These slides + links to papers: hpcgarage.org/dpu-isc23

Principles and practice of algorithm design on DPU systems

DPU.ORNL.GOV @ ISC23

RICH VUDUC – MAY 25

Georgia Tech College of Computing
School of Computational Science and Engineering

Georgia Tech College of Computing
Center for Research into Novel Computing Hierarchies

These slides + links to papers: hpcgarage.org/dpu-isc23

Embracing communication

(A post-pandemic talk)

DPU.ORNL.GOV @ ISC23

RICH VUDUC – MAY 25

Georgia Tech College of Computing
School of Computational Science and Engineering

Georgia Tech College of Computing
Center for Research into Novel Computing Hierarchies

First, principles

Recall: “The” dominant paradigm of CS:

$$\mathcal{O}(N^2) \longrightarrow \mathcal{O}(N)$$

Reduces **energy**: fewer (fl)ops, less storage

Recall:

$$\mathcal{O}(N^2) \longrightarrow \boxed{\mathcal{O}(N)}$$

% time communicating increases

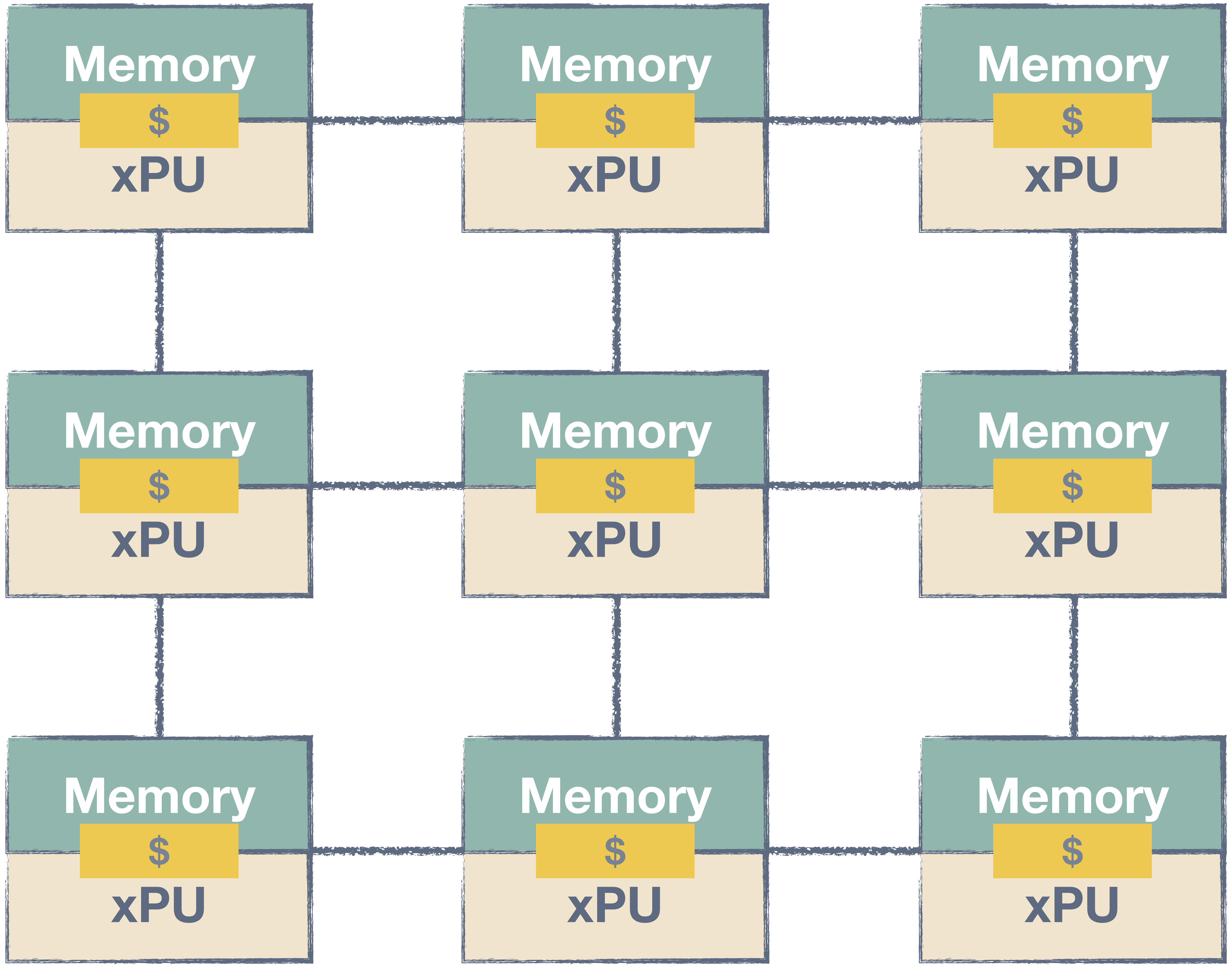
An Iron Law of Parallel and Distributed Computation

A modern cluster or supercomputer is, to first order, a collection of processing nodes. Each node has a processor ("xPU") and a two-level memory hierarchy. Nodes are connected by a network.

As a program executes on this system, it incurs two types of communication cost.

"Vertical" communication occurs in the memory system between, say, RAM and cache.

"Horizontal" communication occurs between nodes across the network.



Two costs: $T_{network} + T_{memory}$

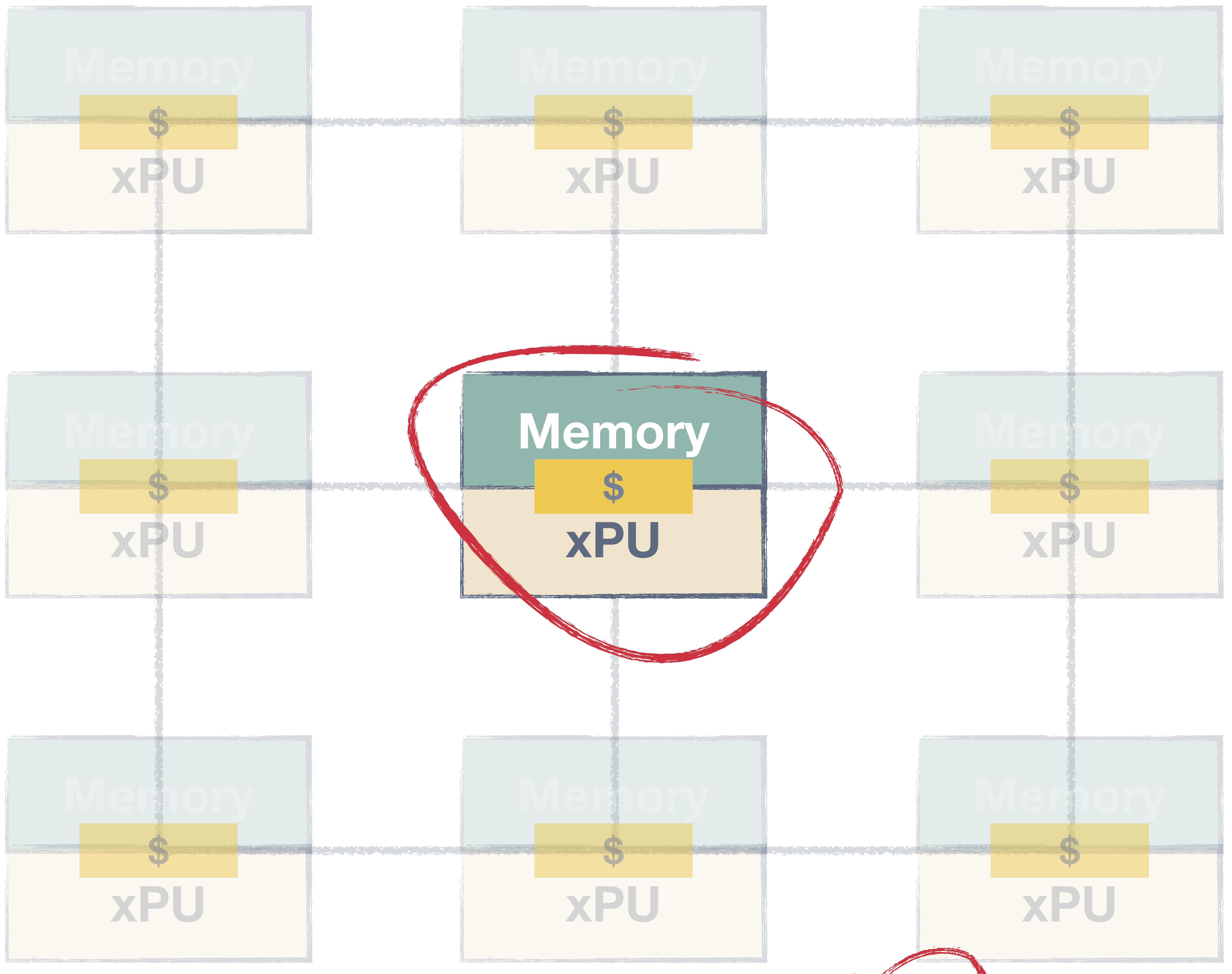
An Iron Law of Parallel and Distributed Computation

A modern cluster or supercomputer is, to first order, a collection of processing nodes. Each node has a processor ("xPU") and a two-level memory hierarchy. Nodes are connected by a network.

As a program executes on this system, it incurs two types of communication cost.

"Vertical" communication occurs in the memory system between, say, RAM and cache.

"Horizontal" communication occurs between nodes across the network.



Two costs: $T_{network} + T_{memory}$

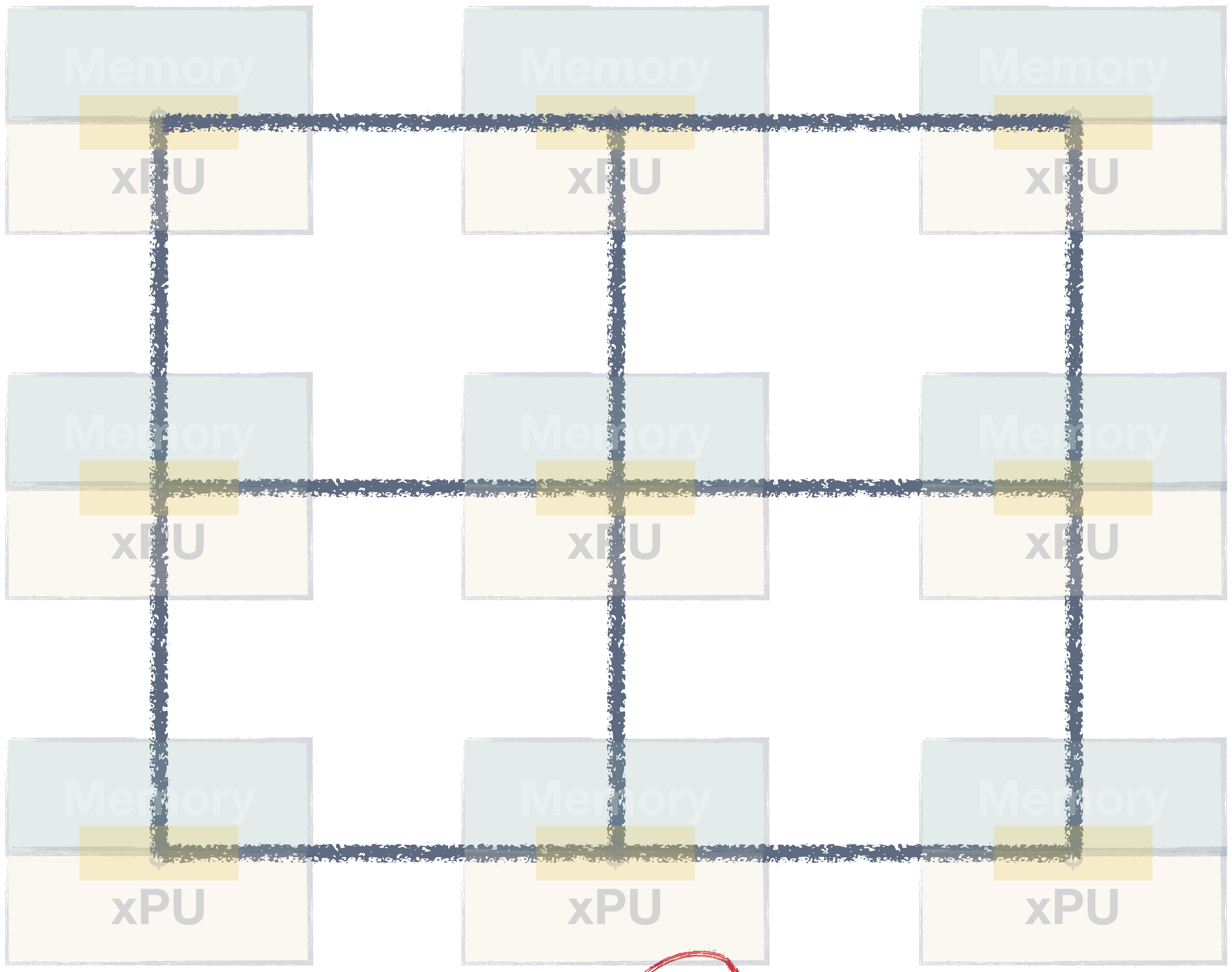
An Iron Law of Parallel and Distributed Computation

A modern cluster or supercomputer is, to first order, a collection of processing nodes. Each node has a processor ("xPU") and a two-level memory hierarchy. Nodes are connected by a network.

As a program executes on this system, it incurs two types of communication cost.

"Vertical" communication occurs in the memory system between, say, RAM and cache.

"Horizontal" communication occurs between nodes across the network.



Two costs: $T_{network} + T_{memory}$

(Asymptotic running time – rules-of-thumb)

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time

$$\frac{W(n)}{P}$$

Jeff Young (GT), R.V. (2016) – “Finding balance in post-Moore’s Law era.” [link]

(Asymptotic running time – rules-of-thumb)

Compute time

$$\frac{W(n)}{P}$$

P-fold speedup, ideally

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time	**Memory time**
$$\frac{W(n)}{P}$$	$$\frac{W(n)}{P \cdot f(Z)}$$
P-fold speedup, ideally	

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time	**Memory time**
$\dfrac{W(n)}{P}$	$\dfrac{W(n)}{P \cdot f(Z)}$
P-fold speedup, ideally	**e.g.,** $\sqrt{Z}$ $\log Z$

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time	Memory time	Network time
$\dfrac{W(n)}{P}$	$\dfrac{W(n)}{P \cdot f(Z)}$	$\dfrac{W(n)}{h(n)} \cdot \dfrac{g(P)}{P}$
P-fold speedup, ideally	e.g., $\sqrt{Z}$, $\log Z$	

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time

Memory time

Network time

$$\frac{W(n)}{P}$$

$$\frac{W(n)}{P \cdot f(Z)}$$

$$\frac{W(n)}{h(n)} \cdot \frac{g(P)}{P}$$

P-fold speedup, ideally

e.g., $\sqrt{Z}$, $\log Z$

Asymptotic reduction

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time	Memory time	Network time
$\frac{W(n)}{P}$	$\frac{W(n)}{P \cdot f(Z)}$	$\frac{W(n)}{h(n)} \cdot \frac{g(P)}{P}$
P-fold speedup, ideally	e.g., $\sqrt{Z}$, $\log Z$	Asymptotic reduction ($h(n)$); Tradeoff ($\frac{g(P)}{P}$)

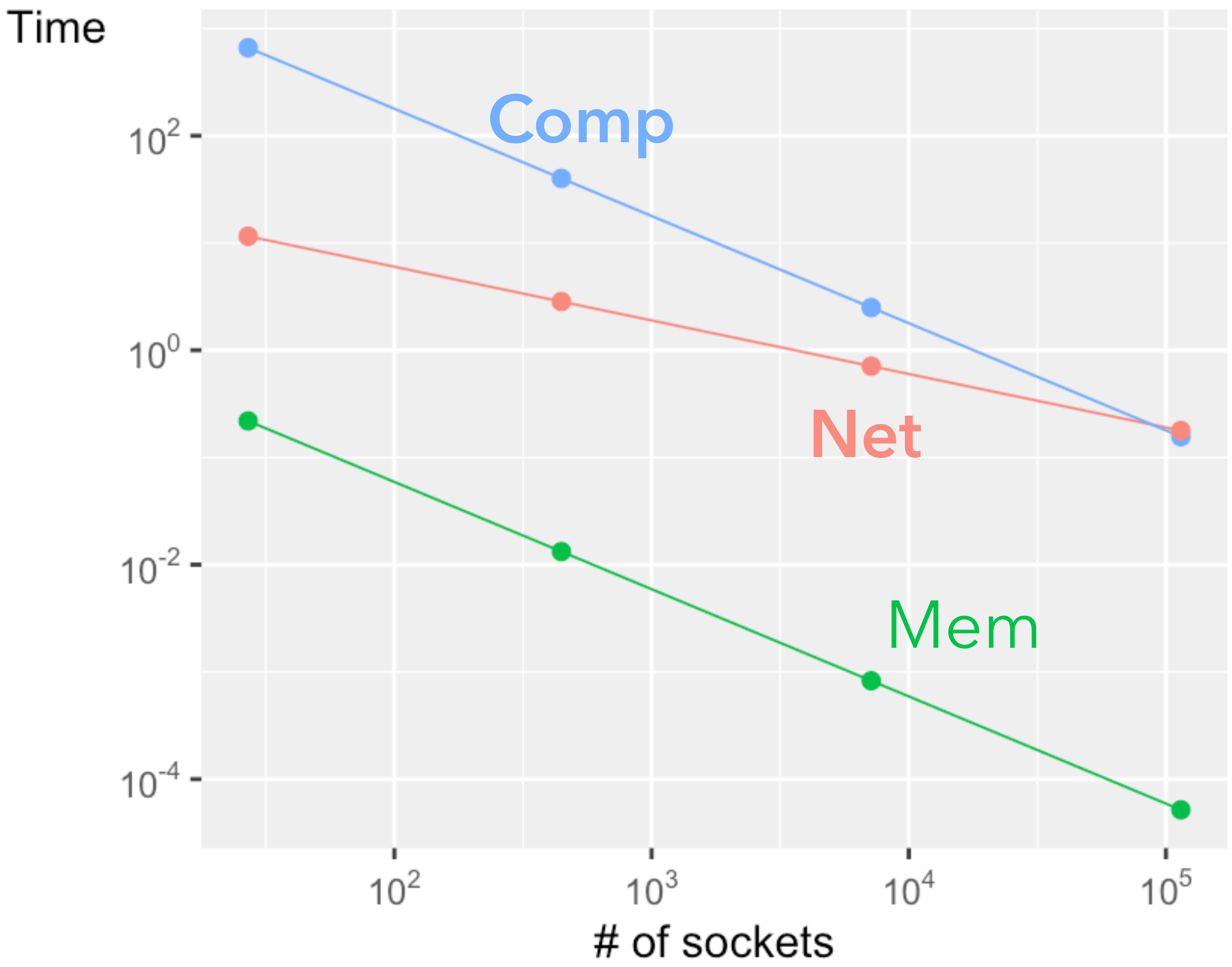
Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

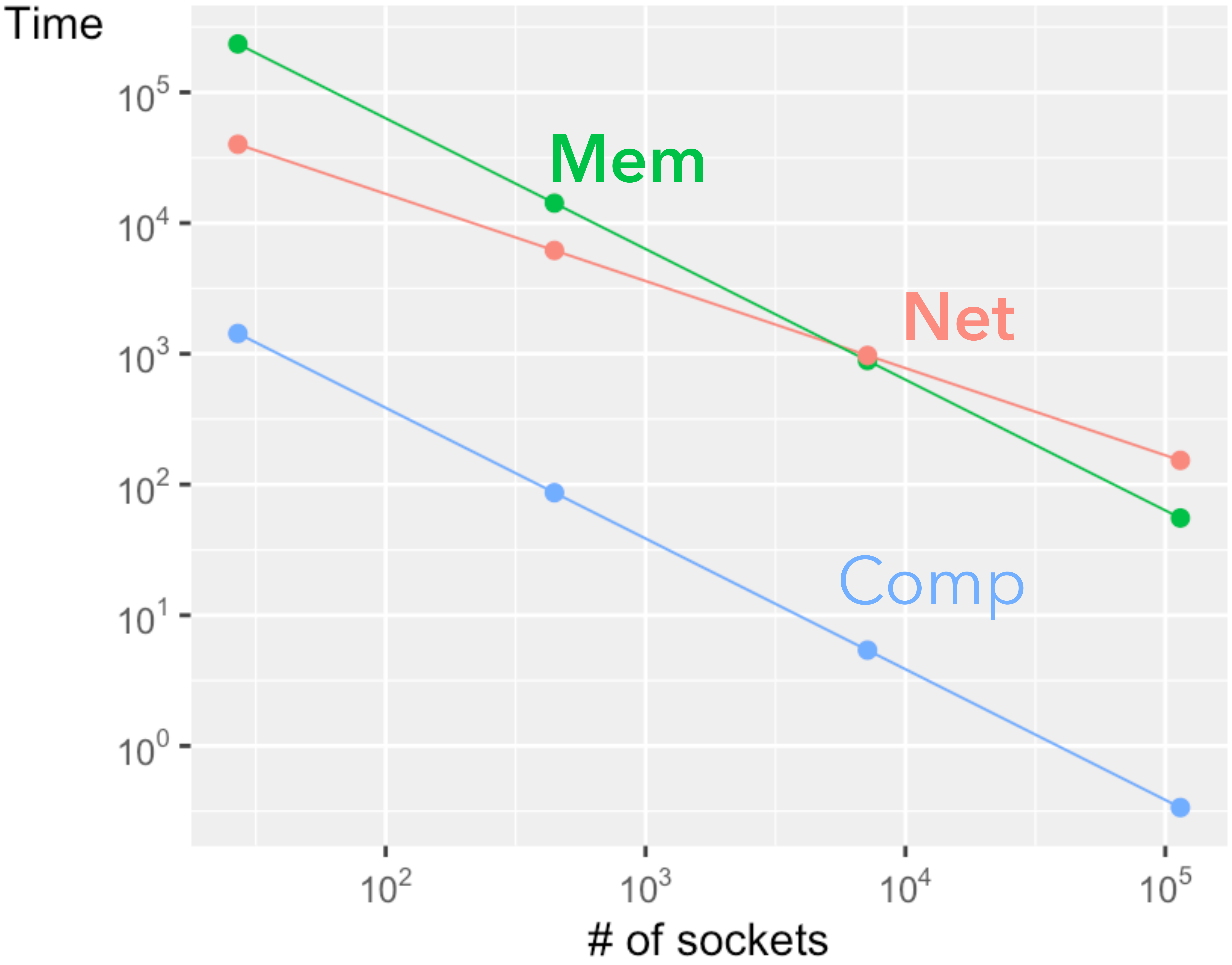
Compute time	Memory time	Network time
$\frac{W(n)}{P}$	$\frac{W(n)}{P \cdot f(Z)}$	$\frac{W(n)}{h(n)} \cdot \frac{g(P)}{P}$
P-fold speedup, ideally	e.g., $\sqrt{Z}$, $\log Z$	Asymptotic reduction ($h(n)$); Tradeoff ($\frac{g(P)}{P}$)

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

Modeled Matrix Multiply: POWER9-like
Best effective performance: 93.5 PF/s



Modeled 3-D FFT: POWER9-like sockets
Best effective performance: 324 TF/s



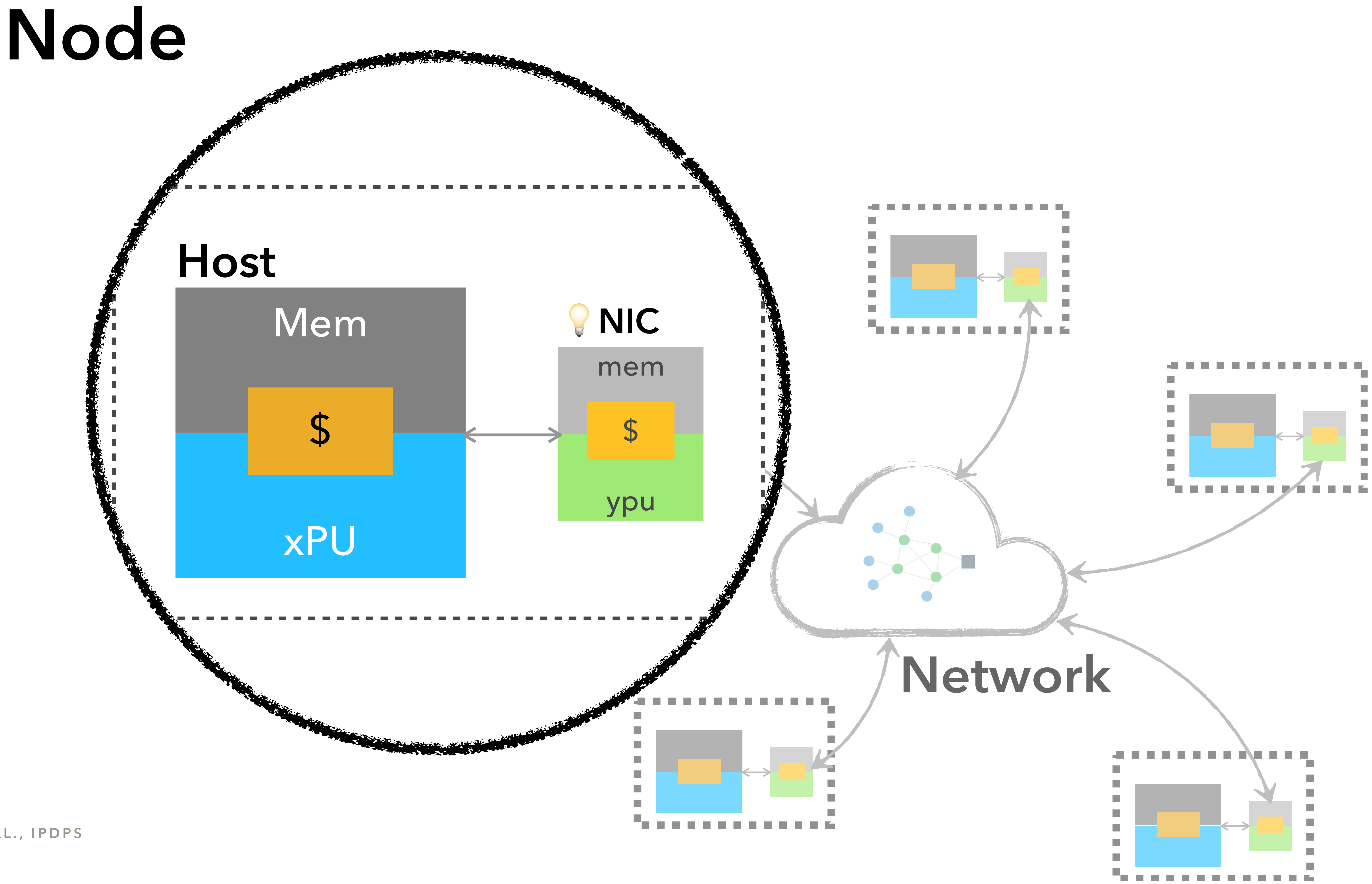
Sara Karamati

MiniMD case study

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Sara Karamati

Target platform: NVIDIA (née Mellanox) BF2

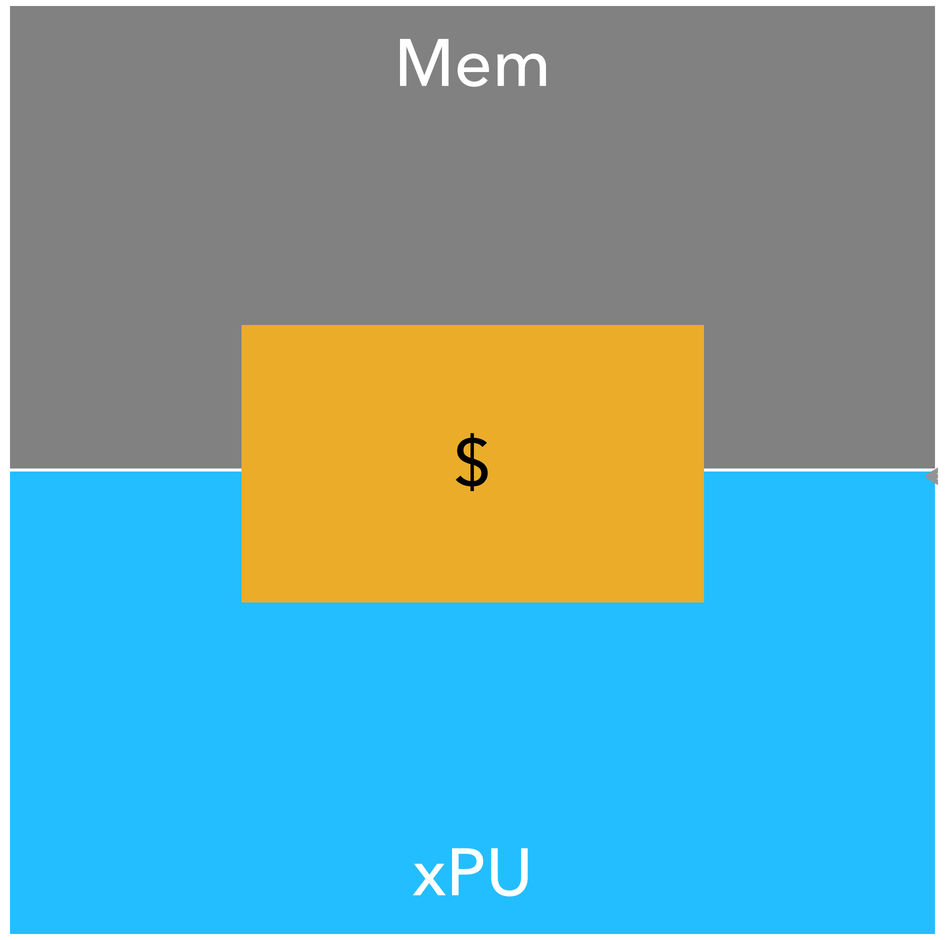


S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Sara Karamati

Target platform: NVIDIA (née Mellanox) BF2

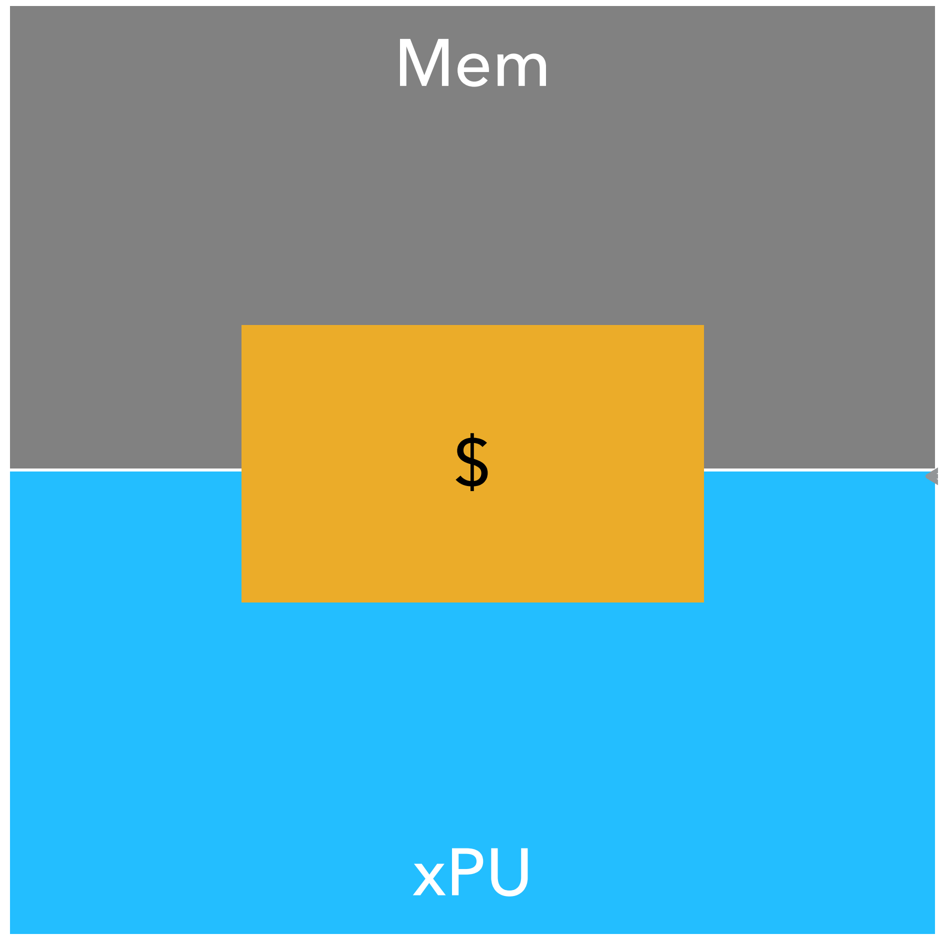
One host xPU (16 cores)



S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Target platform: NVIDIA (née Mellanox) BF2

One host xPU (16 cores)



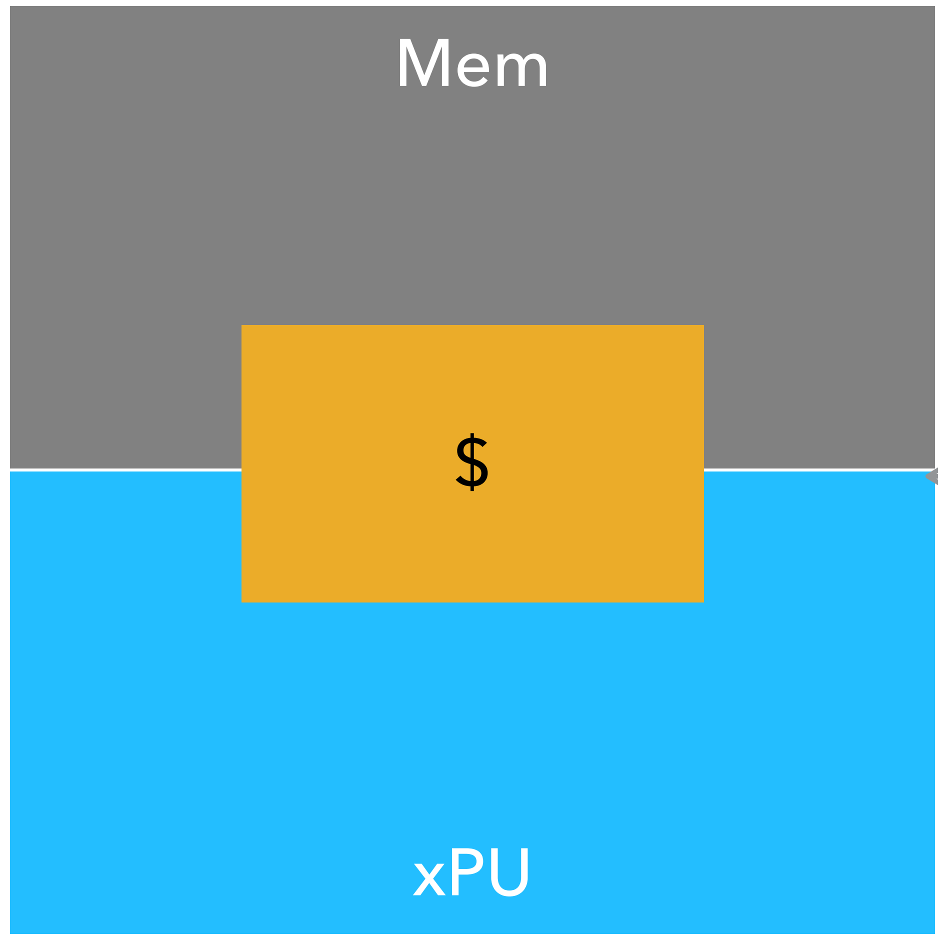
657 GF/s (fp64)

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Sara Karamati

Target platform: NVIDIA (née Mellanox) BF2

One host xPU (16 cores)



657 GF/s (fp64)

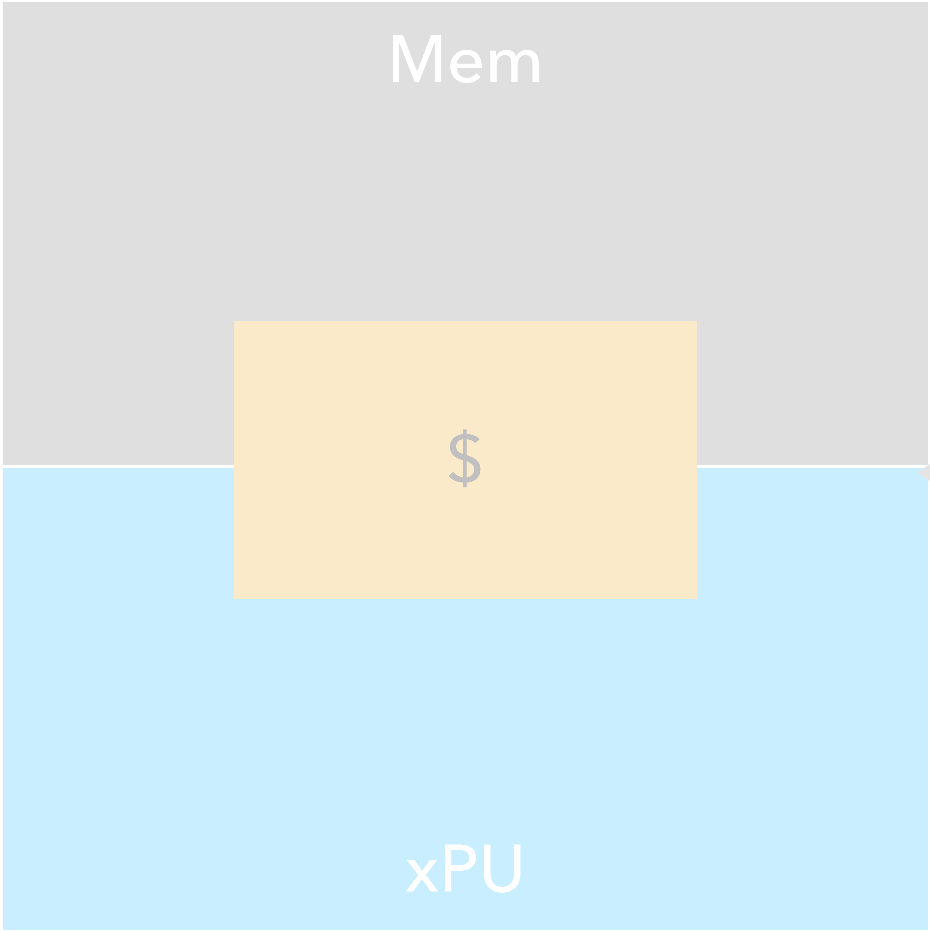
76.8 GB/s

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – “Smarter NICs for faster molecular dynamics: a case study.” doi:10.1109/IPDPS53621.2022.00063

Sara Karamati

Target platform: NVIDIA (née Mellanox) BF2

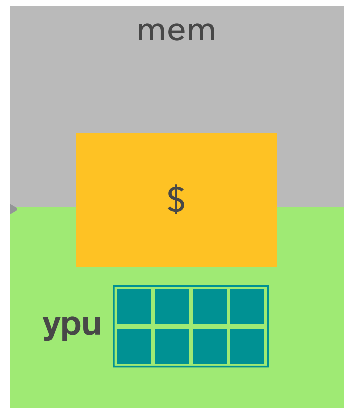
One host xPU (16 cores)



657 GF/s (fp64)

76.8 GB/s

BF-2 yPUs (no host)

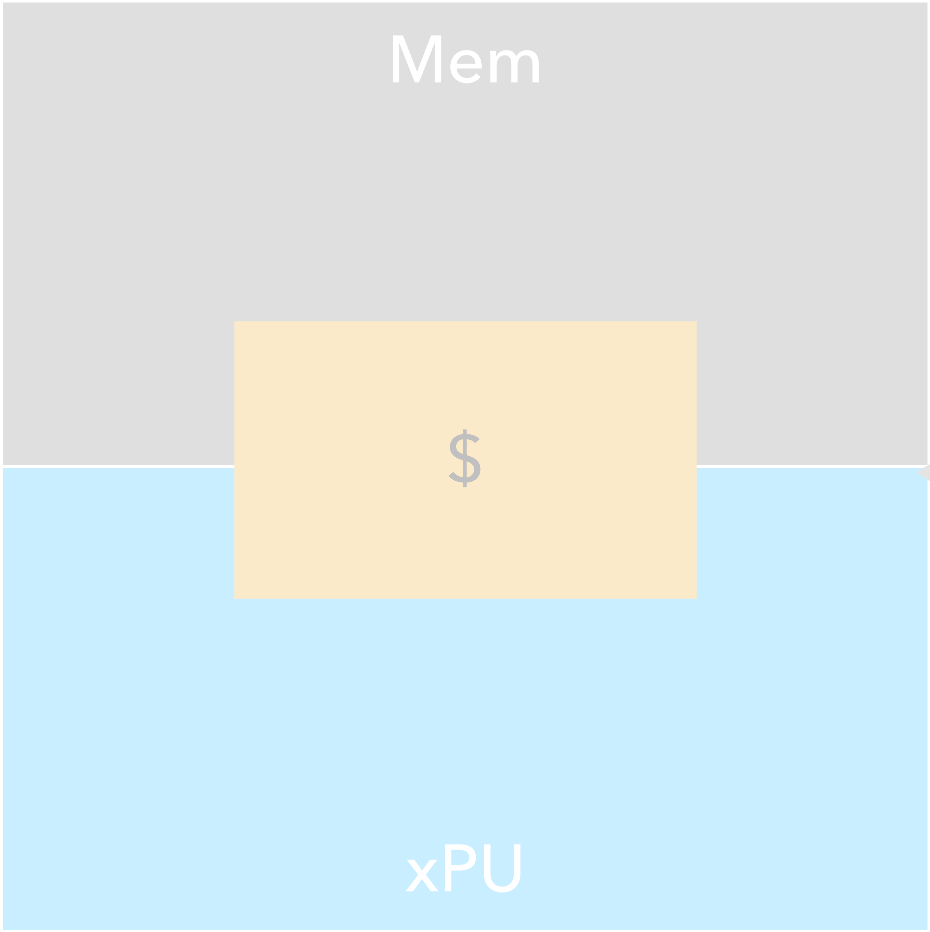


S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Sara Karamati

Target platform: NVIDIA (née Mellanox) BF2

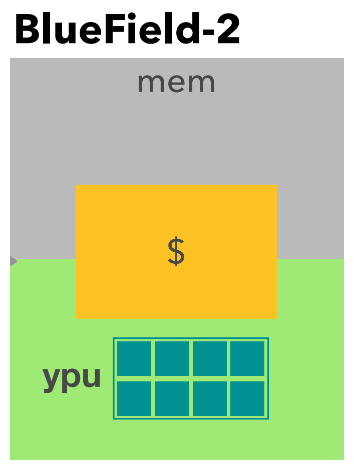
One host xPU (16 cores)



657 GF/s (fp64)

76.8 GB/s

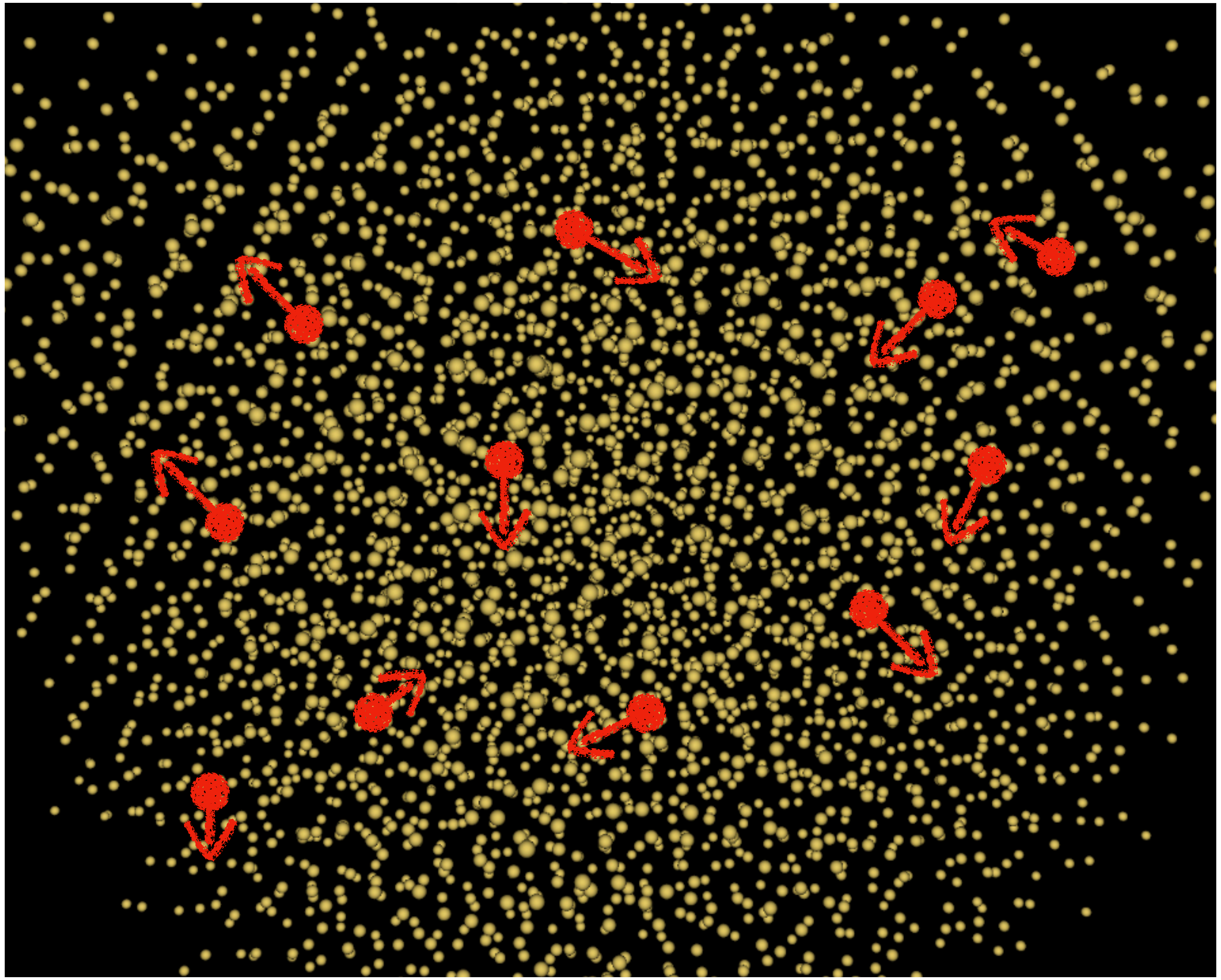
BF-2 yPUs (no host)



80 GF/s

25.6 GB/s

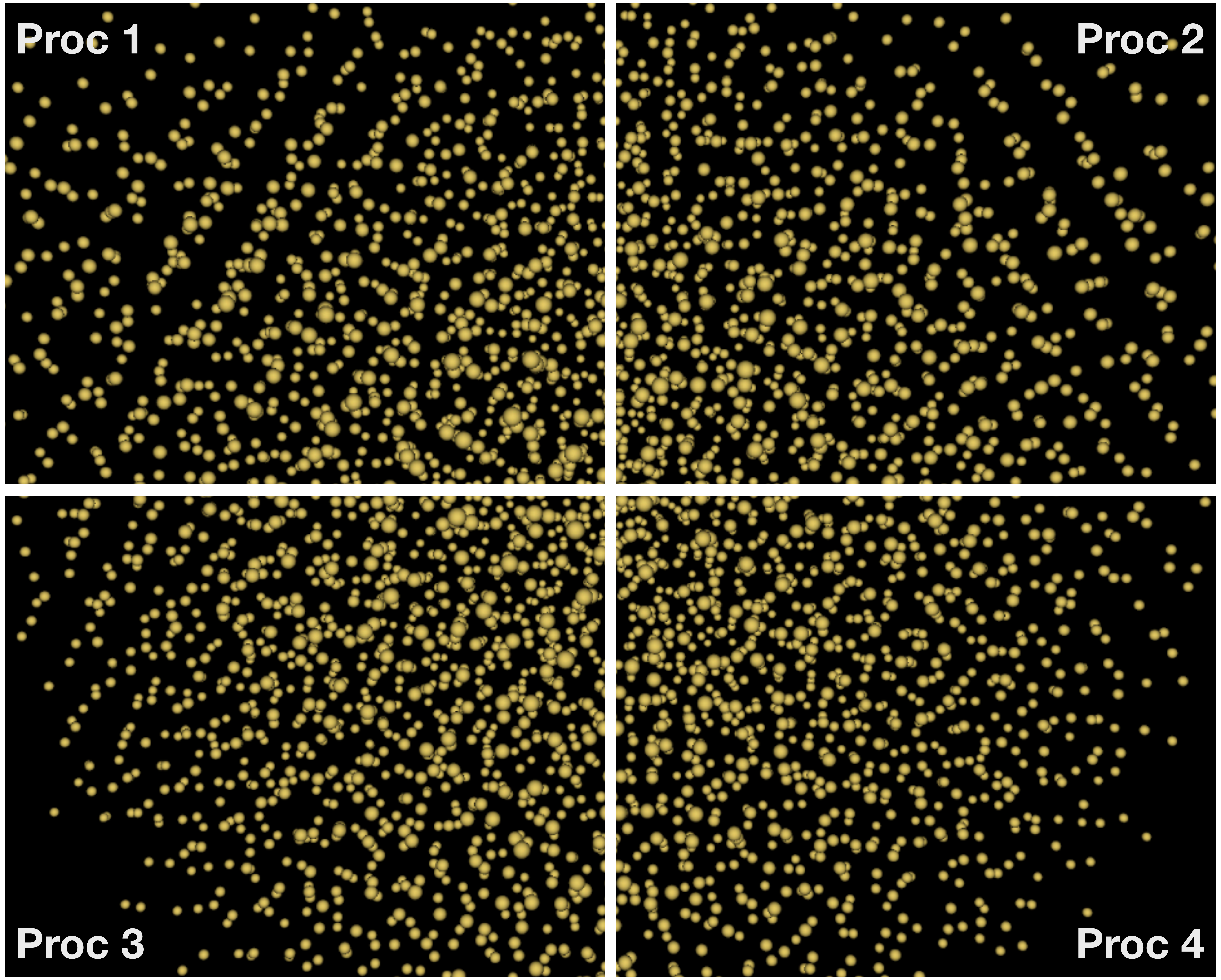
S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063



Baseline MiniMD

MiniMD is a molecular dynamics proxy-app. It calculates the position and velocity of a set of interacting particles in discrete time steps (iterations).

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063



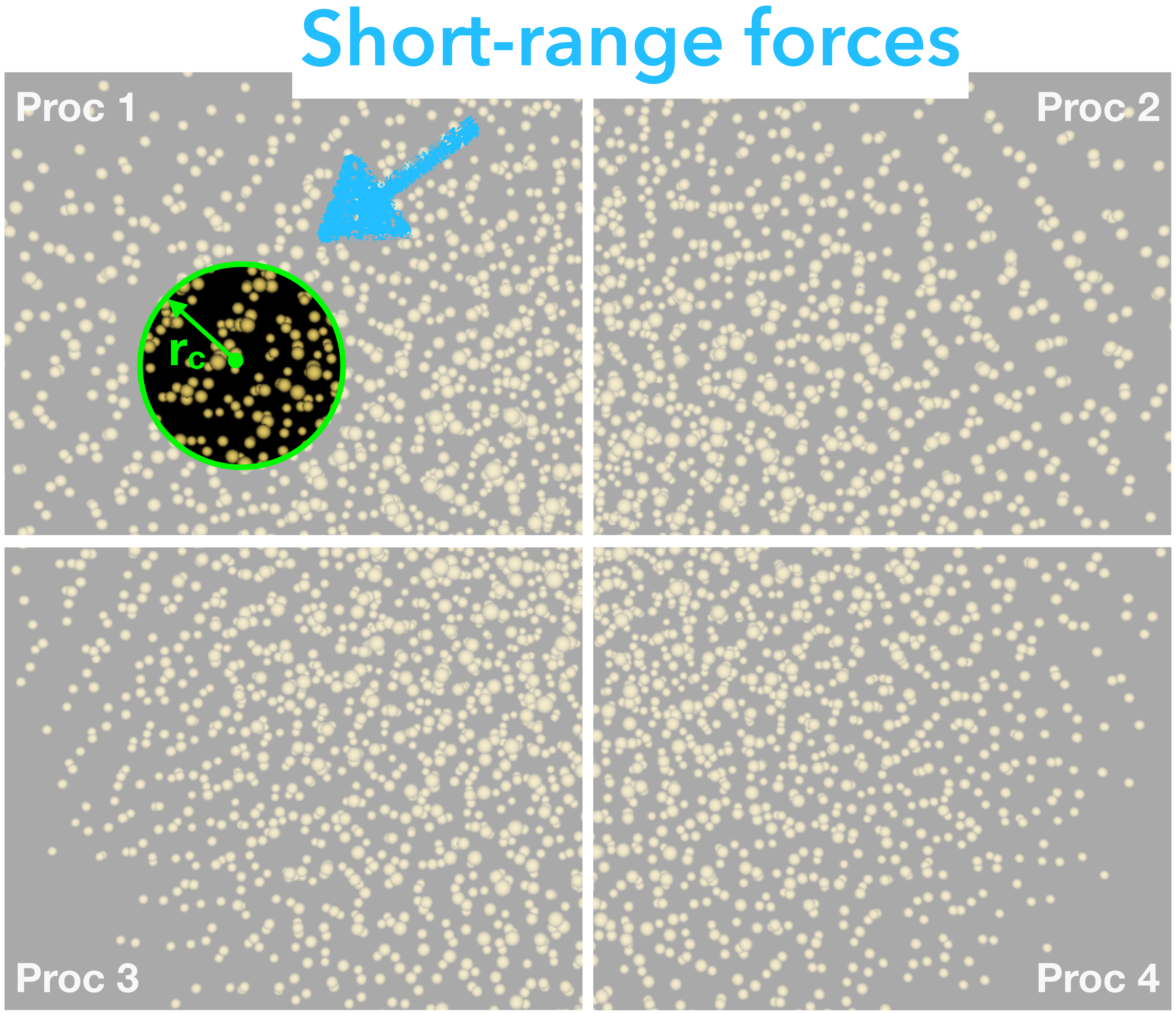
Baseline MiniMD

MiniMD is a molecular dynamics proxy-app. It calculates the position and velocity of a set of interacting particles in discrete time steps (iterations).

In the distributed-memory setting, the simulation domain is divided spatially among MPI processes.

Every process owns its particles, computes force on these particles and then updates the position and velocity of these particles.

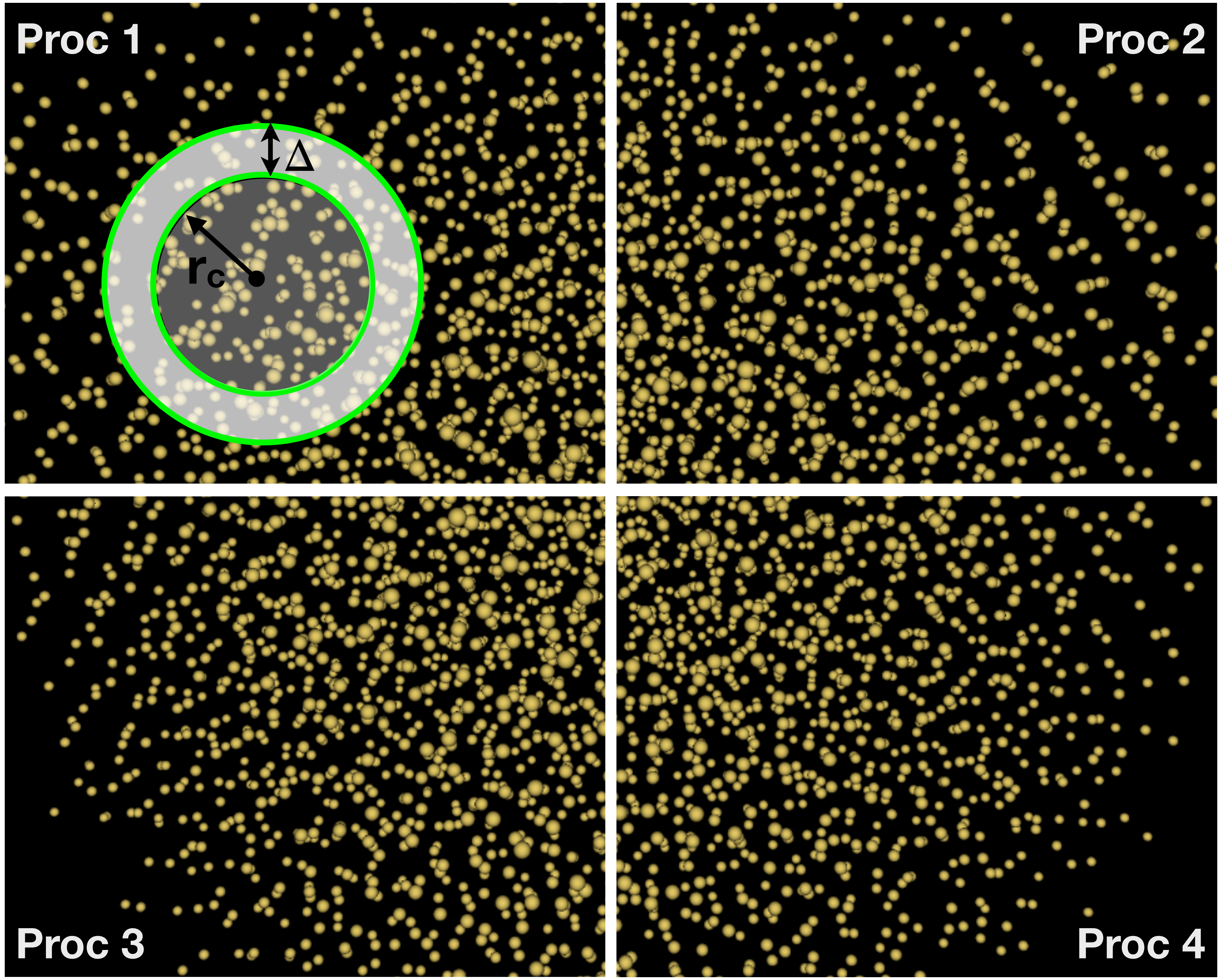
S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063



Baseline MiniMD

In each iteration, every particle interacts with others that lie within a some **cutoff distance** (r_c). A particle's **neighbor list** stores references to them.

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

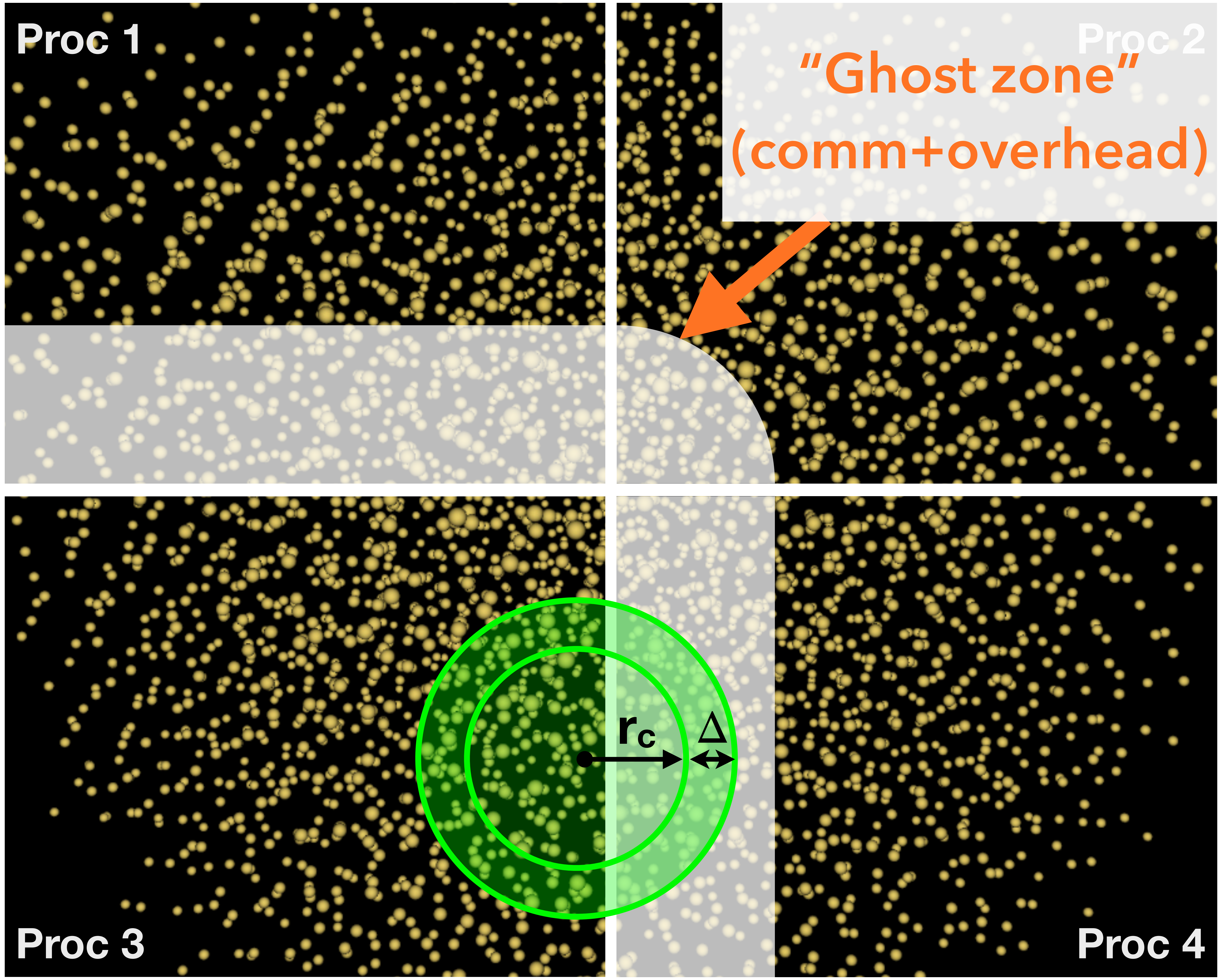


Baseline MiniMD

In each iteration, every particle interacts with others that lie within a some **cutoff distance** (r_c). A particle's **neighbor list** stores references to them.

The neighbor list must be updated as particles move. But such **updates are expensive**! So every list includes a buffer of "extra" particles that lie within a surrounding annulus, or "**skin**," parameterized by its thickness (Δ).

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

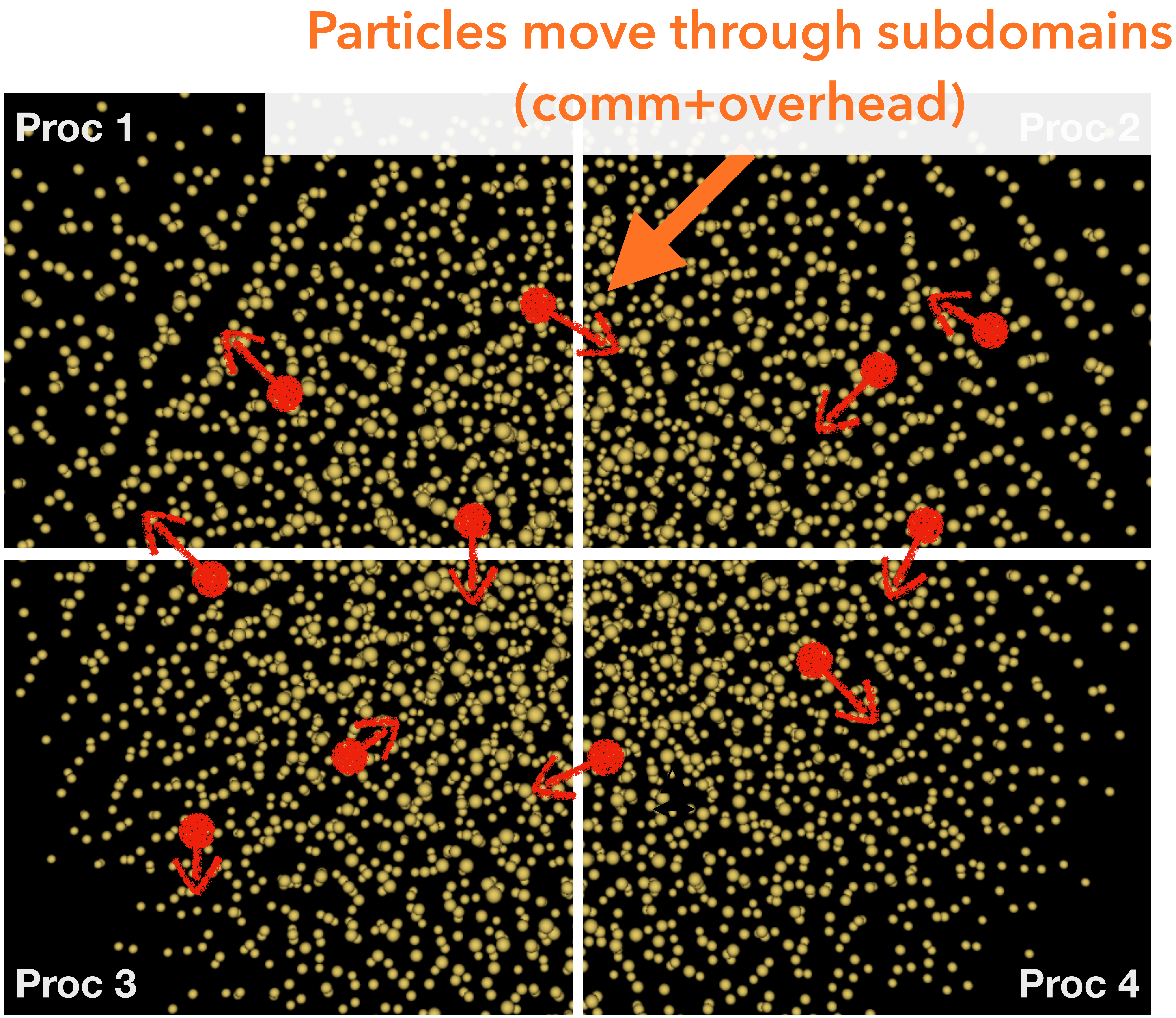


Baseline MiniMD

The cutoff distance (r_c) and skin thickness (Δ) imply the size of the interaction region just outside the boundaries of each process.

Each process keeps a copy of particles in that region.

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063



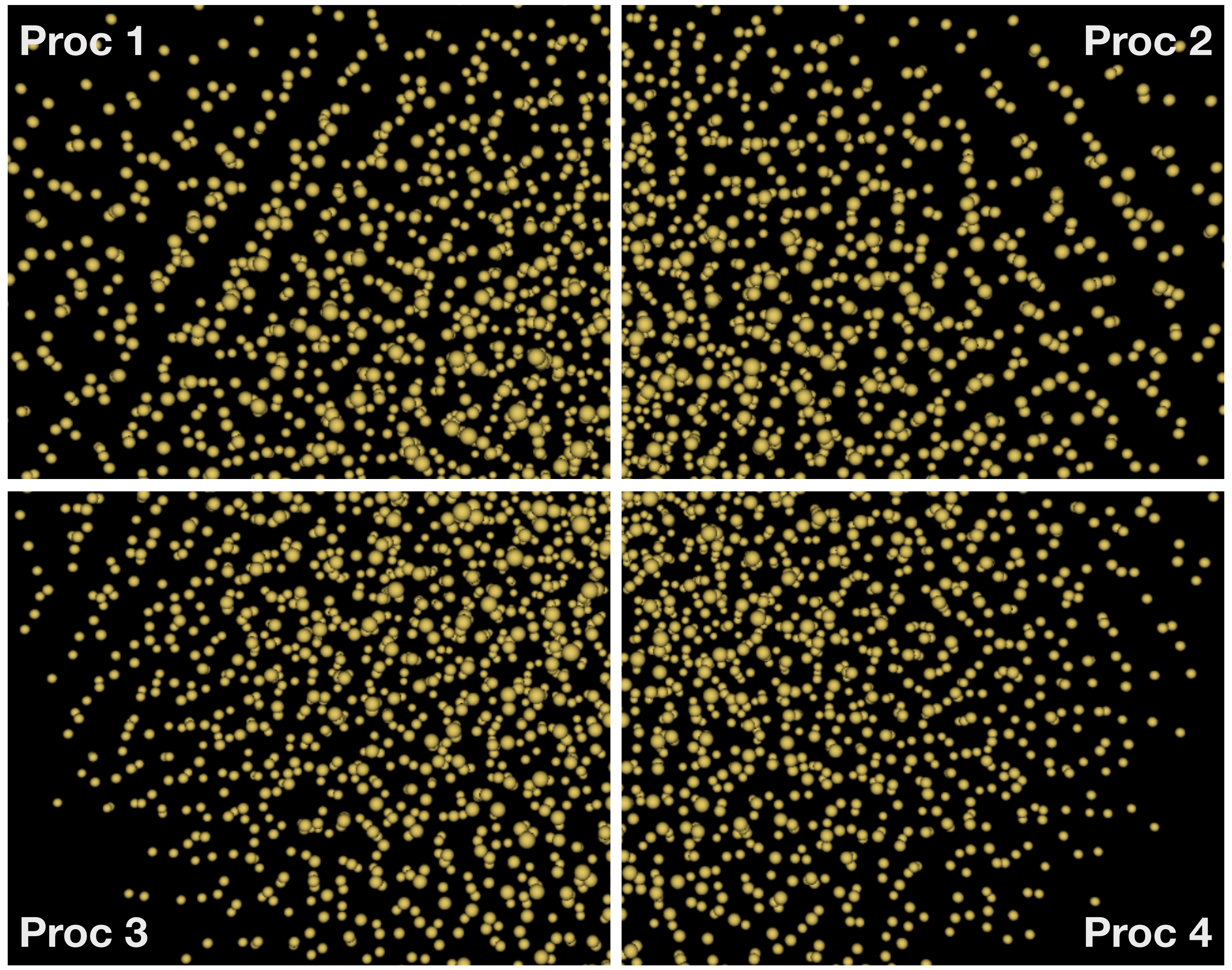
Baseline MiniMD

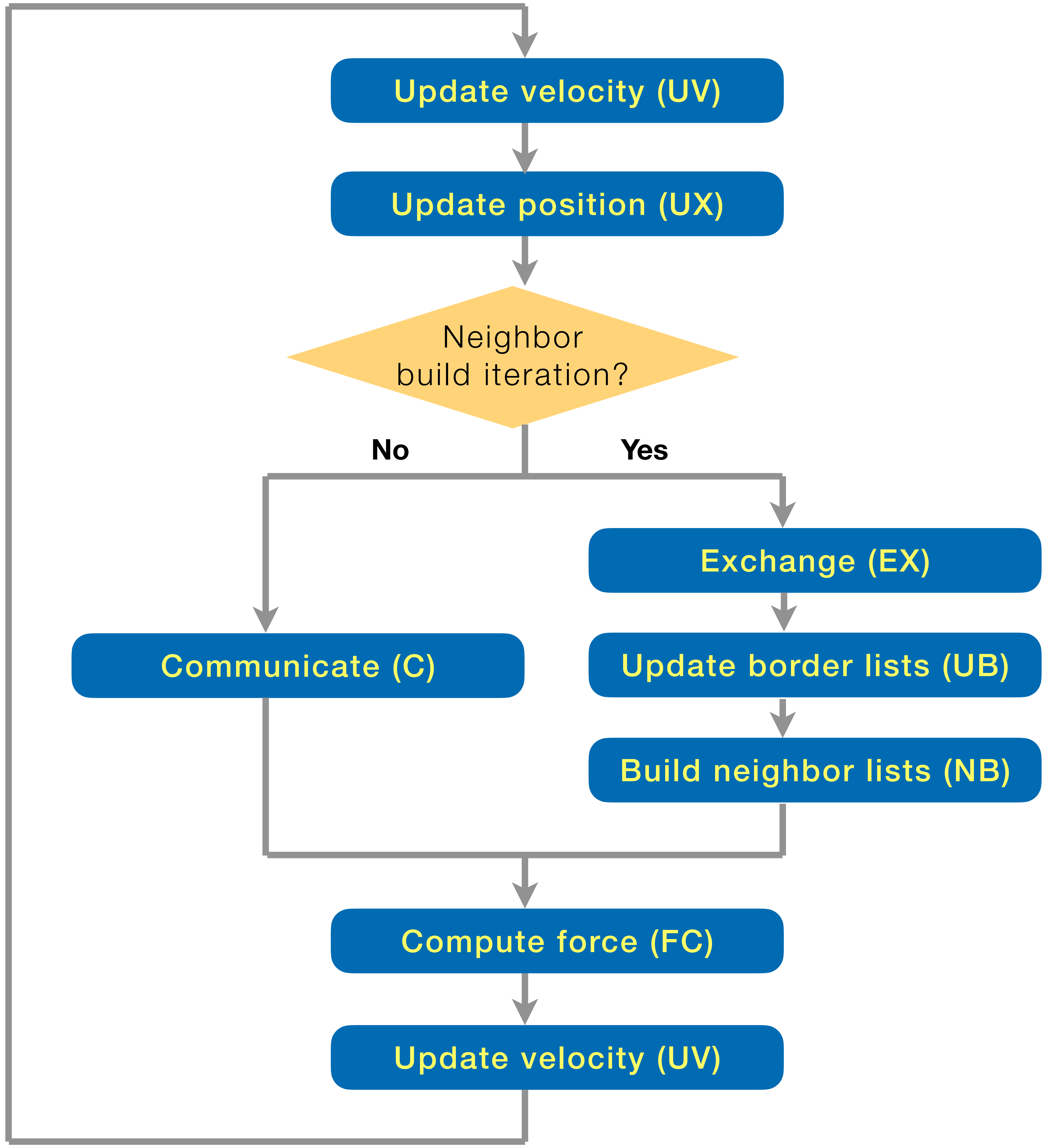
Particles are reassigned to new processes as they move through the spatial domain.

Neighbor list updates, boundary region exchanges, and particles reassignment to processes are triggered every so often via a user-selected parameter (e.g., every k iterations).

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Baseline MiniMD

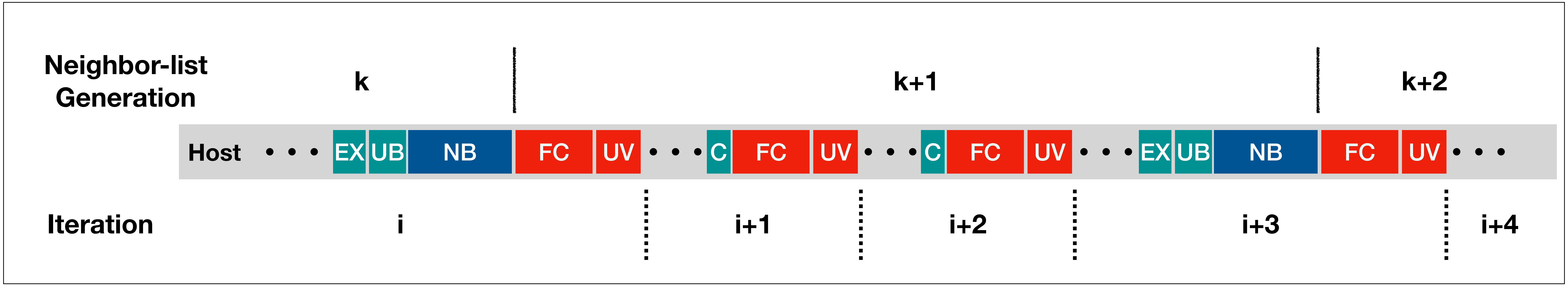




Each task is **parallelizable** but the sequence is **sequential** as shown by edges

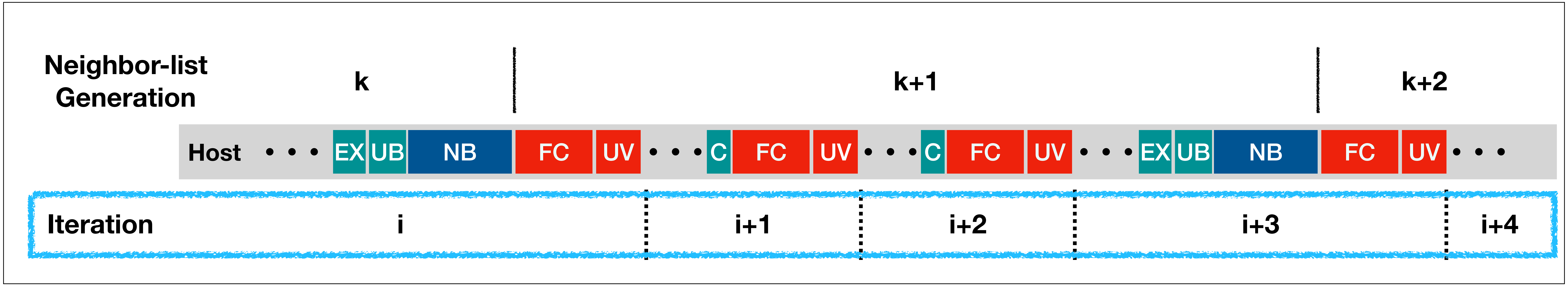
S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Serial dependencies



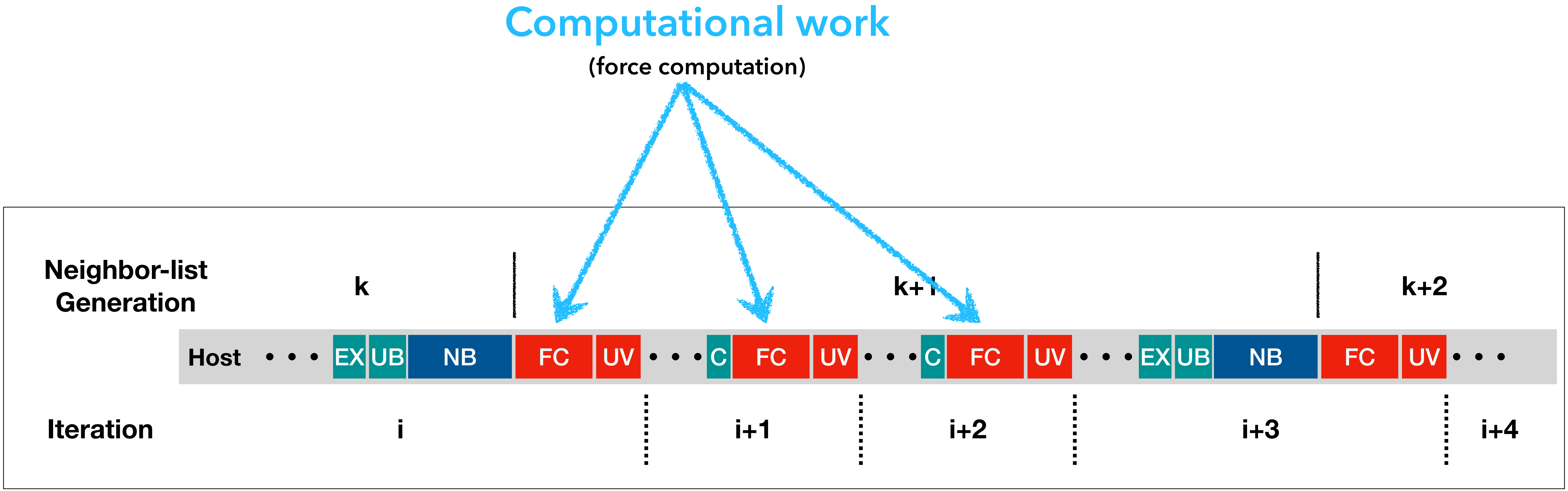
S. Karamati (GT) et al. (Sandia, Queens U.; 2022) — "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Serial dependencies



S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

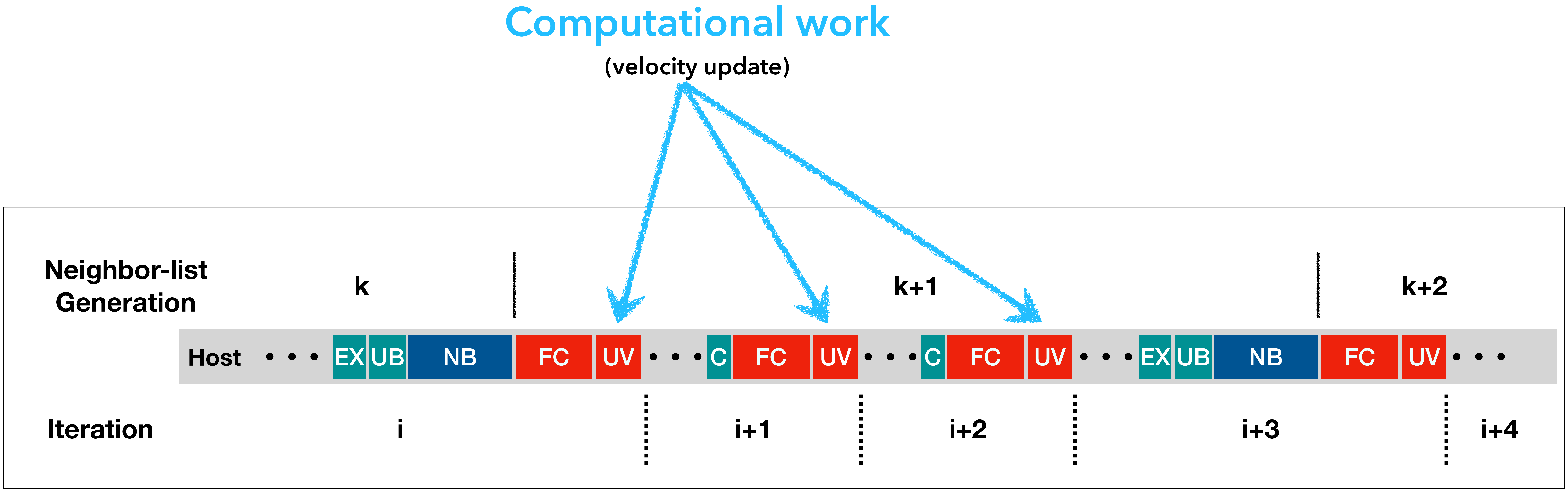
Serial dependencies



S. Karamati (GT) et al. (Sandia, Queens U.; 2022) — "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

FC work: force computation

Serial dependencies

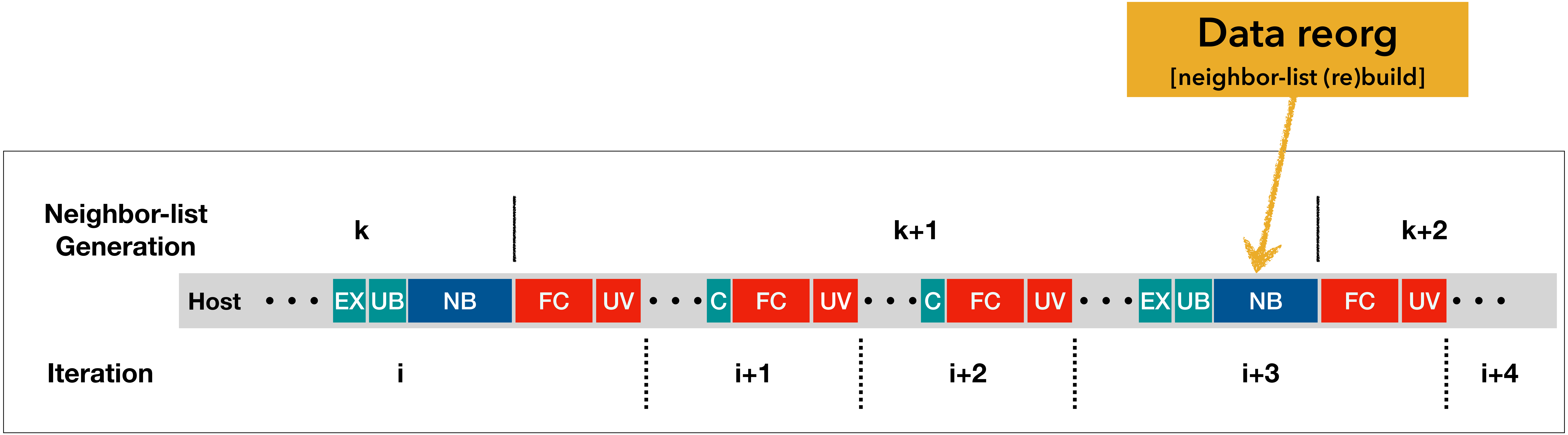


S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

FC work: force computation

UV work: update velocity

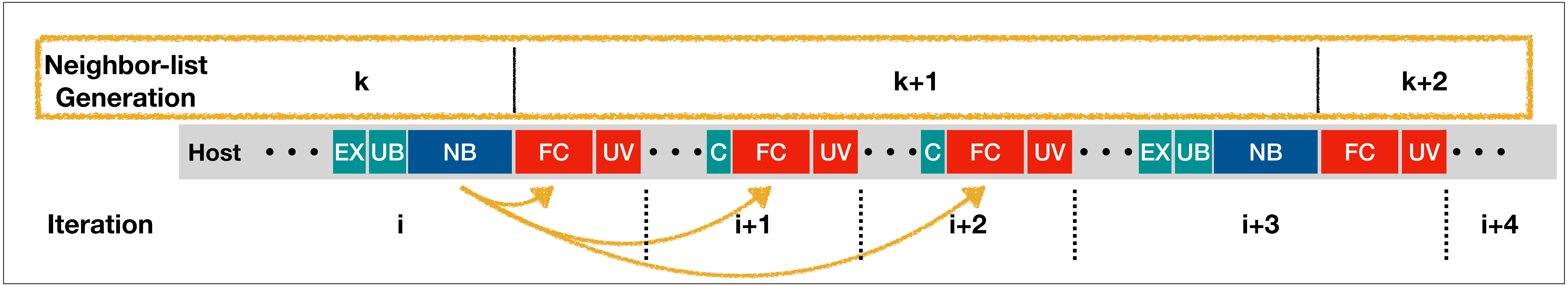
Serial dependencies



S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

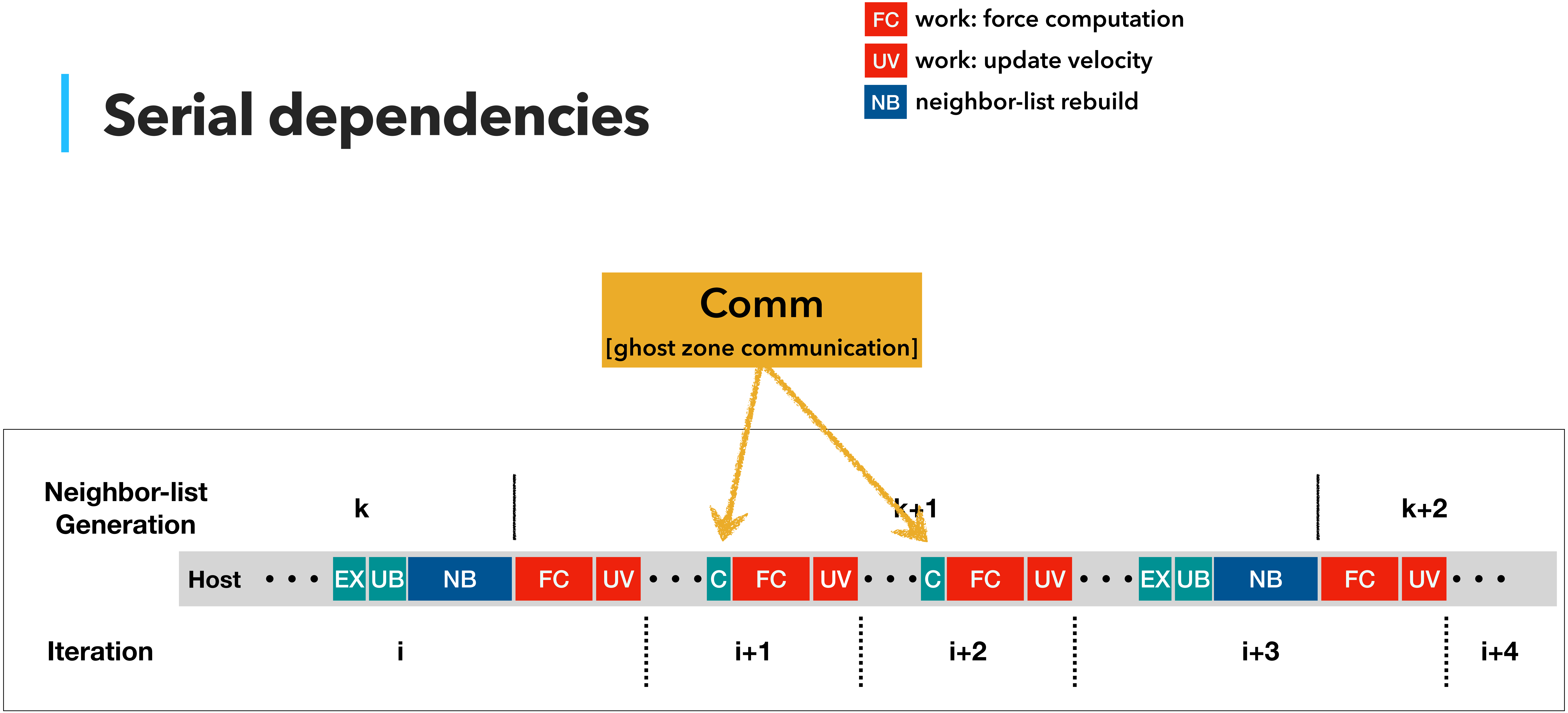
Serial dependencies

FC work: force computation
UV work: update velocity
NB neighbor-list rebuild



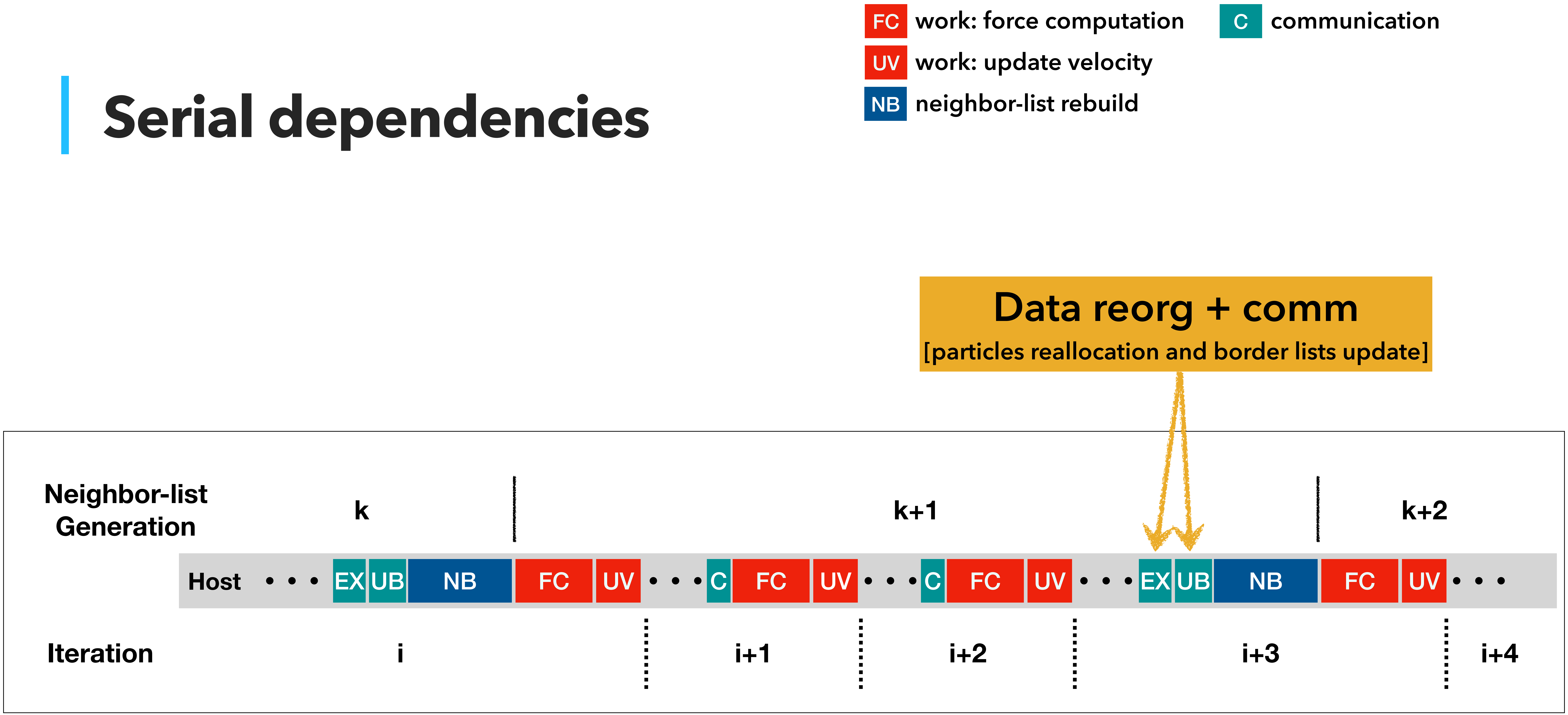
S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Serial dependencies



S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

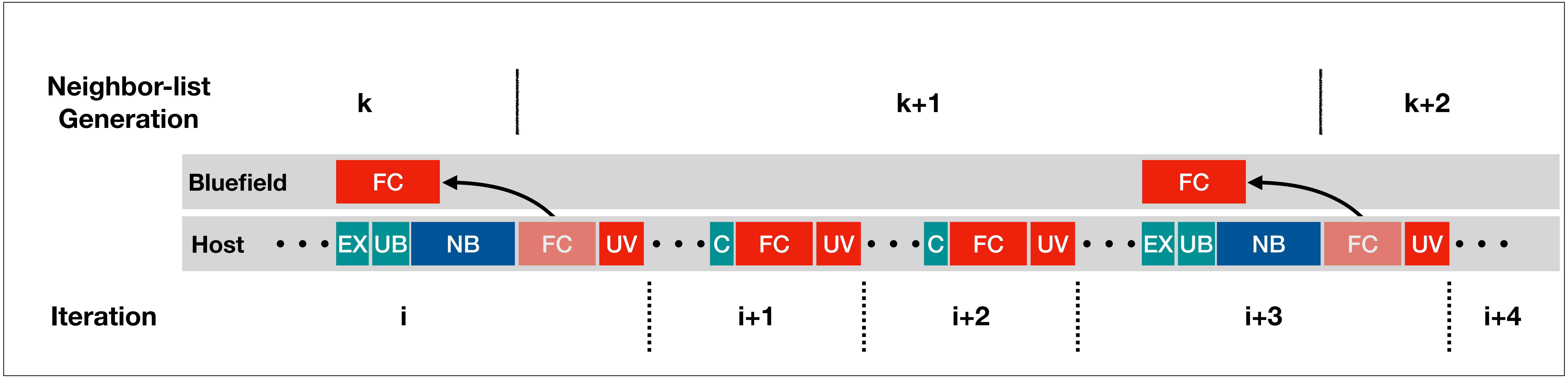
Serial dependencies



S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

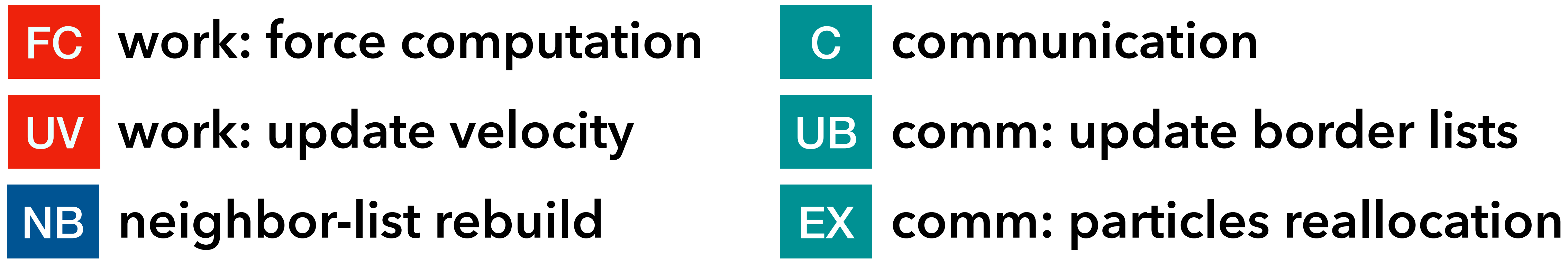
Breaking the dependencies
("Off-path" algorithm)

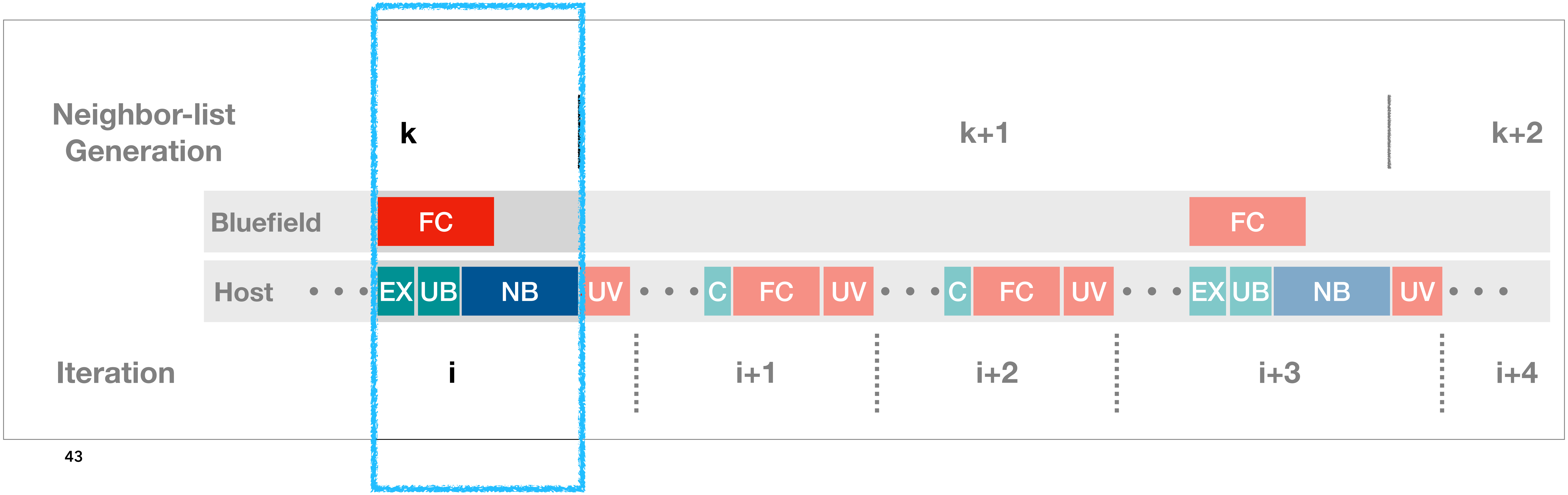
FC	work: force computation
UV	work: update velocity
NB	neighbor-list rebuild
C	communication
UB	comm: update border lists
EX	comm: particles reallocation



S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Breaking the dependencies
("Off-path" algorithm)

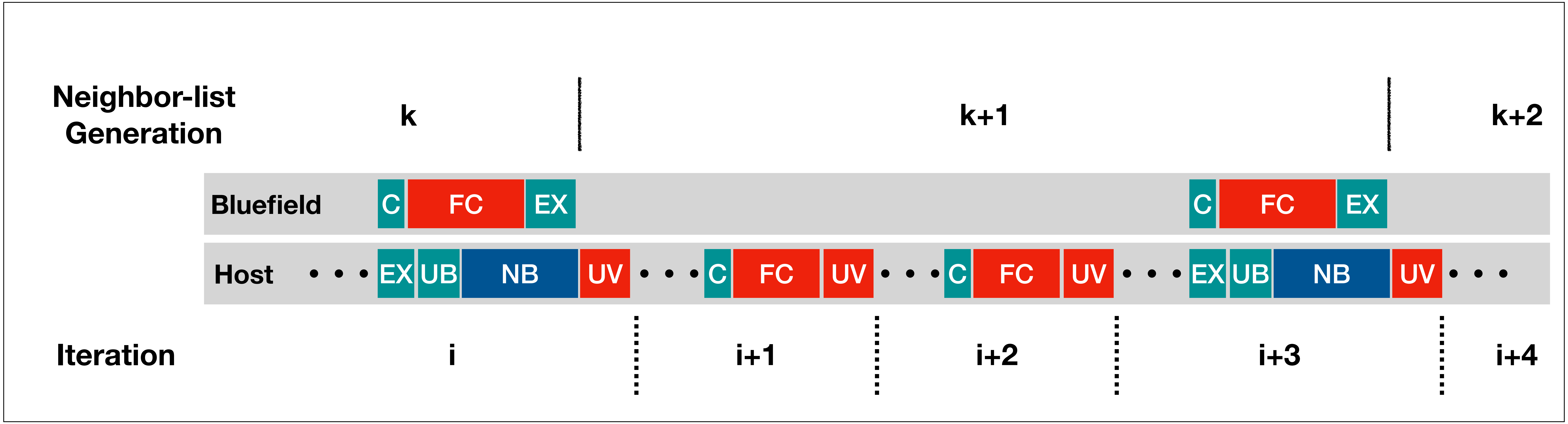




S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Breaking the dependencies
("Off-path" algorithm)

- FC work: force computation
- UV work: update velocity
- NB neighbor-list rebuild
- C communication
- UB comm: update border lists
- EX comm: particles reallocation



x

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

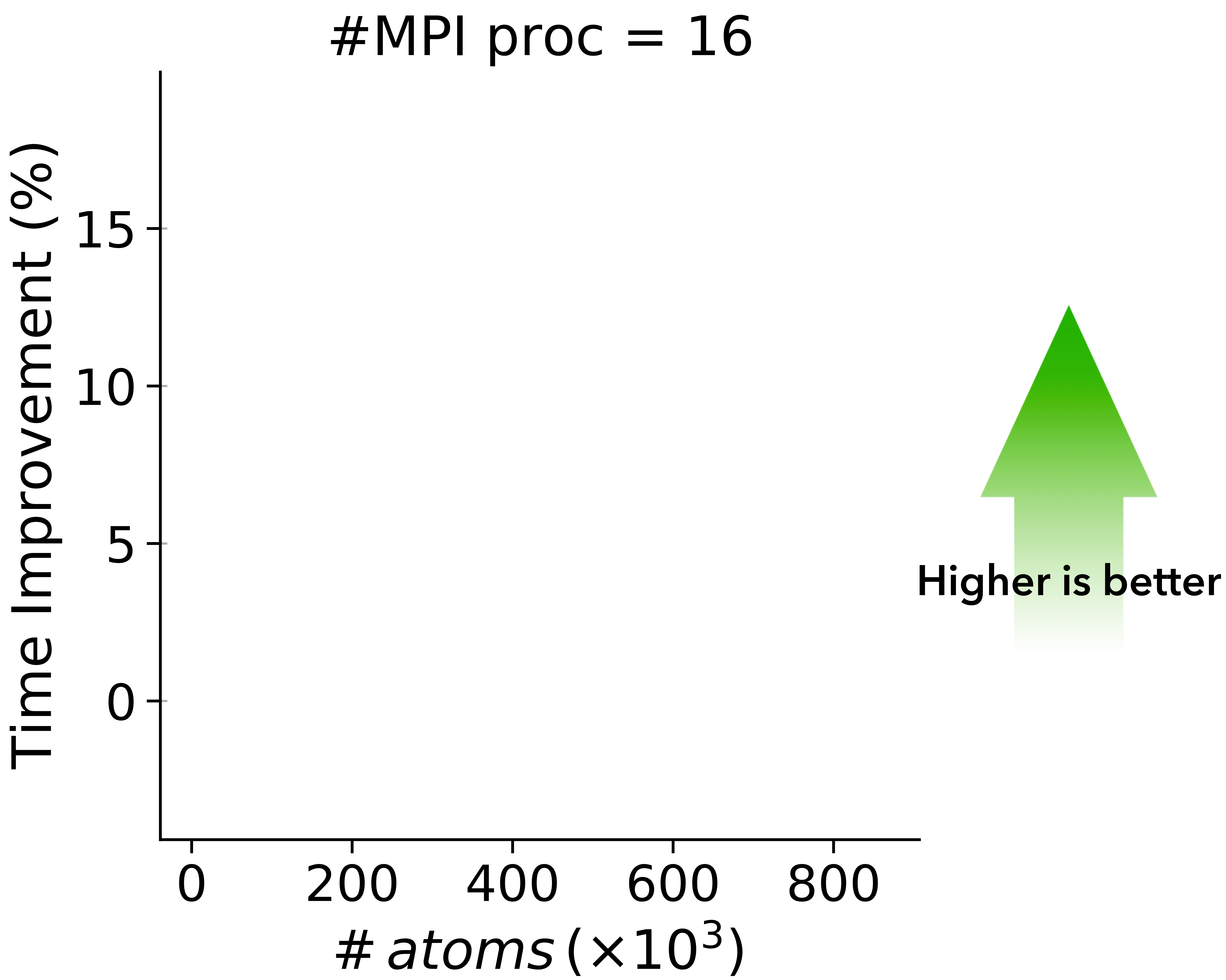
Baseline experiments

"THOR" CLUSTER, MAINTAINED BY THE HPC·AI ADVISORY COUNCIL [LINK]

- **System**: 16 nodes, Infiniband HDR (100 Gbps)
- **Hosts**: (2-socket) x (16-core Intel Broadwell E5-2697A, 2.6 GHz) + (256 GiB DDR4 RAM, 2400 MHz)
- **NICs per node**
  - 1 x NVIDIA **ConnectX-6 HDR100** (100 Gbps) InfiniBand/VPI adapters
  - 1 x NVIDIA **BlueField-2 SoC** – (8-core ARMv8 A72, 2.5 GHz) + (16 GiB DDR4 RAM) + (HDR100)

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Restructured method …

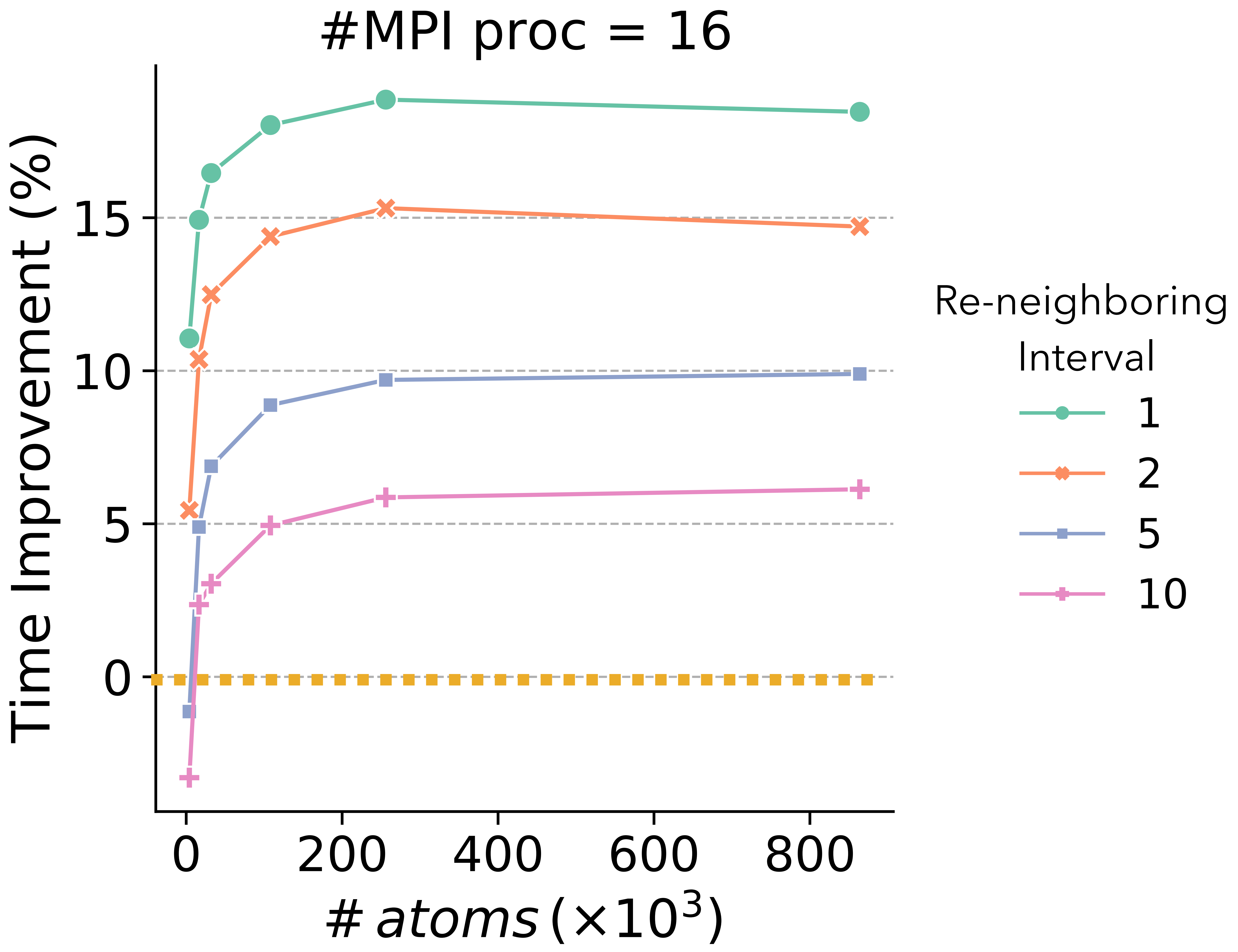


S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Restructured method is faster

We observe small, but largely uniform, **speedups of up to 20%** compared to host-only execution with conventional NICs.

This improvement compares favorably with the **power increase** on each node due to BF2, which we estimate from sensors to be **as little as 6%**.



S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

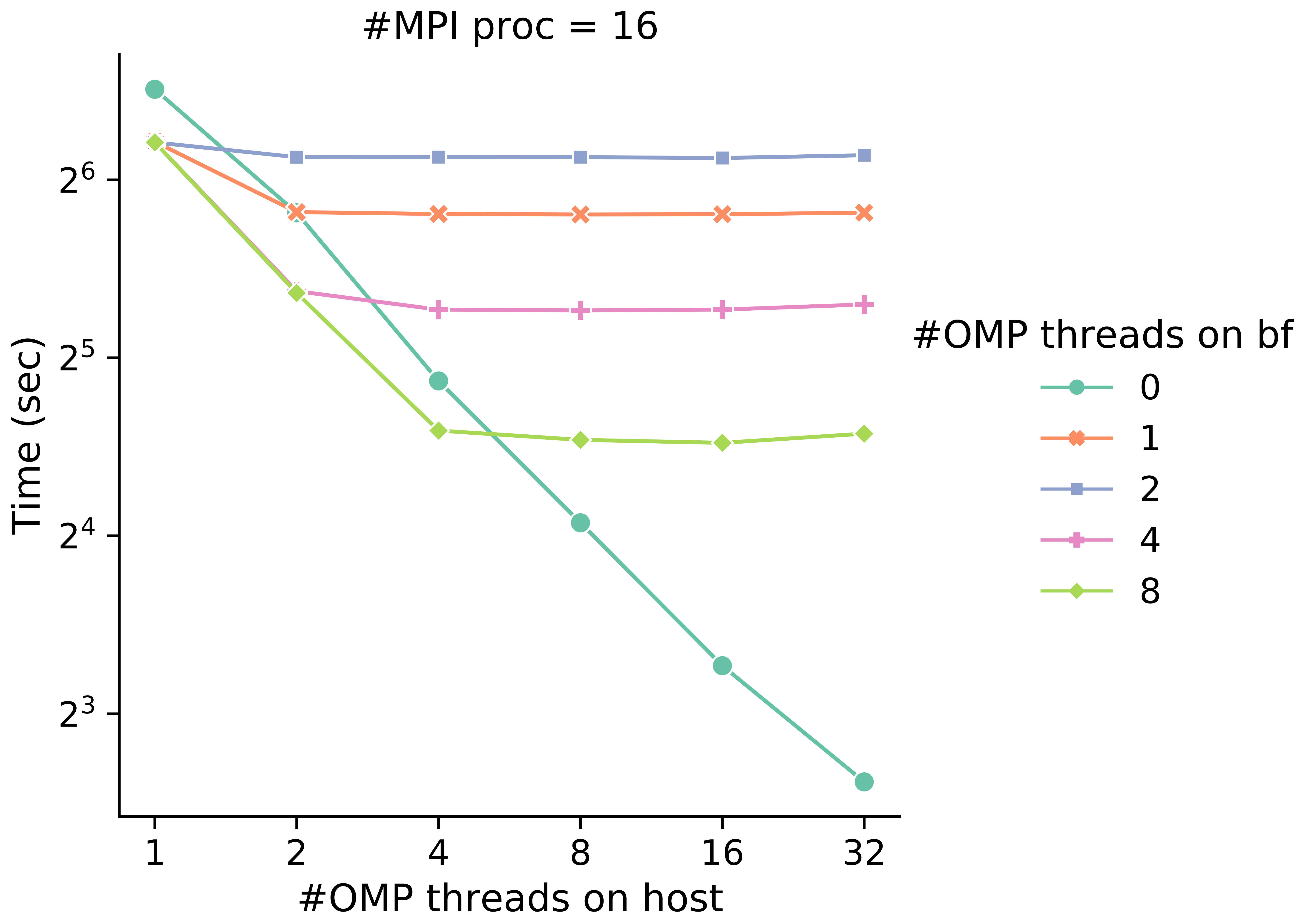
Hybrid MPI/OpenMP performance results

Our algorithm works best when it can completely hide the force computation time on BlueField.

The degree of achievable overlap depends on the relative computational power of the host and BlueField.

The knee of each curve indicates where the running times of neighbor-build on the host and force-compute on the BlueField are closest.

Thread synchronization overhead in the force computation routine causes the performance not to scale proportionally to the number of threads.



x

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

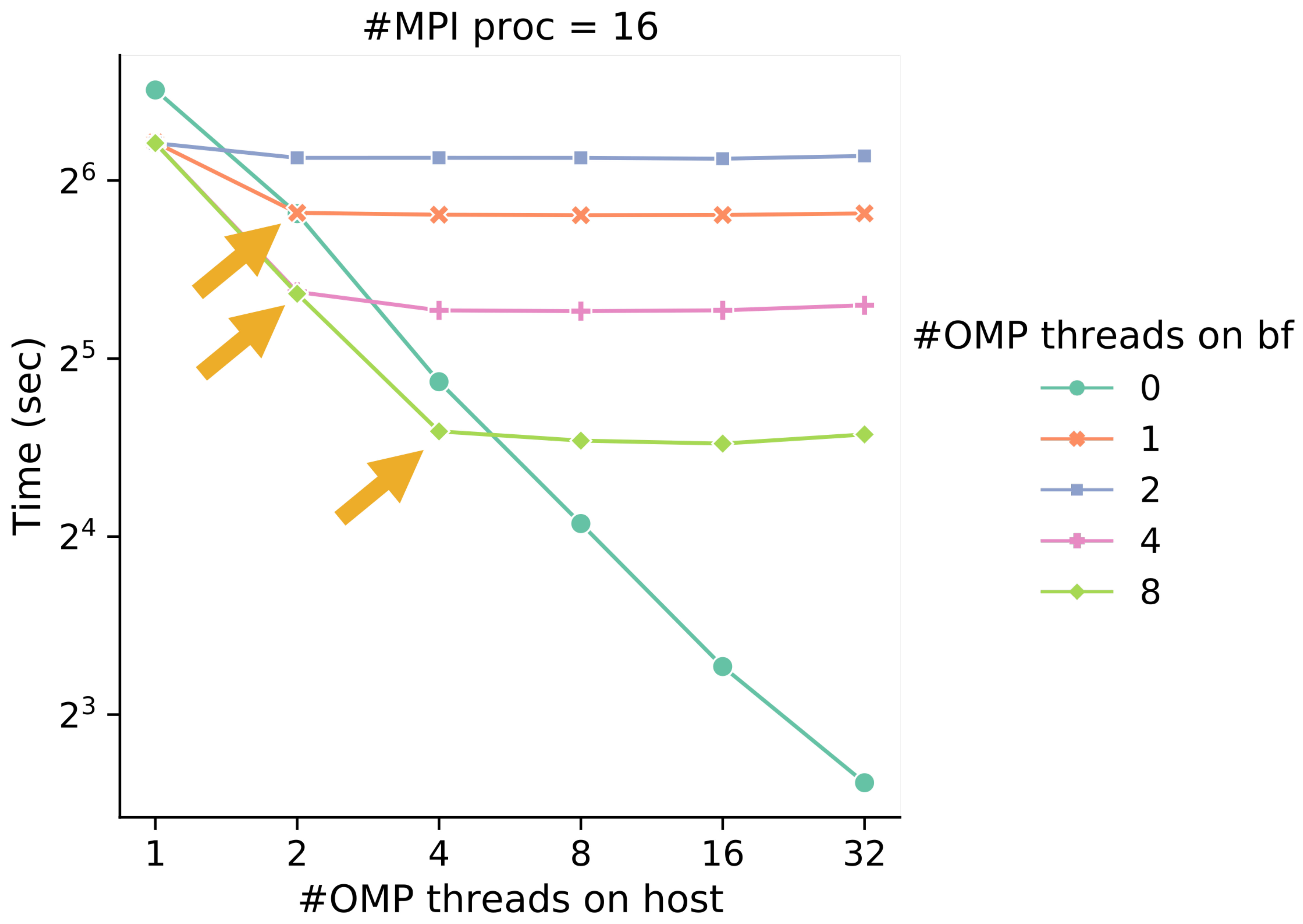
Hybrid MPI/OpenMP performance results

Our algorithm works best when it can completely hide the force computation time on BlueField.

The degree of achievable overlap depends on the relative computational power of the host and BlueField.

The **knee of each curve** indicates where the running times of neighbor-build on the host and force-compute on the BlueField are closest.

Thread synchronization overhead in the force computation routine causes the performance not to scale proportionally to the number of threads.



S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

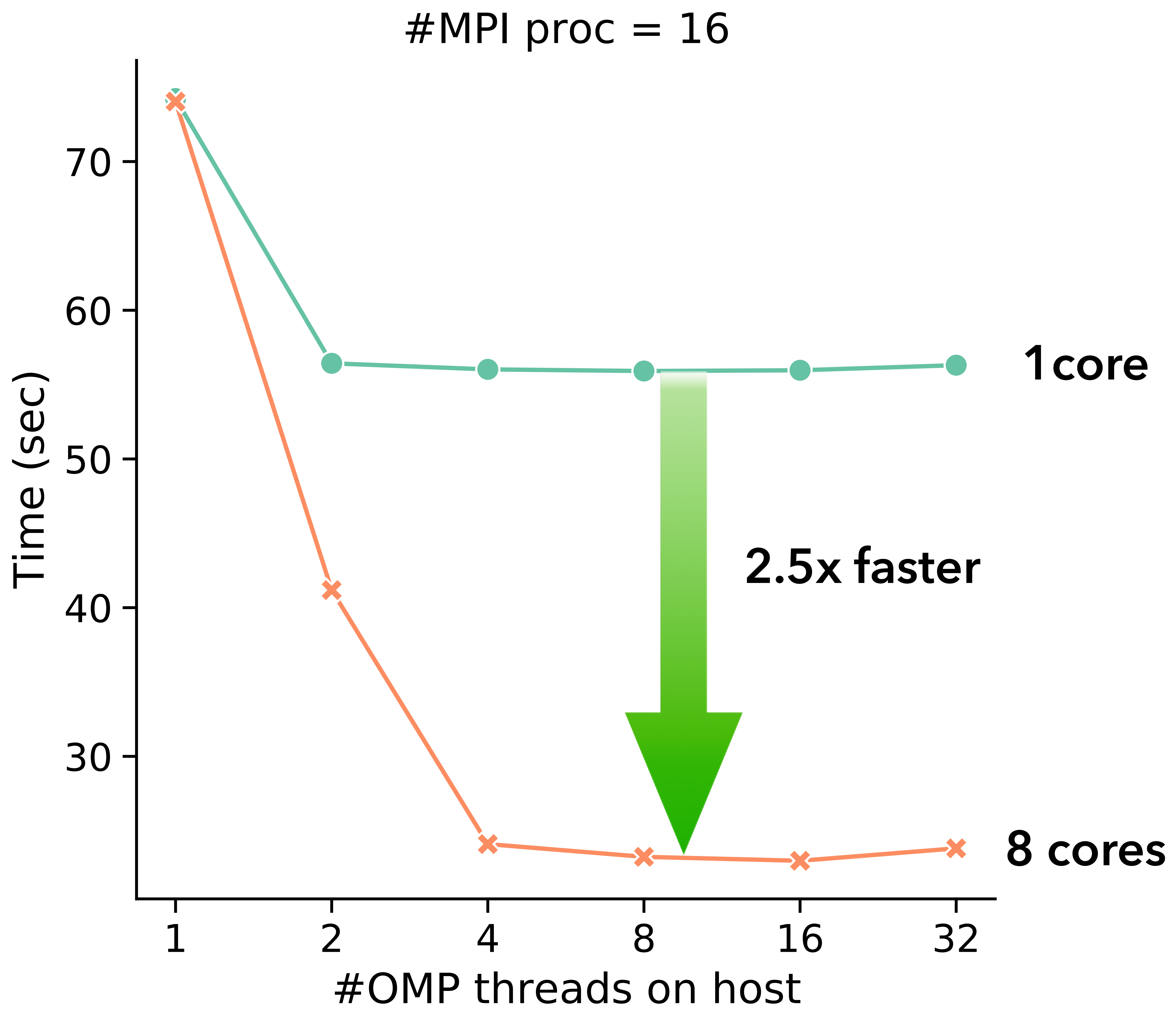
Hybrid MPI/OpenMP performance results

Our algorithm works best when it can completely hide the force computation time on BlueField.

The degree of achievable overlap depends on the relative computational power of the host and BlueField.

The knee of each curve indicates where the running times of neighbor-build on the host and force-compute on the BlueField are closest.

Thread synchronization overhead in the force computation routine causes the performance not to scale proportionally to the number of threads.

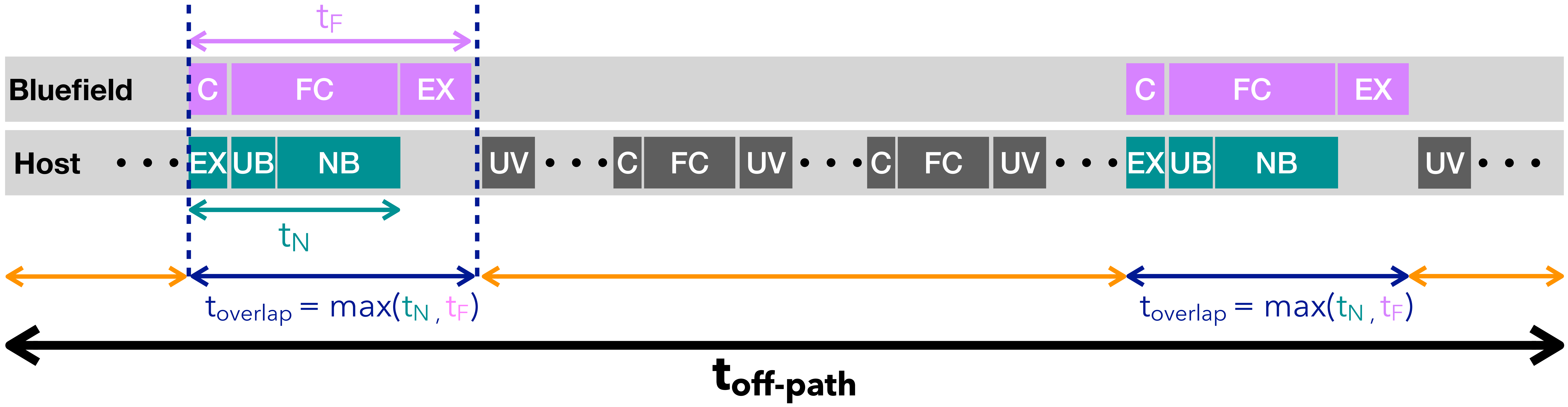


S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

An explanatory performance model

(See paper for deets)

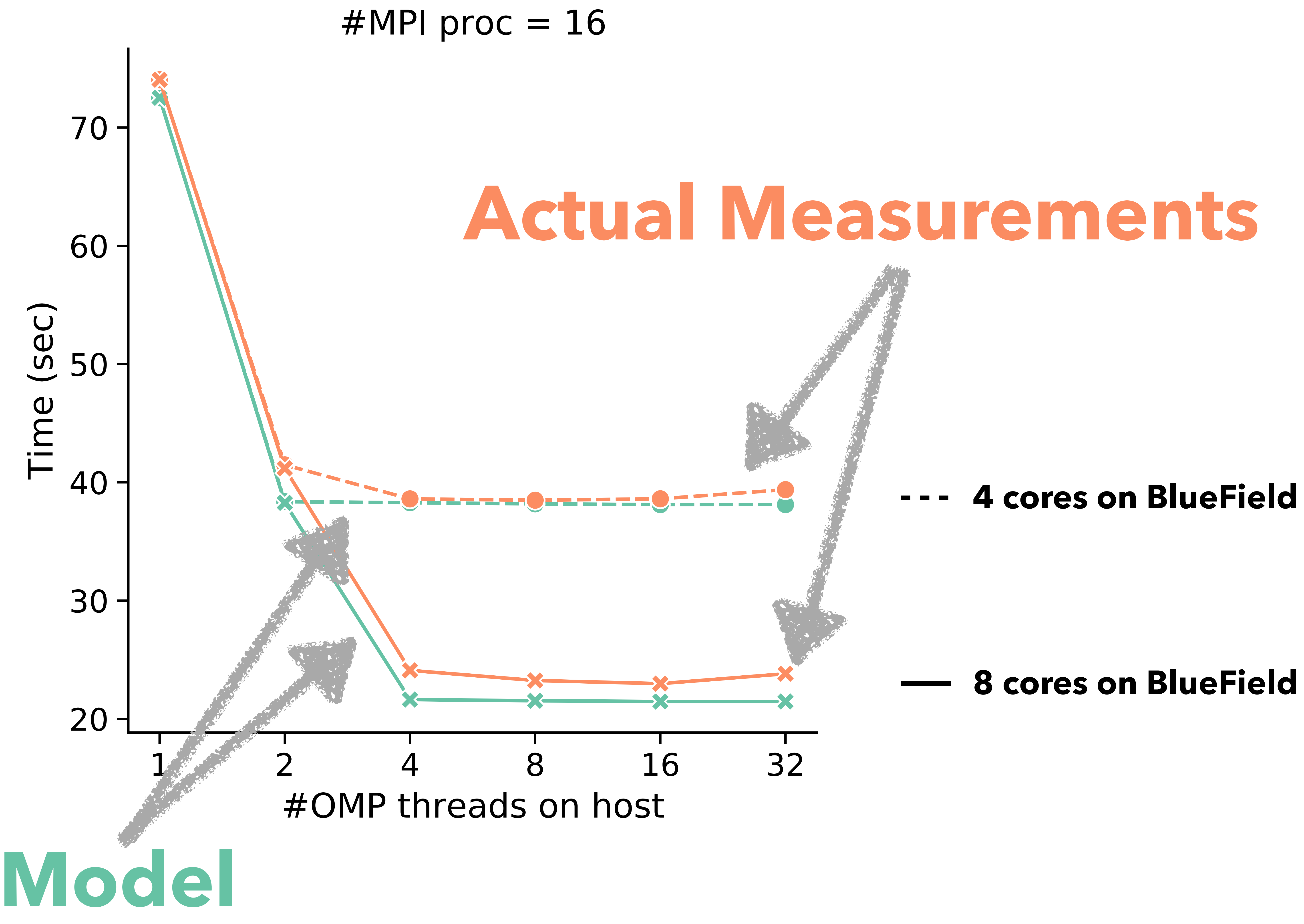
$$t_{\text{off-path}} = t_{\text{remain}} + \max(t_N, t_F) \times \#\text{re-neighboring}$$



S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Predictive power of our performance model

The model can closely predict the algorithm runtime.



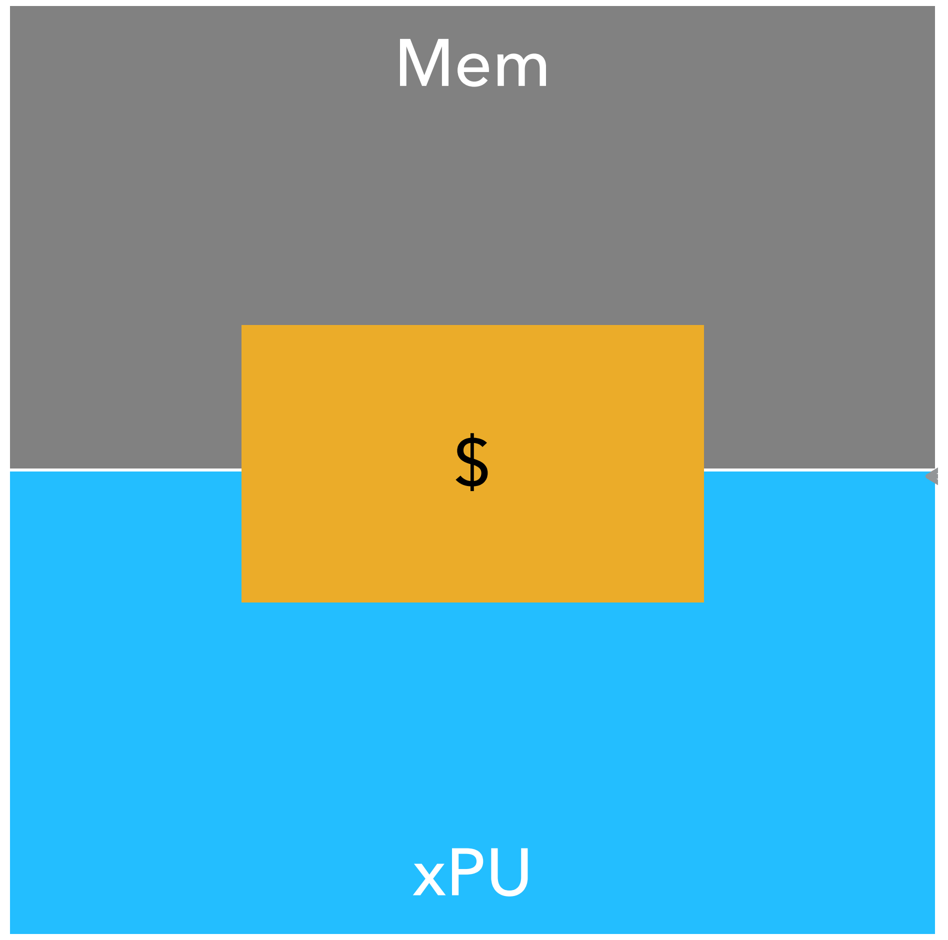
x

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Hypothetical: Multi-SmartNIC

a.k.a., revisiting the “iron law”

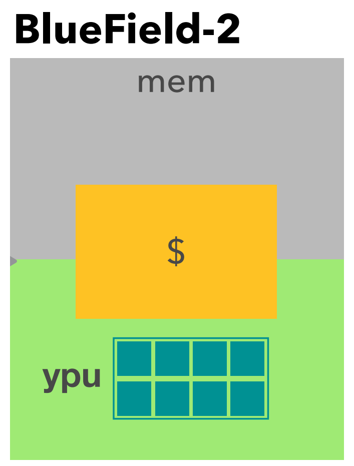
One host xPU (16 cores)



657 GF/s (fp64)

76.8 GB/s

BF-2 yPUs (no host)



80 GF/s

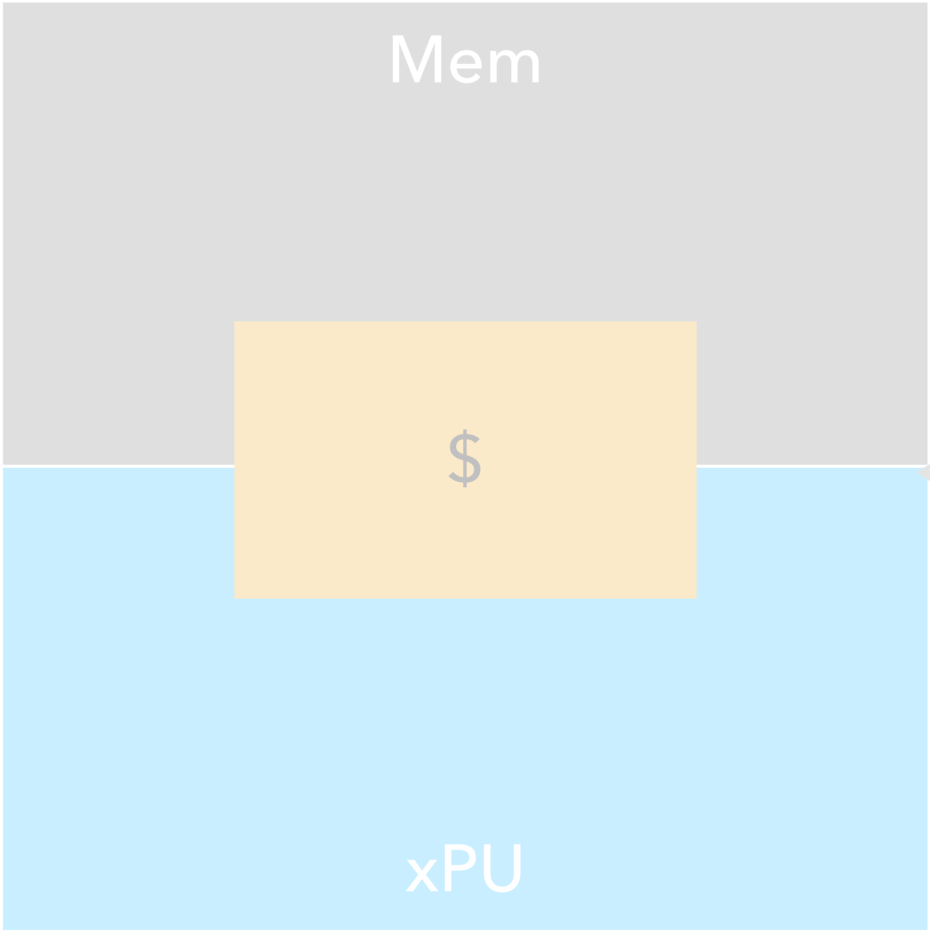
25.6 GB/s

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – “Smarter NICs for faster molecular dynamics: a case study.” doi:10.1109/IPDPS53621.2022.00063

Hypothetical: Multi-SmartNIC

a.k.a., revisiting the "iron law"

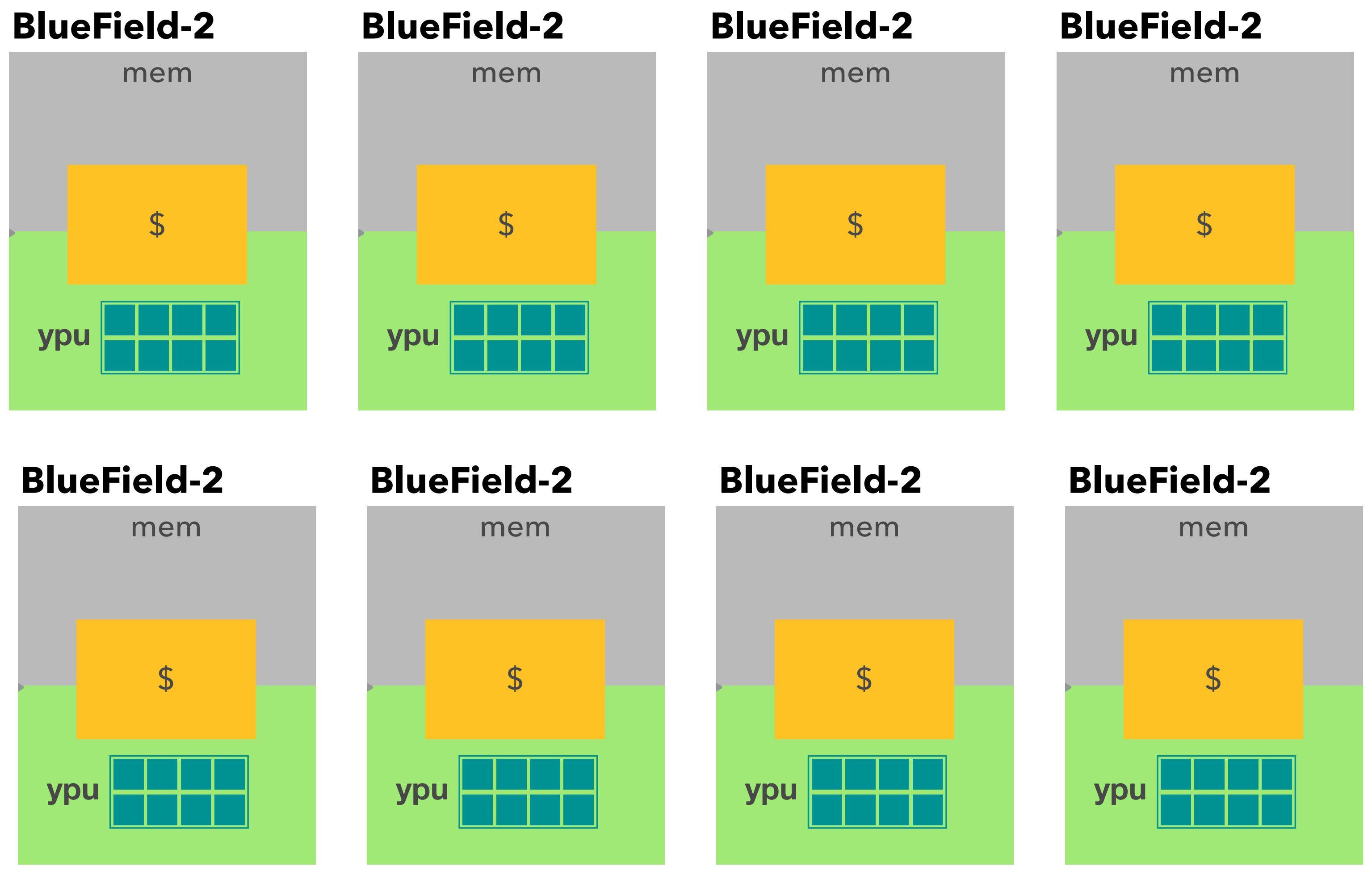
One host xPU (16 cores)



657 GF/s (fp64)

76.8 GB/s

8 x BF-2 yPUs (no host)



640 GF/s

204 GB/s

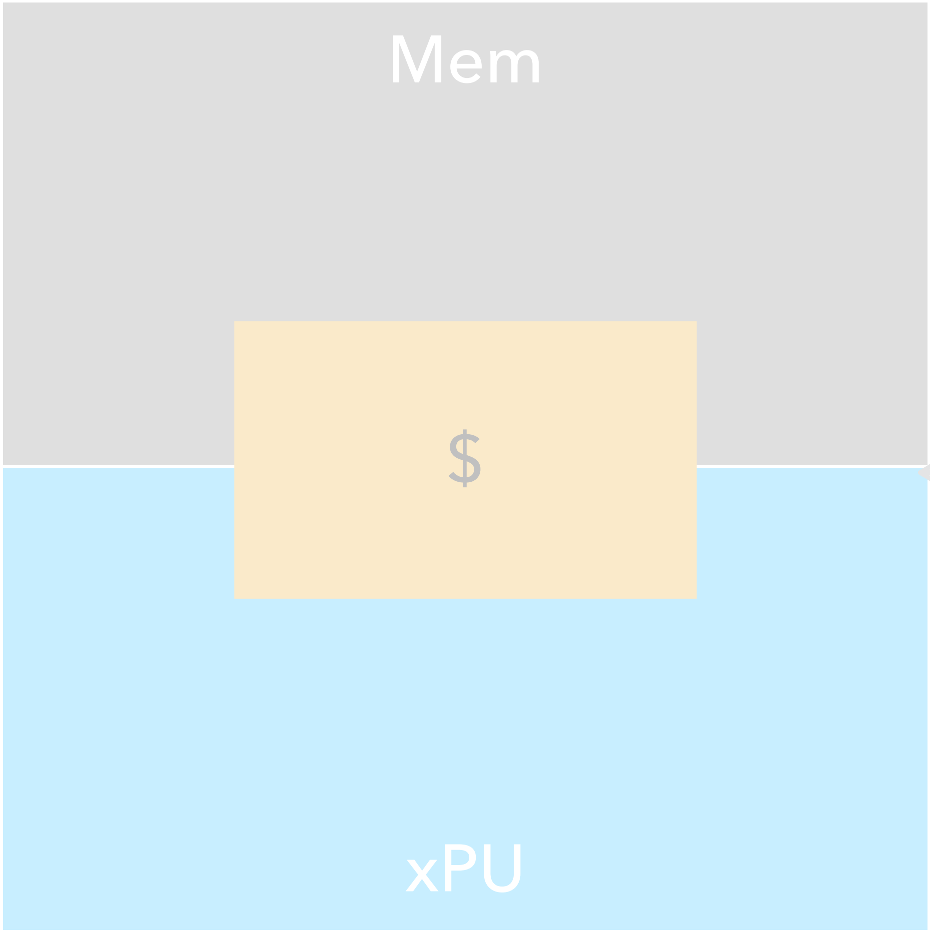
(aggregate)

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Hypothetical: Multi-SmartNIC

a.k.a., revisiting the "iron law"

One host xPU (16 cores)



~ 8.5 F:B

8 x BF-2 yPUs (no host)

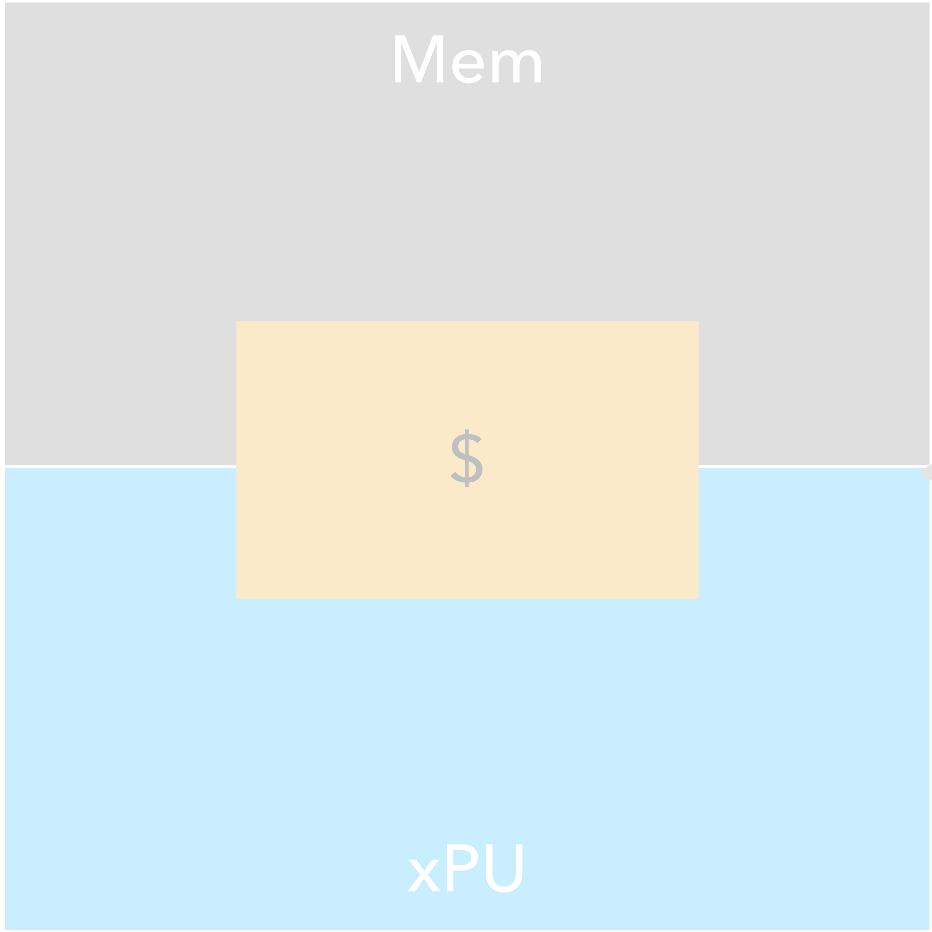
~ 3.1 F:B

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Hypothetical: Multi-SmartNIC

a.k.a., revisiting the "iron law"

One host xPU (16 cores)



Time = "1"

using all cores

8 x BF-2 yPUs (no host)

Speedup ~ 1.7x

Real measurement on MiniMD!

(Similar for P3DFFT, SuperLU_DIST)

S. Karamati (GT) et al. (Sandia, Queens U.; 2022) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Summary

Communication is fundamental and inevitable, so anything that addresses it should be pursued vigorously.

Restructuring algorithms, especially increasing asynchrony, can exploit smartNICs in HPC. We are pursuing a variety of candidates, including distributed time-tiled stencils, AMR, novel collectives, among others.

Many open questions remain, regarding other techniques, programming, runtimes, and performance modeling.