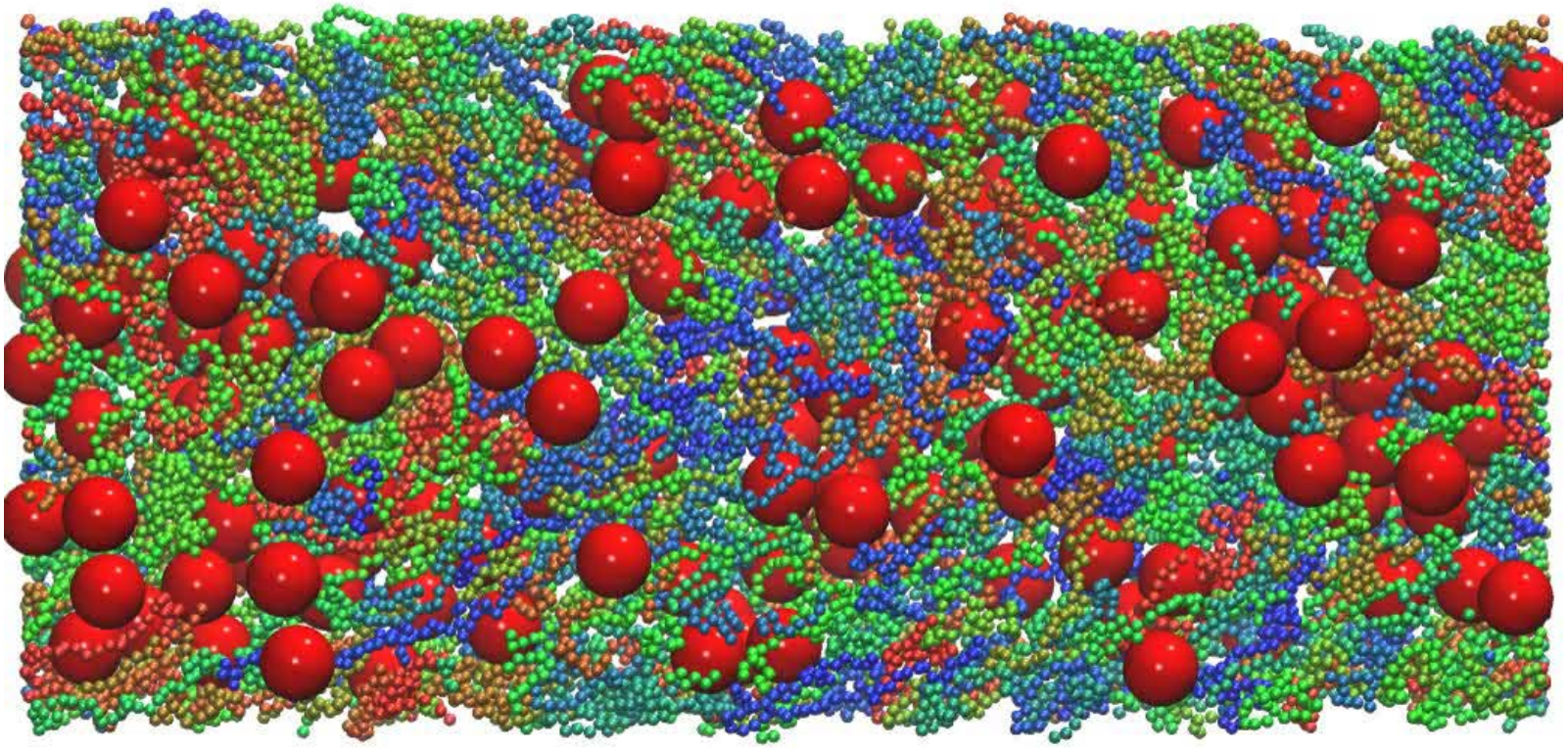

Rigid objects are coupled during the streaming step.[3]






[2] A. Malevanets and J.M. Yeomans. *Europhys. Lett.* **52**, 231-237 (2000).
[3] G. Gompper, T. Ihle, D.M. Kroll, and R.G. Winkler. *Advanced Computer Simulation Approaches for Soft Matter Sciences III*, Advances in Polymer Science Vol. 221 (Springer, 2009), p. 1-87.

# Multiparticle collision dynamics



PRINCETON UNIVERSITY

[4] A. Nikoubashman, N.A. Mahynski, A.H. Pirayandeh, and A.Z. Panagiotopoulos. *J. Chem. Phys.* **140**, 094903 (2014).

[5] M.P. Howard, A.Z. Panagiotopoulos, and A. Nikoubashman, *J. Chem. Phys.* **142**, 224908 (2015).

# Why Blue Waters?

Typical MPCD simulations require <u>many</u> particles with simple interactions.

( 10 particles per cell ) x ( $100^3$ - $400^3$ cells ) = 10–640 million particles

MPCD readily lends itself to parallelization because of its particle- and cell-based nature:

1. Accelerators (e.g., GPU) within a node.
2. Domain decomposition to many nodes.

For molecular dynamics, GPU acceleration can significantly increase the scientific throughput by as much as an order of magnitude.

Blue Waters is the <u>only</u> NSF-funded system currently giving access to GPU resources at massive scale!

# Implementation

HOOMD-blue simulation package[6,7] for design philosophy and flexible user interface.

```
import hoomd
hoomd.context.initialize()
from hoomd import mpcd

# initialize an empty hoomd system first
box = hoomd.data.boxdim(L=200.)
hoomd.init.read_snapshot(hoomd.data.make_snapshot(N=0, box=box))

# then initialize the MPCD system randomly
s = mpcd.init.make_random(N=int(10*box.get_volume()), kT=1.0, seed=7)
s.sorter.set_period(period=100)

# use the SRD collision rule
ig = mpcd.integrator(dt=0.1, period=1)
mpcd.collide.srd(seed=1, period=1, angle=130.)

# run for 5000 steps
hoomd.run(5000)
```

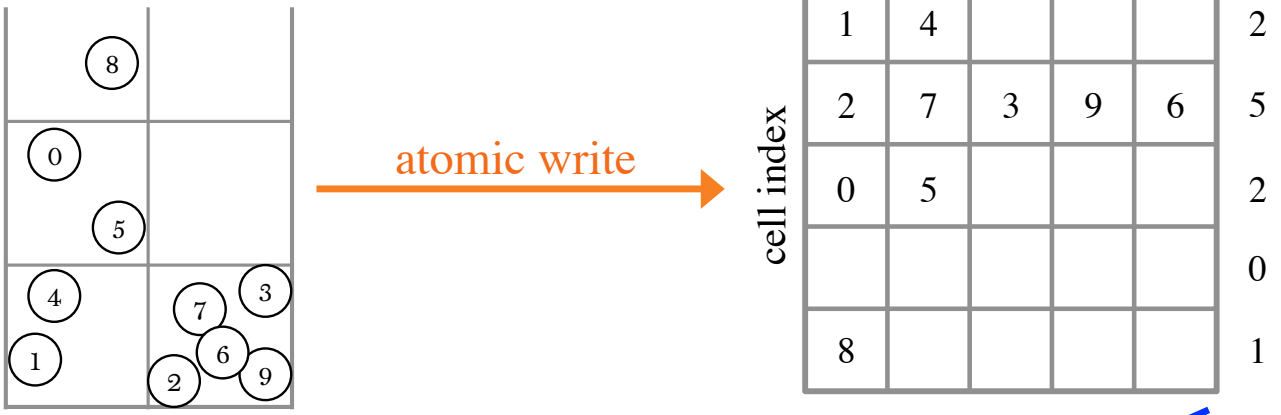
HOOMD-blue is available open-source for use on Blue Waters:

http://glotzerlab.engin.umich.edu/hoomd-blue

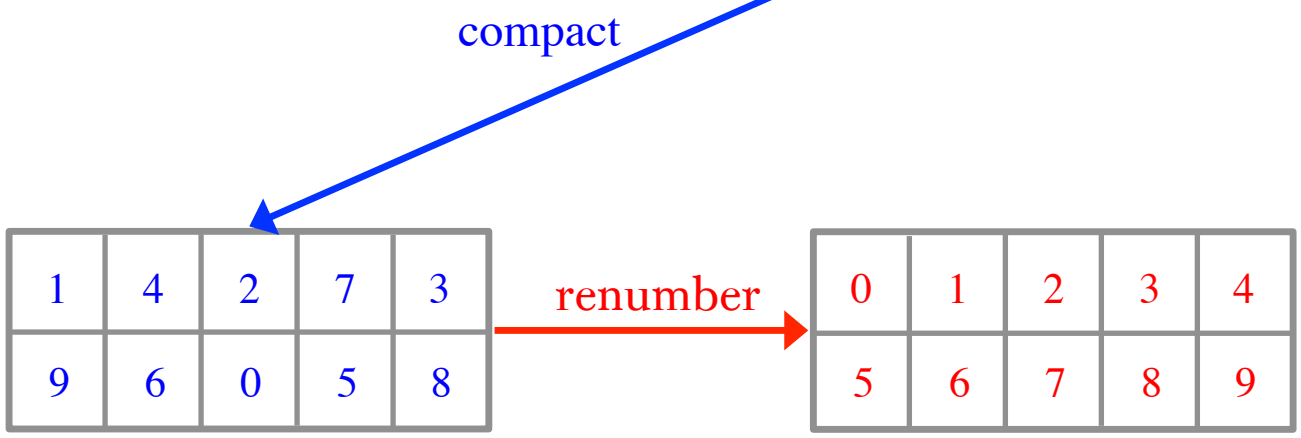


[6] J.A. Anderson, C.D. Lorenz, and A. Travesset. *J. Comput. Phys.* **227**, 5342-5359 (2008).
[7] J. Glaser et al. *Comput. Phys. Commun.* **192**, 97-107 (2015).

# Implementation: cell list

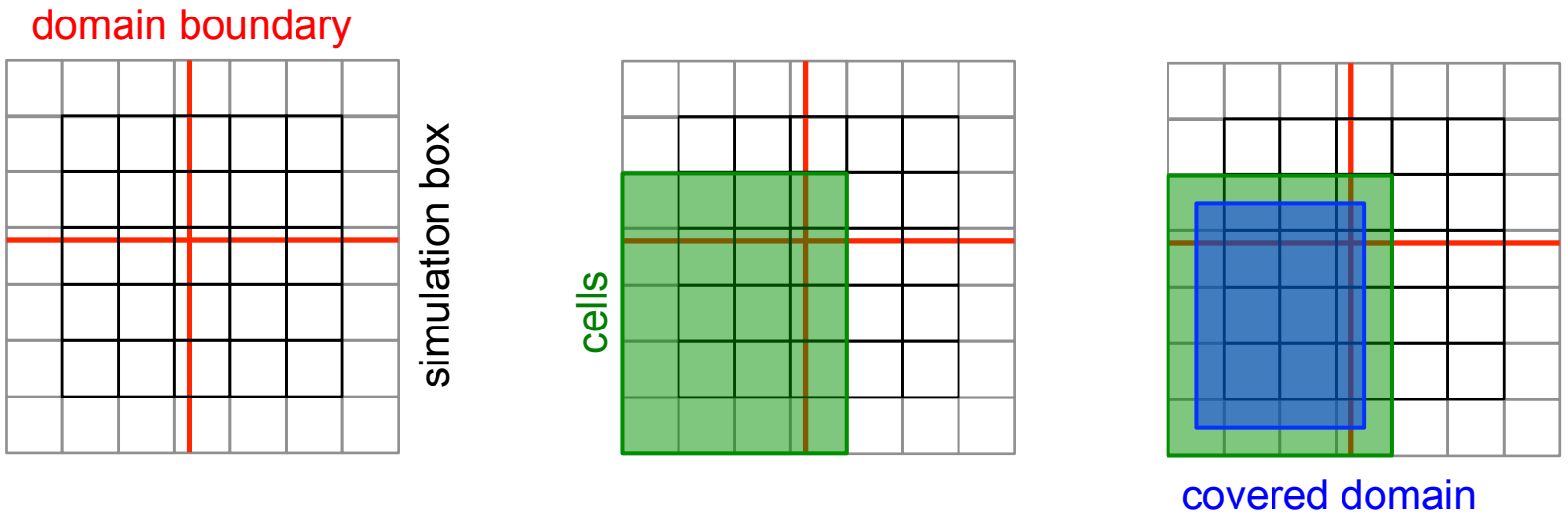



Improve performance by periodically sorting particles.



# Implementation: domain decomposition

HOOMD-blue supports non-uniform domain boundaries. To simplify integration with molecular dynamics code, we adopt the same decomposition.

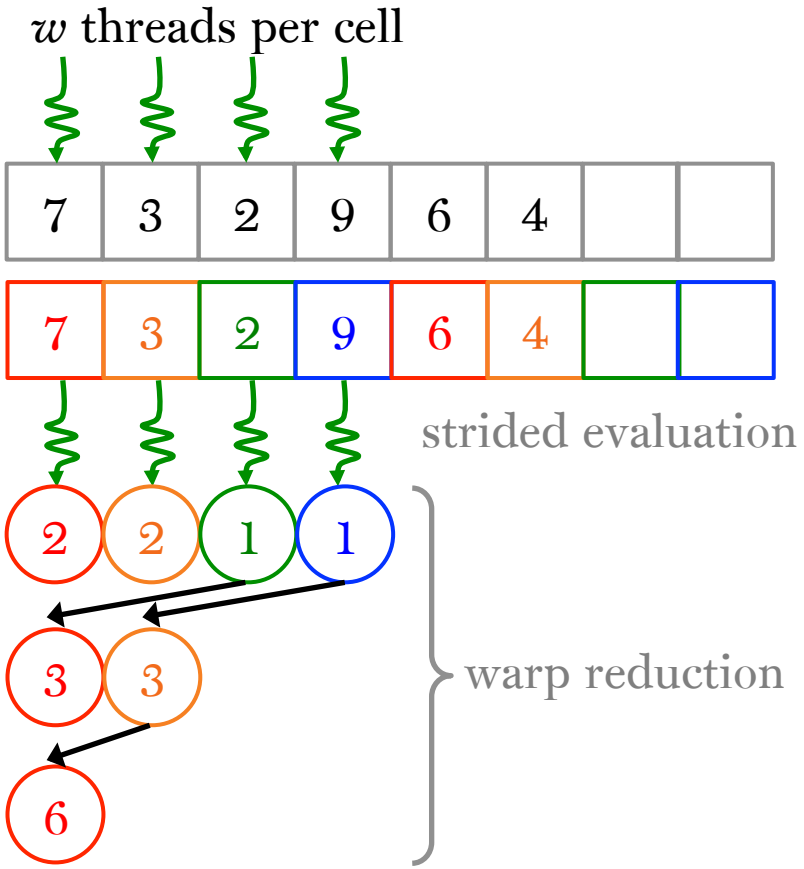



Any cells overlapping between domains require communication for computing cell-level averages, like the velocity or kinetic energy.

All MPI buffers are packed and unpacked on the GPU. Some performance can be gained from using CUDA-aware MPI when GPUDirect RDMA is available.

# Implementation: CUDA optimizations

## Wide parallelism




## Runtime autotuning

Significant performance variation from size of the thread-block that is architecture and system-size dependent.

| | block size | $w$ |
|---|---|---|
| K20x | 448 | 4 |
| P100 | 224 | 16 |

*Solution*: use periodic runtime tuning with CUDA event timings to scan through and select optimal parameters.

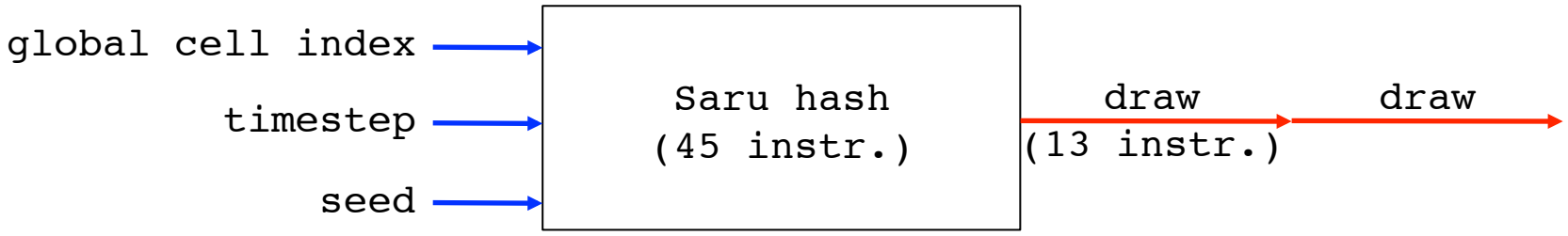
```
m_tuner->begin();
gpu::cell_thermo(..., m_tuner->getParam());
m_tuner->end();
```

Only a small fraction of time is spent running at non-optimal parameters for most kernels.

# Implementation: random number generation

Need efficient random number generation for the collision step.

One PRNG state is created per-thread and per-kernel using a cryptographic hash.[8] Micro-streams of random numbers are then drawn.




The additional instructions needed to prepare the PRNG state can be more efficient than loading a state from memory on the GPU.

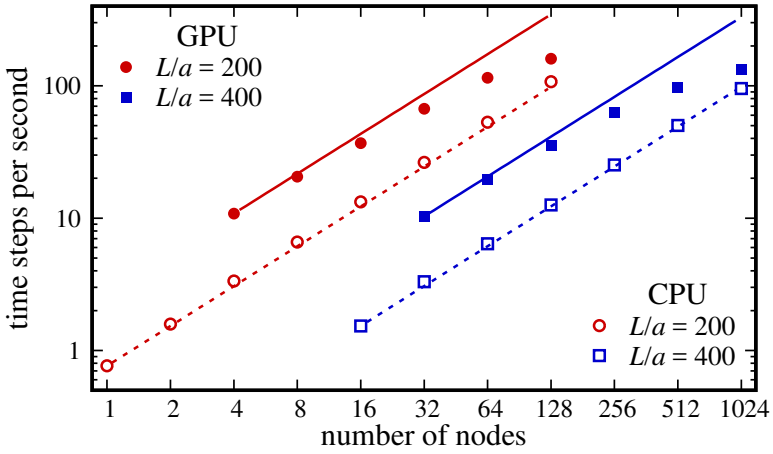
Additional benefit: significantly reduced cell-level MPI communication since no synchronization of random numbers is required.

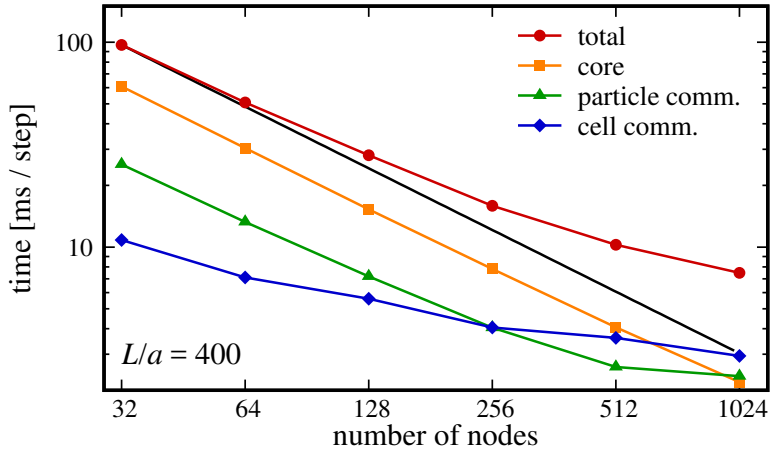


[8] C.L. Phillips, J.A. Anderson, and S.C. Glotzer. *J. Comput. Phys.* **230**, 7191-7201 (2011).

# Performance on Blue Waters

Strong scaling tested for $L = 200$ and $L = 400$ cubic boxes with 10 particles / cell (80 million and 640 million particles).




CPU benchmarks (16 processes per XE node) show excellent scaling.

GPU benchmarks (1 process per XK node) show good scaling, with some loss of performance at high node counts.

GPU profile shows that performance is bottlenecked by cell communication.

# Future plans

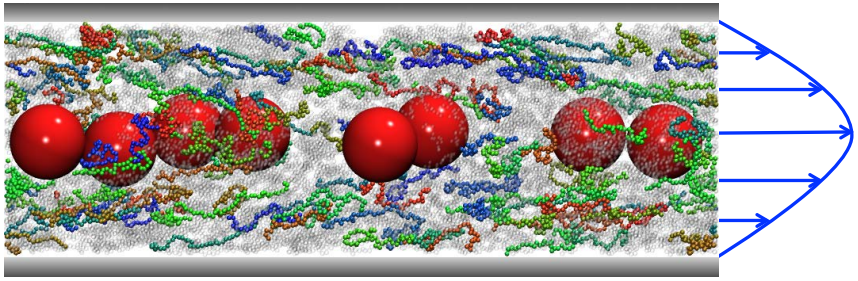
Performance enhancements

1. Overlap outer cell communication with inner cell computations.
2. Test a mixed-precision model for floating-point particle and cell data.

Feature enhancements

1. Implement additional MPCD collision rules for a multiphase fluid.
2. Couple particles to solid bounding walls.

*Scientific application:* Compare MPCD multiphase fluid to MD simulations, and study droplet physics in confined complex fluid.



All code is available online! https://bitbucket.org/glotzer/hoomd-blue

BLUE WATERS SUSTAINED PETASCALE COMPUTING

NVIDIA

NATIONAL SCIENCE FOUNDATION MRSEC MATERIALS RESEARCH SCIENCE & ENGINEERING CENTERS