Communication-Avoiding Direct N-Body Algorithms

Penporn Koanantakool

penpornk@eecs.berkeley.edu

CS294/Math270: Communication-Avoiding Algorithms
University of California, Berkeley

April 11, 2016

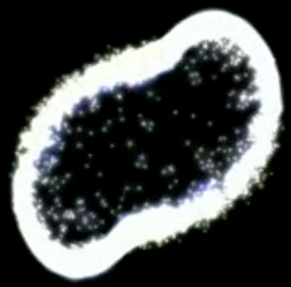
Outline

- Introduction
  - N-Body
  - Communication Lower Bounds
- Communication-Avoiding Algorithm
- Extension for cutoff distance
- Applications
- Conclusions

N-Body Galaxy Simulation

$O(n^2)$ -- 8,192 stars with variable mass

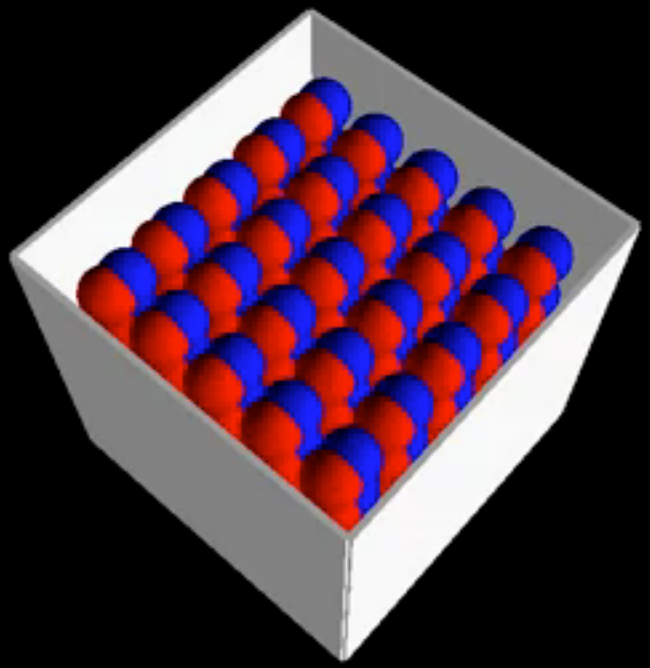
Source: https://www.youtube.com/watch?v=qzjaQ2rjup8



N-Body Molecular Dynamics Simulation

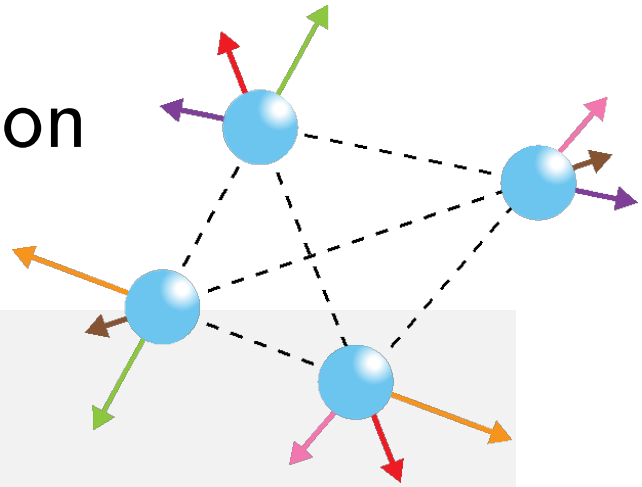
5x5x5 Carbon Monoxide Molecules (Morse & Lennard-Jones Potential)

Source: https://www.youtube.com/watch?v=t6o02qbWmQ8



Direct N-Body

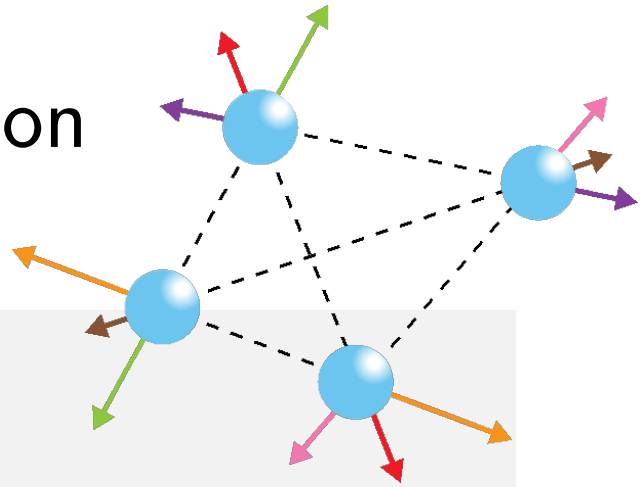
- $\mathbf{n}$ particles, all-pairs interaction
  - Molecules, galaxies, etc.



```
for t=1:timesteps
    for i=1:n
        for j=1:n
            force[i] += interact(particle[i], particle[j])
    for i=1:n
        move(particle[i], force[i])
```

Direct N-Body

- $\mathbf{n}$ particles, all-pairs interaction
  - Molecules, galaxies, etc.



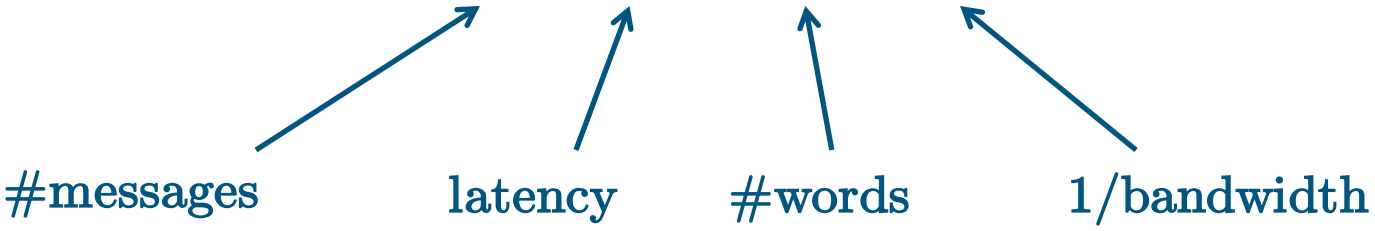
```
for t=1:timesteps
    for i=1:n
        for j=1:n                    } O(n²)
            force[i] += interact(particle[i], particle[j])
    for i=1:n
        move(particle[i], force[i])
```

- Approximated method is not considered here.
- Goal: Minimize communication.

Communication Model

- Per-processor cost along critical path
- Alpha-Beta model

$$cost = S \cdot \alpha + W \cdot \beta$$



- Lower-bound S and W to see if the algorithm is communication-optimal

Communication Lower Bounds

- From **Minimizing Communication in Numerical Linear Algebra** [Ballard et al. 2011a]:
  - F #flops per processor — $\mathrm{O(n^2/p)}$ flops
  - M size of fast memory in words — $\mathrm{O(M)}$ particles
  - H max #flops with M words — $\mathrm{O(M^2)}$ flops

$$\underset{\text{(\#messages)}}{S} = \Omega\left(\frac{F}{H}\right), \qquad \underset{\text{(\#words)}}{W} = \Omega\left(S \cdot M\right) = \Omega\left(\frac{F \cdot M}{H}\right)$$

- N-Body: $\mathbf{n}$ particles, $\mathbf{p}$ processors

$$S_{\text{N-Body}} = \Omega\left(\frac{n^2/p}{M^2}\right), \qquad W_{\text{N-Body}} = \Omega\left(\frac{n^2/p}{M}\right)$$

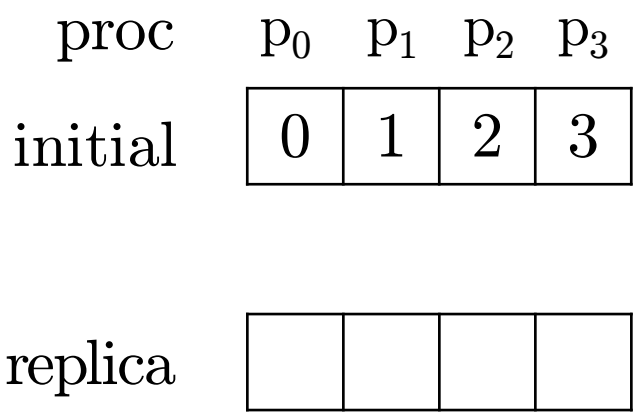
Finding H with HBL

- From **Communication Lower Bounds and Optimal Algorithms for Programs that Reference Arrays** [Christ et al. 2013]
  - $H = O\left(M^s\right),\ s = \max_{\Delta x \leq \mathbf{1}} \left(x_i + x_j\right)$ where

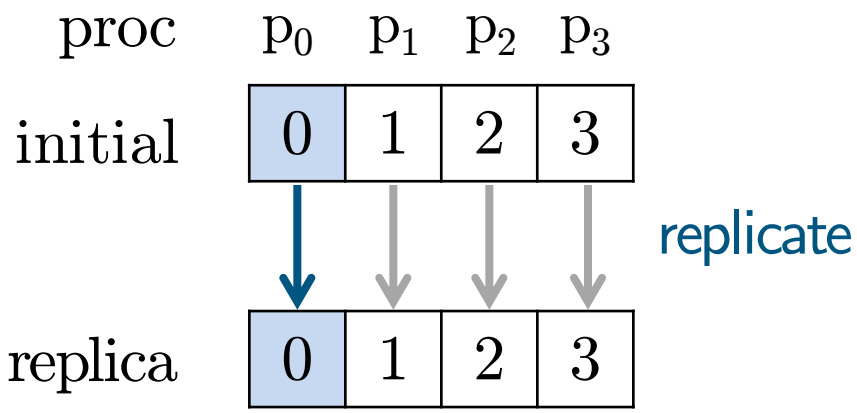
$$\Delta = \begin{array}{r} \\ \text{force} \\ \text{particle}_1 \\ \text{particle}_2 \end{array} \begin{array}{c} \begin{array}{cc} i & j \end{array} \\ \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}, \qquad x = \begin{pmatrix} x_i \\ x_j \end{pmatrix}$$

- $x = (1\ 1)^{\mathsf{T}}$ so $s = 2$ and $\mathrm{H} = \mathrm{O}(\mathrm{M}^2)$ ..... same!

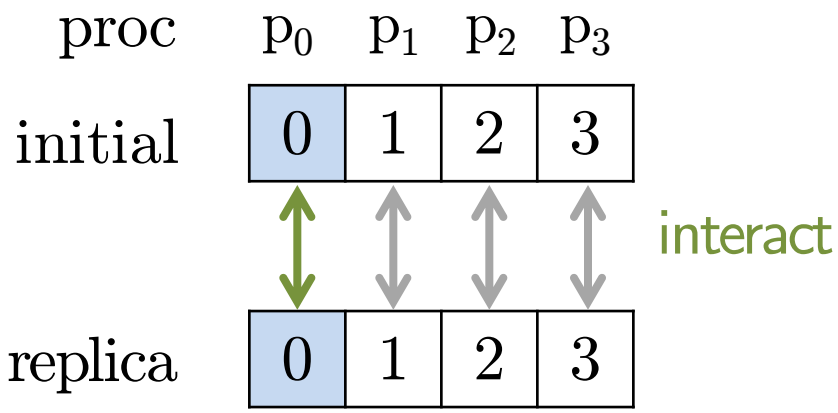
Naïve All-Pairs Interaction Algorithm



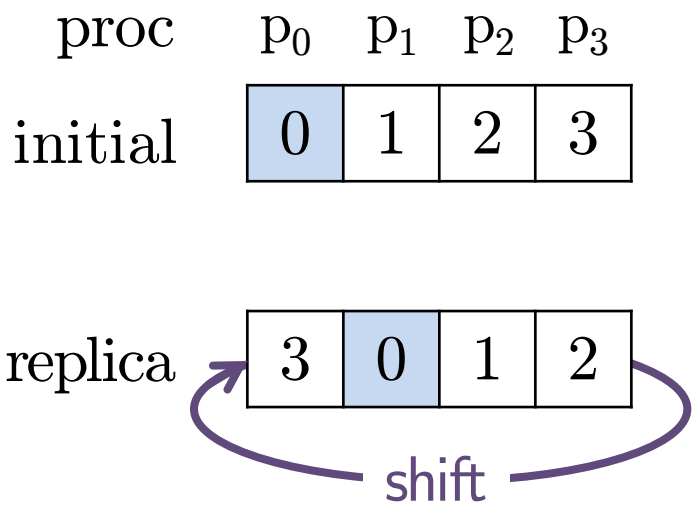
Naïve All-Pairs Interaction Algorithm



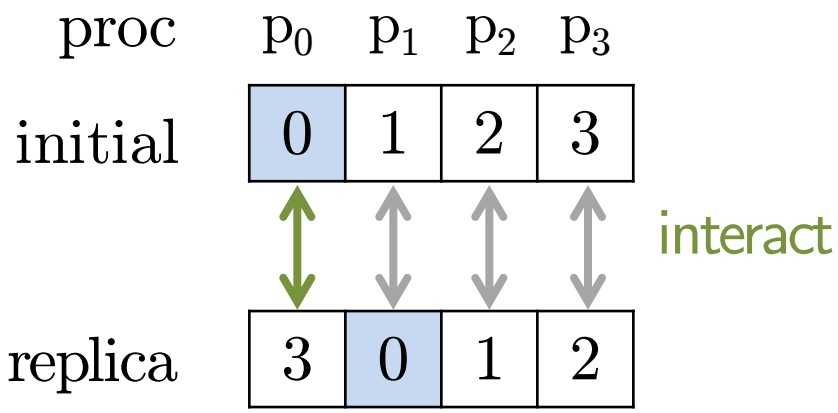
Naïve All-Pairs Interaction Algorithm



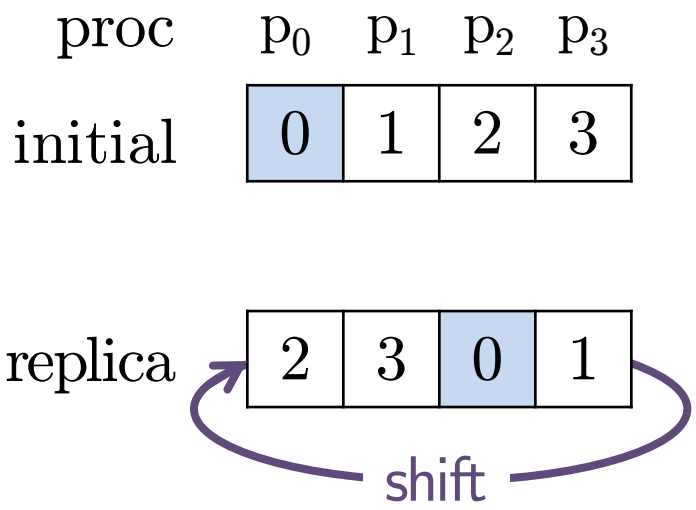
Naïve All-Pairs Interaction Algorithm



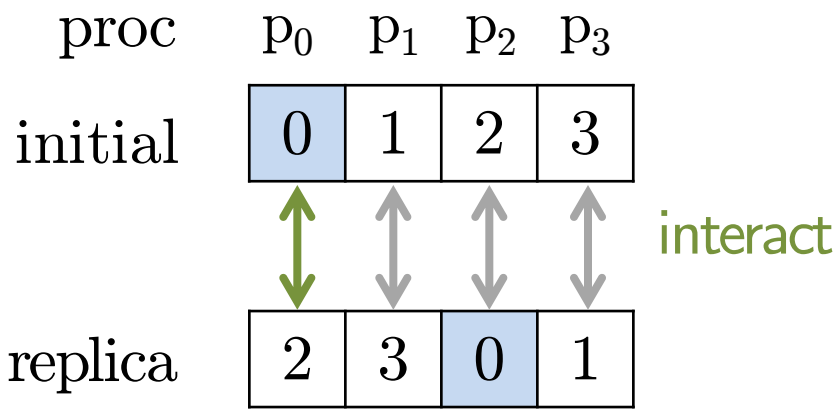
Naïve All-Pairs Interaction Algorithm



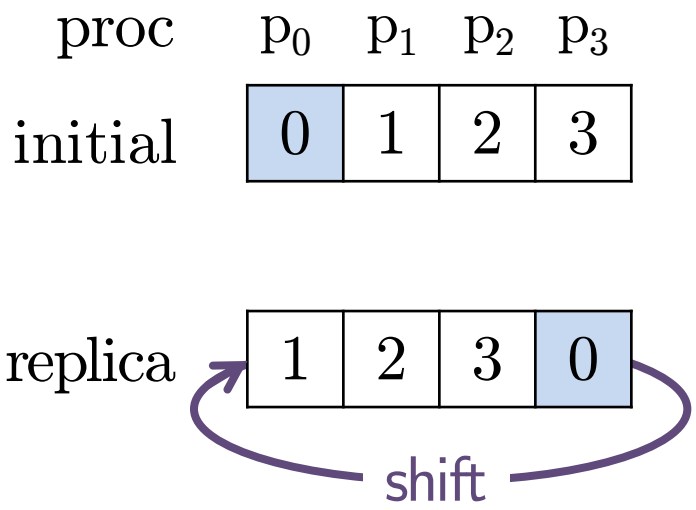
Naïve All-Pairs Interaction Algorithm



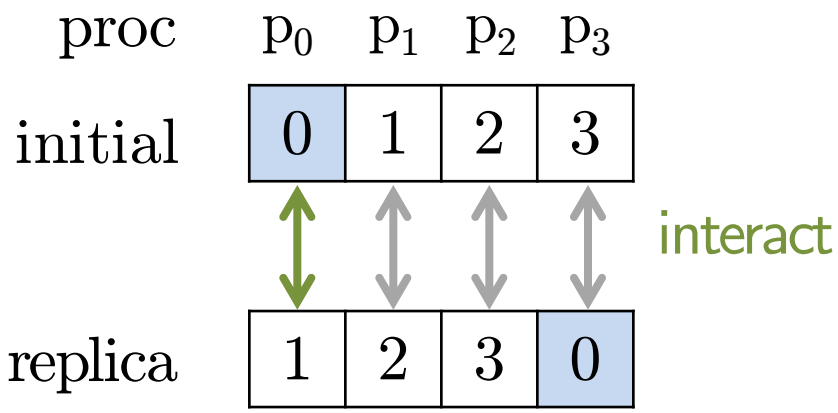
Naïve All-Pairs Interaction Algorithm



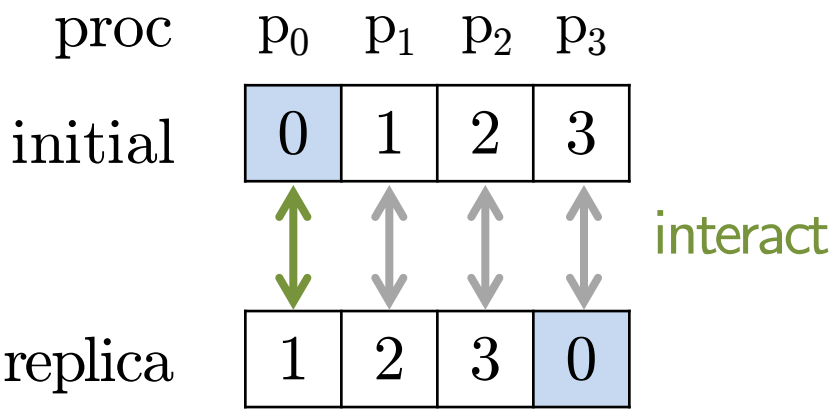
Naïve All-Pairs Interaction Algorithm



Naïve All-Pairs Interaction Algorithm



Naïve All-Pairs Interaction Algorithm



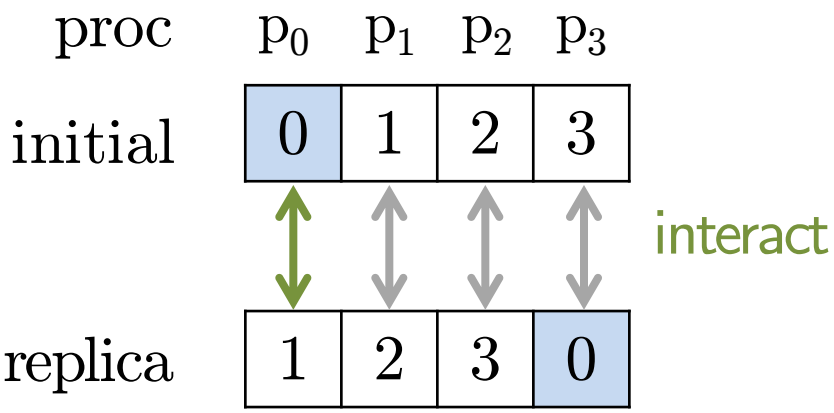
message size: $O(n/p)$
#messages: $O(p)$
#words sent: $O(n)$

- Lower Bounds for $M=O(n/p)$

$$S_{\text{allpairs}} = \Omega\left(\frac{n^2/p}{M^2}\right)$$

$$W_{\text{allpairs}} = \Omega\left(\frac{n^2/p}{M}\right)$$

Naïve All-Pairs Interaction Algorithm



message size: $\mathrm{O(n/p)}$

#messages: $\mathrm{O(p)}$

#words sent: $\mathrm{O(n)}$

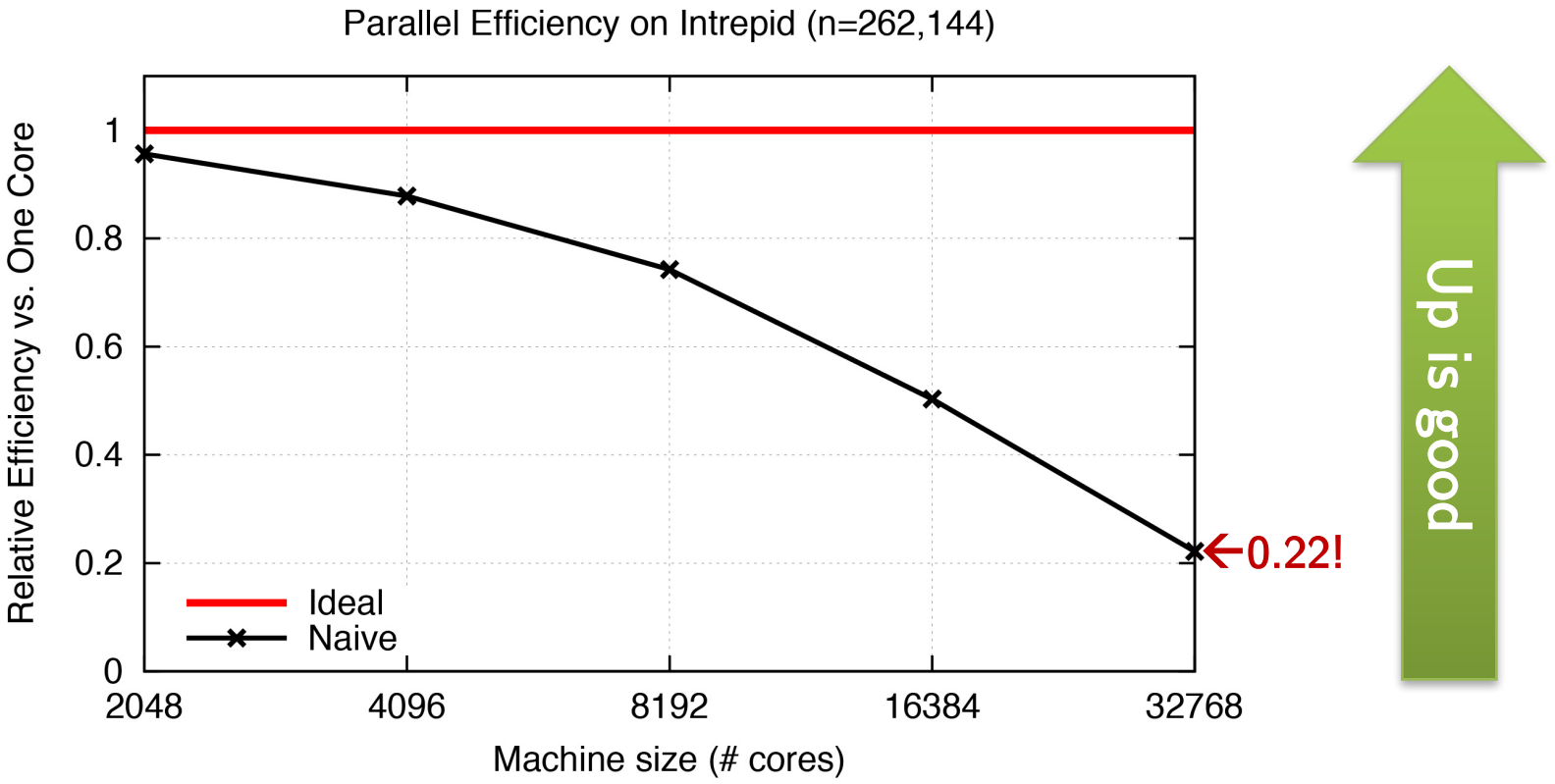
- Lower Bounds for $\mathrm{M{=}O(n/p)}$

$$S_{\text{allpairs}} = \Omega\left(\frac{n^2/p}{M^2}\right) = \Omega\left(\frac{n^2/p}{(n/p)^2}\right) = \Omega\left(p\right)$$

$$W_{\text{allpairs}} = \Omega\left(\frac{n^2/p}{M}\right) = \Omega\left(\frac{n^2/p}{n/p}\right) = \Omega\left(n\right)$$

- Communication-optimal for $\mathrm{M{=}O(n/p)}$

Bad Scalability



- We can do better

$$S_{\text{N-Body}} = \Omega\left(\frac{n^2/p}{M^2}\right), \quad W_{\text{N-Body}} = \Omega\left(\frac{n^2/p}{M}\right)$$

Exploiting Extra Memory

- By making c replicas $M = O\left(c \cdot \frac{n}{p}\right)$, we get

$$S_{\text{N-Body}} = \Omega\left(\frac{F}{M^2}\right) = \Omega\left(\frac{p}{c^2}\right)$$

$$W_{\text{N-Body}} = \Omega\left(\frac{F}{M}\right) = \Omega\left(\frac{n}{c}\right)$$

- Existing algorithms only use $c = \sqrt{p}$
  - [Plimpton 1995], [Snir 2004], etc.
- CA N-Body [Driscoll et al. 2013] allows $1 \leq c \leq \sqrt{p}$*
  - Support machines with limited memory
  - The best replication factor is often not $\sqrt{p}$

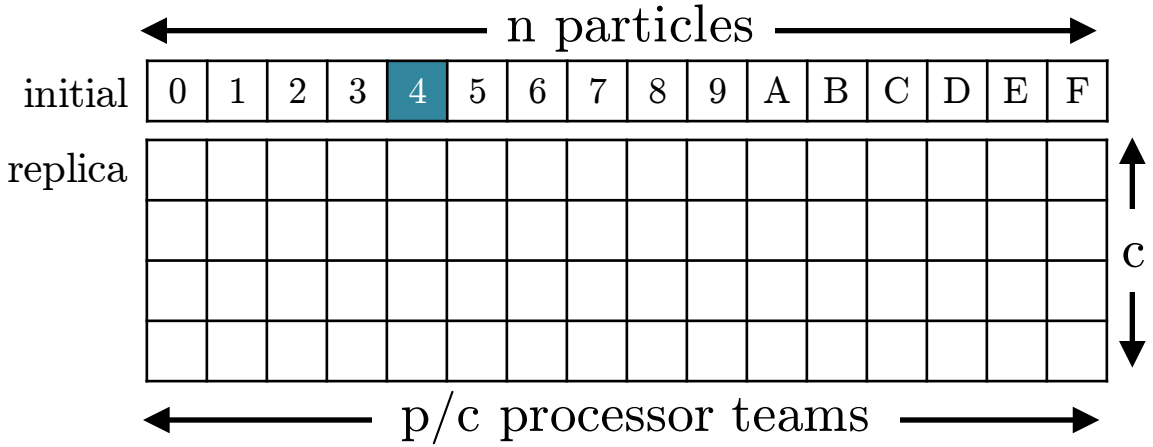
*Formal proof in HBL paper

Outline

- Introduction
  - N-Body
  - Communication Lower Bounds
- Communication-Avoiding Algorithm
- Extension for cutoff distance
- Applications
- Conclusions

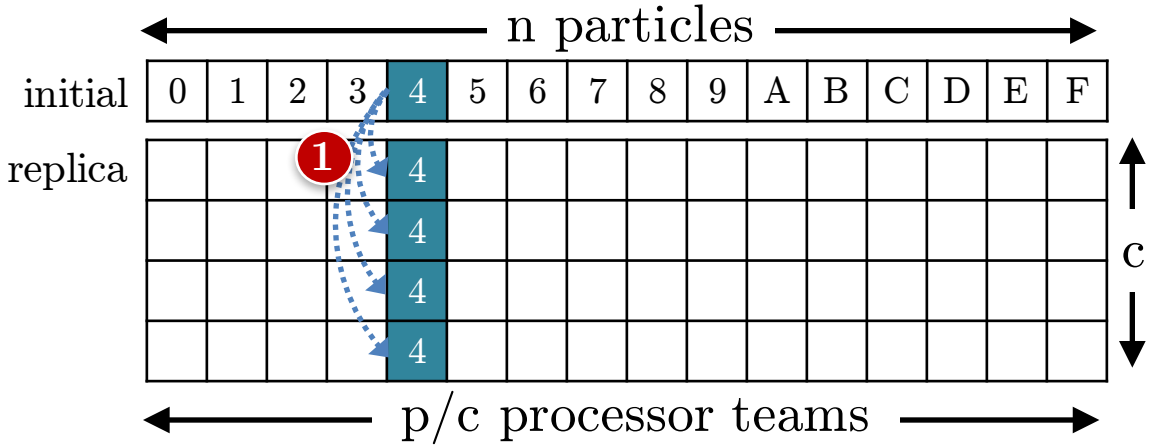
The CA-All-Pairs Algorithm

- $p = 64$, $c = 4$, 16 teams



The CA-All-Pairs Algorithm

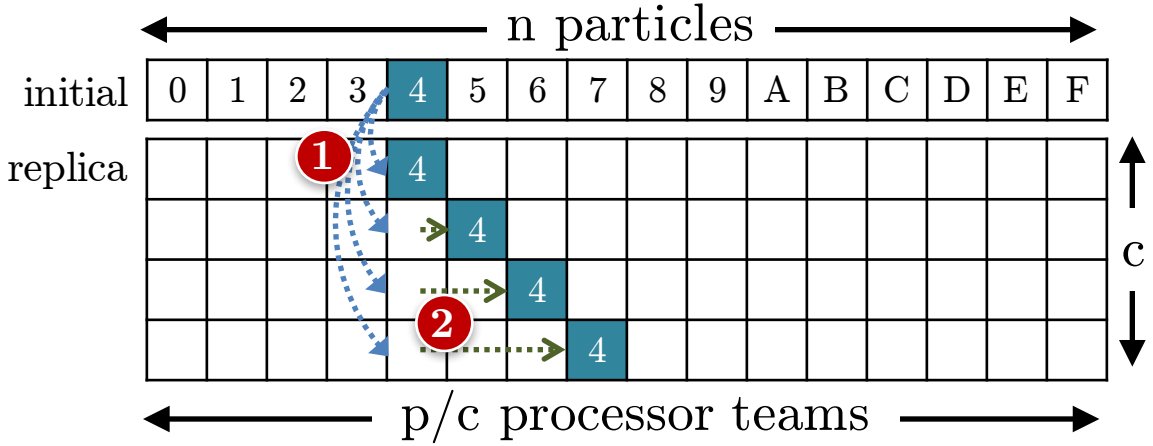
- $p = 64$, $c = 4$, 16 teams



1. Broadcast to team members

The CA-All-Pairs Algorithm

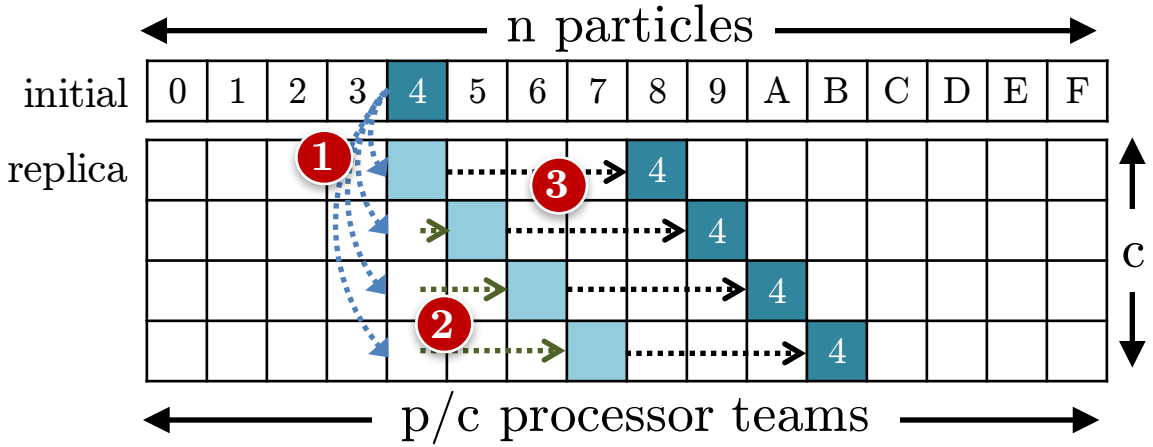
- $p = 64$, $c = 4$, 16 teams



2. Shift by row ID,
   and calculate interactions

The CA-All-Pairs Algorithm

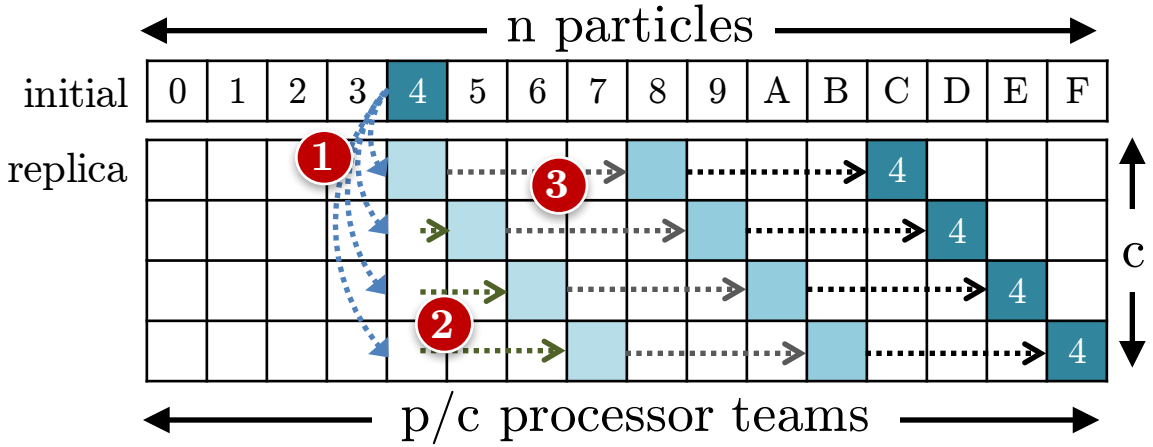
- $p = 64$, $c = 4$, 16 teams



3. Shift by c,
   and calculate interactions

The CA-All-Pairs Algorithm

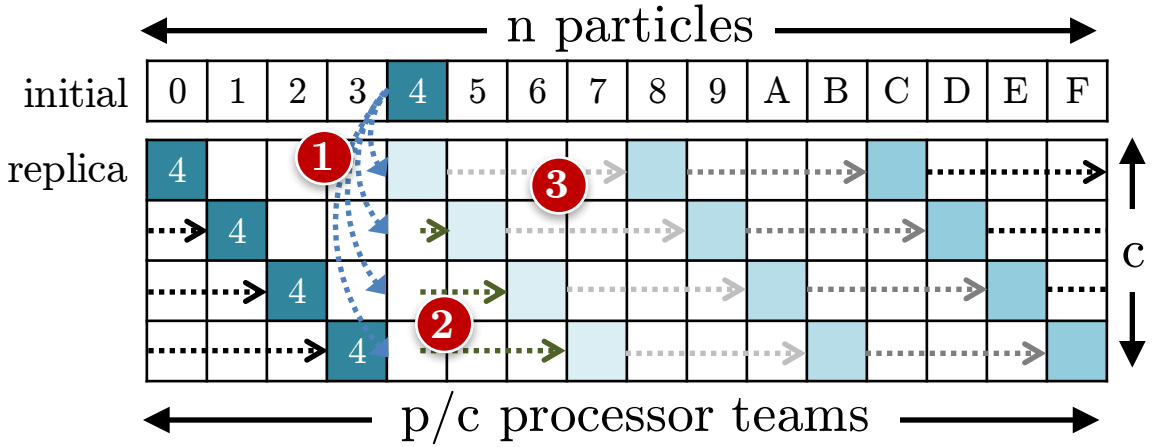
- $p = 64$, $c = 4$, 16 teams



3. Shift by c,
   and calculate interactions

The CA-All-Pairs Algorithm

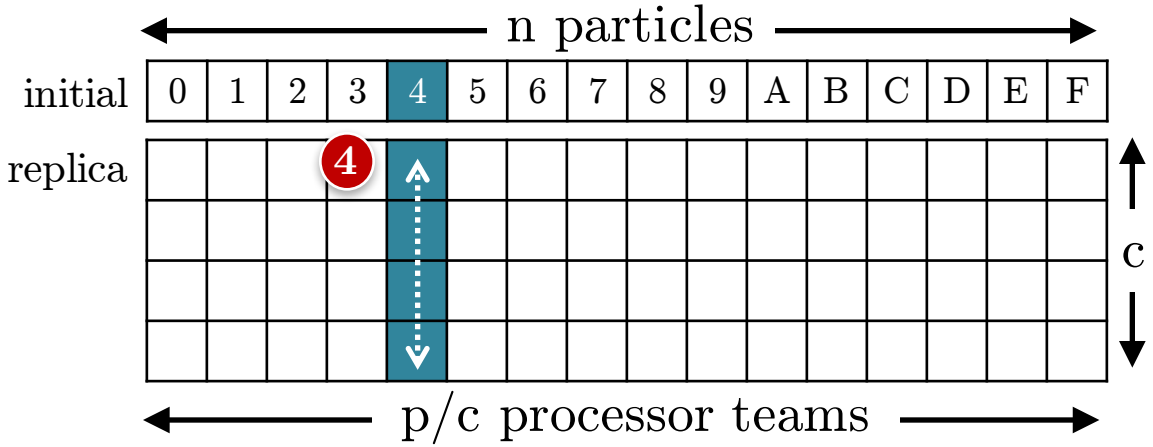
- $p = 64$, $c = 4$, 16 teams



3. Shift by c,
   and calculate interactions

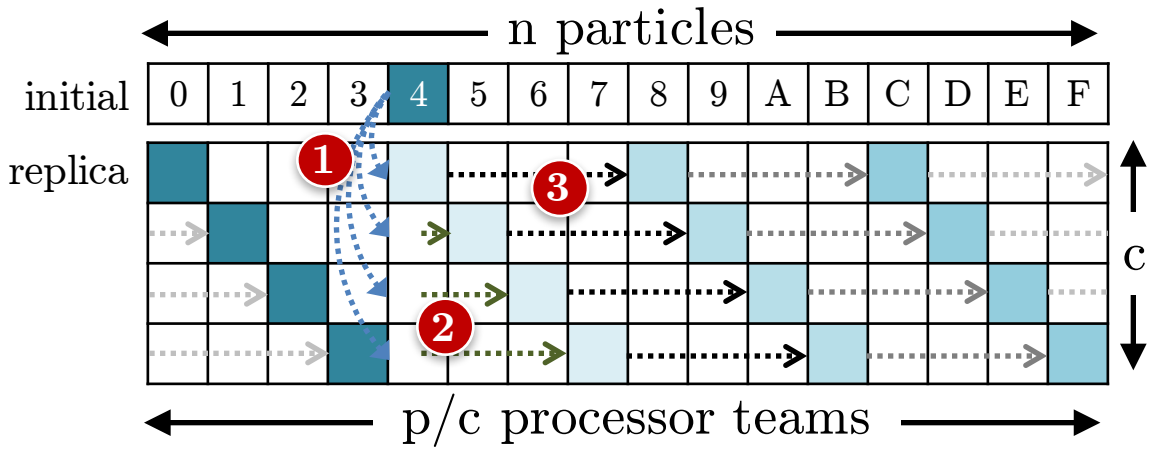
The CA-All-Pairs Algorithm

- $p = 64$, $c = 4$, 16 teams



4. Reduce all interactions calculated by all team members

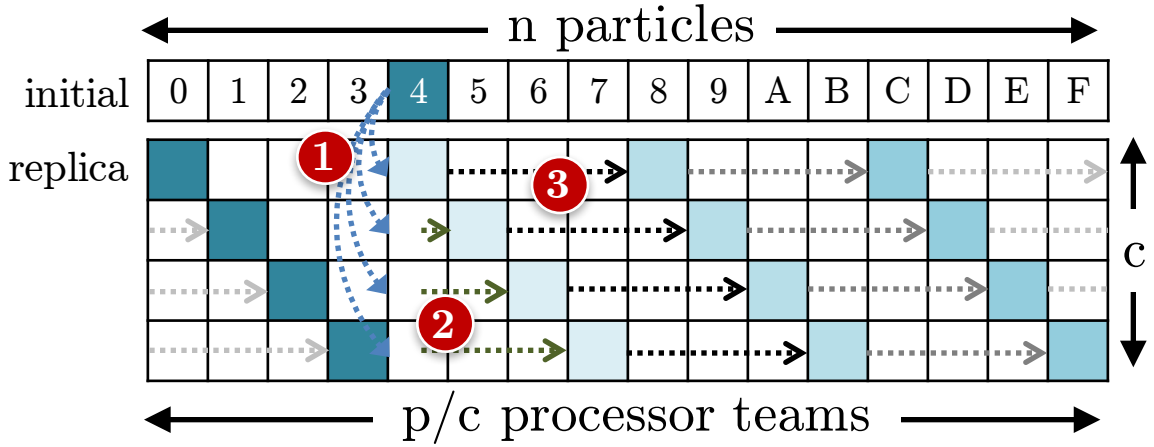
Communication Optimality



n: #particles
p: #processors
c: #copies

	Message Size	#messages (S)	#words (W)
1. Broadcast			
2. Skew			
3. Shift			
4. Reduce			

Communication Optimality



n: #particles
p: #processors
c: #copies

	Message Size	#messages (S)	#words (W)
1. Broadcast	$O\left(\frac{nc}{p}\right)$	$O(\log c)$	$O\left(\frac{nc\log c}{p}\right)$
2. Skew	$O\left(\frac{nc}{p}\right)$	$O(1)$	$O\left(\frac{nc}{p}\right)$
3. Shift	$O\left(\frac{nc}{p}\right)$	$O\left(\frac{p/c}{c}\right) = \boldsymbol{O\left(\frac{p}{c^2}\right)}$	$\boldsymbol{O\left(\frac{n}{c}\right)}$
4. Reduce	$O\left(\frac{nc}{p}\right)$	$O(\log c)$	$O\left(\frac{nc\log c}{p}\right)$

Dominant factor

Bounds

$$S_{\text{N-Body}} = \Omega\left(\frac{p}{c^2}\right)$$

$$W_{\text{N-Body}} = \Omega\left(\frac{n}{c}\right)$$

The costs match the lower bounds!

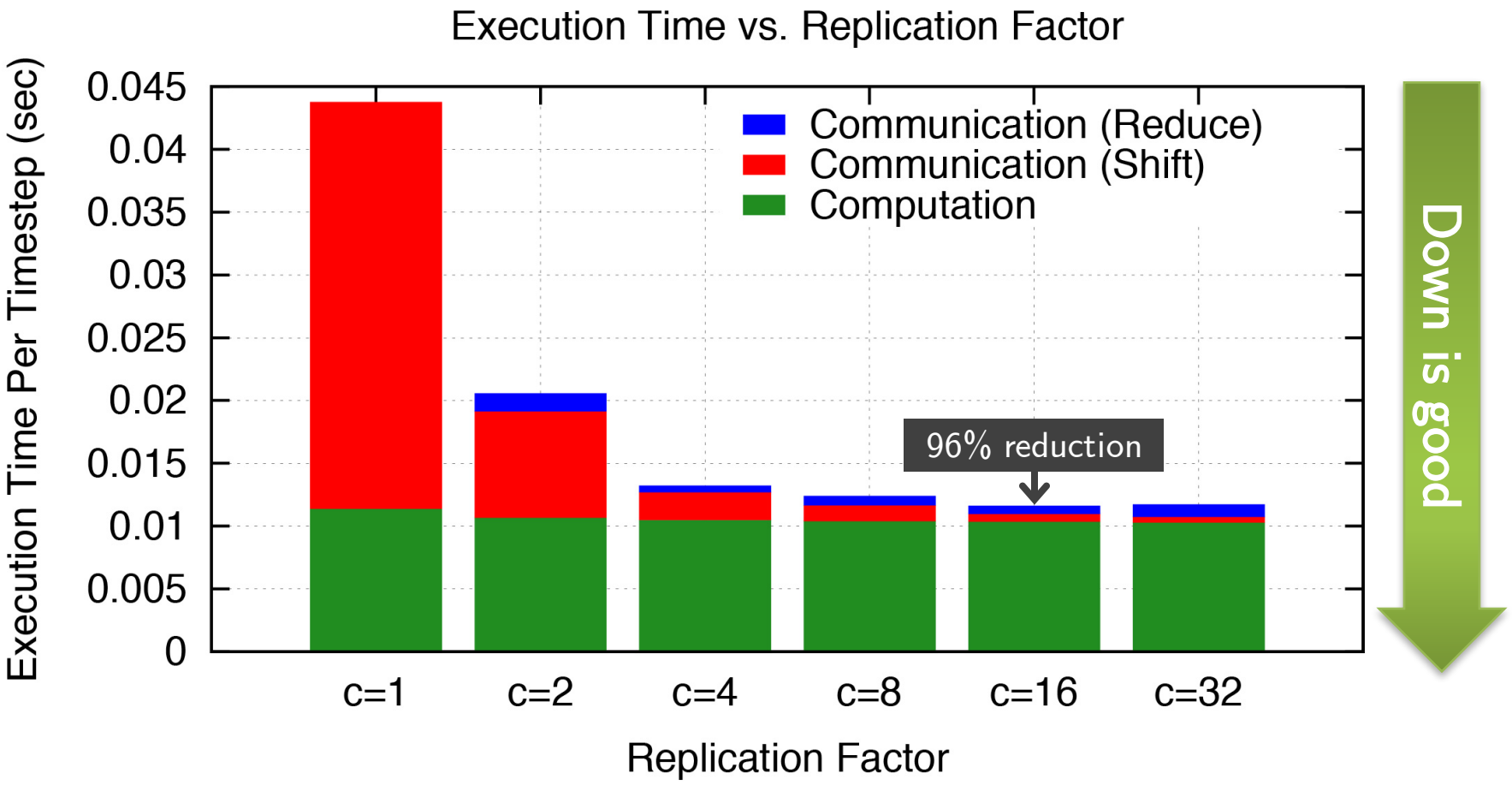
Test Environment

- N-Body code
  - Flat MPI
  - 56-byte particles
  - Repulsive force proportional to $1/r^2$
  - Reflective boundary conditions

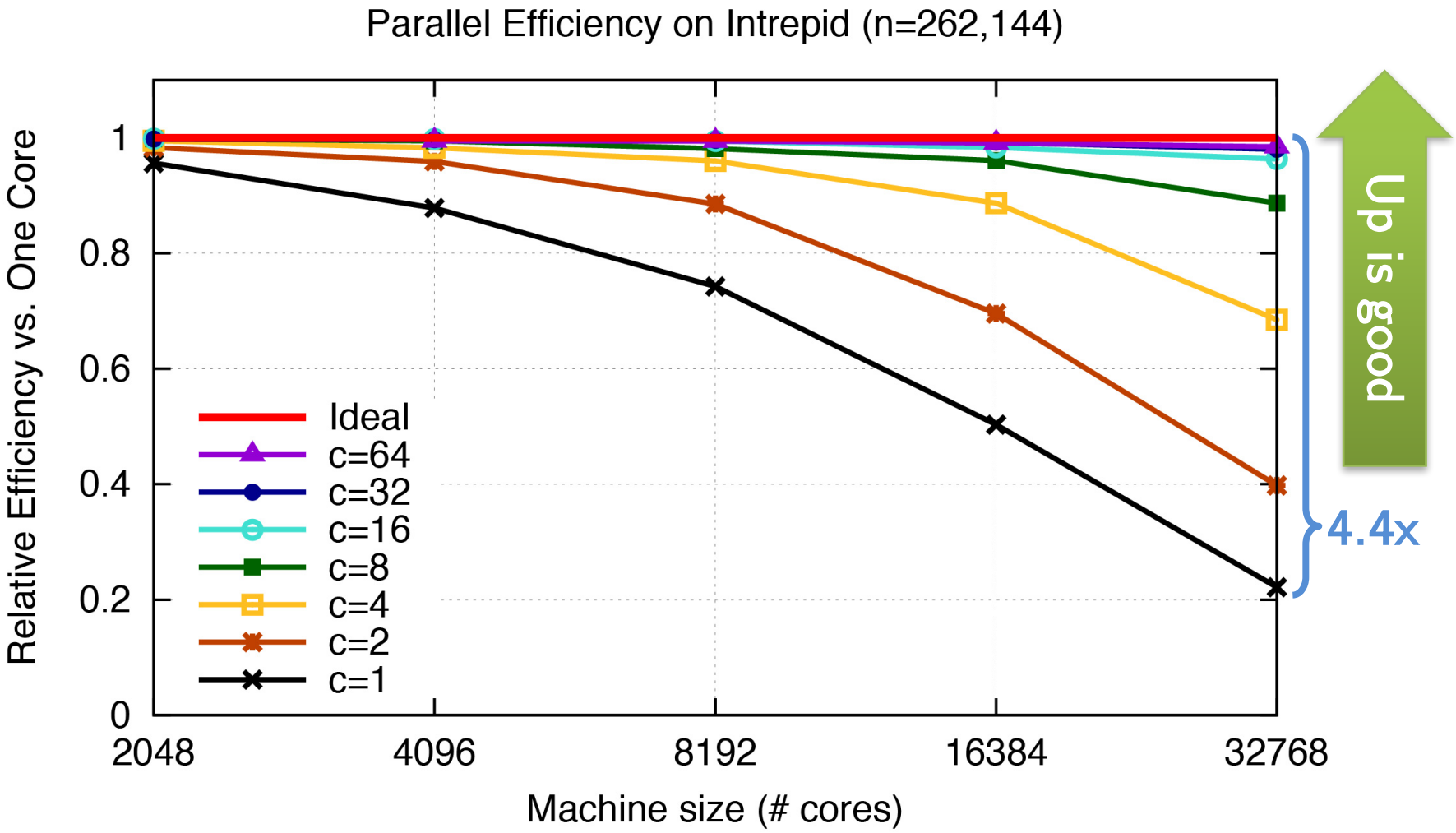
- Platforms
  - **Hopper @NERSC**
    - Cray XE-6: fat 24-core **NUMA** node
    - 3D-torus Cray Gemini interconnect
  - **Intrepid @ALCF**
    - IBM BlueGene/P: quad-core node
    - 3D-torus, **topology-aware partitioning & collectives**

Less Communication..

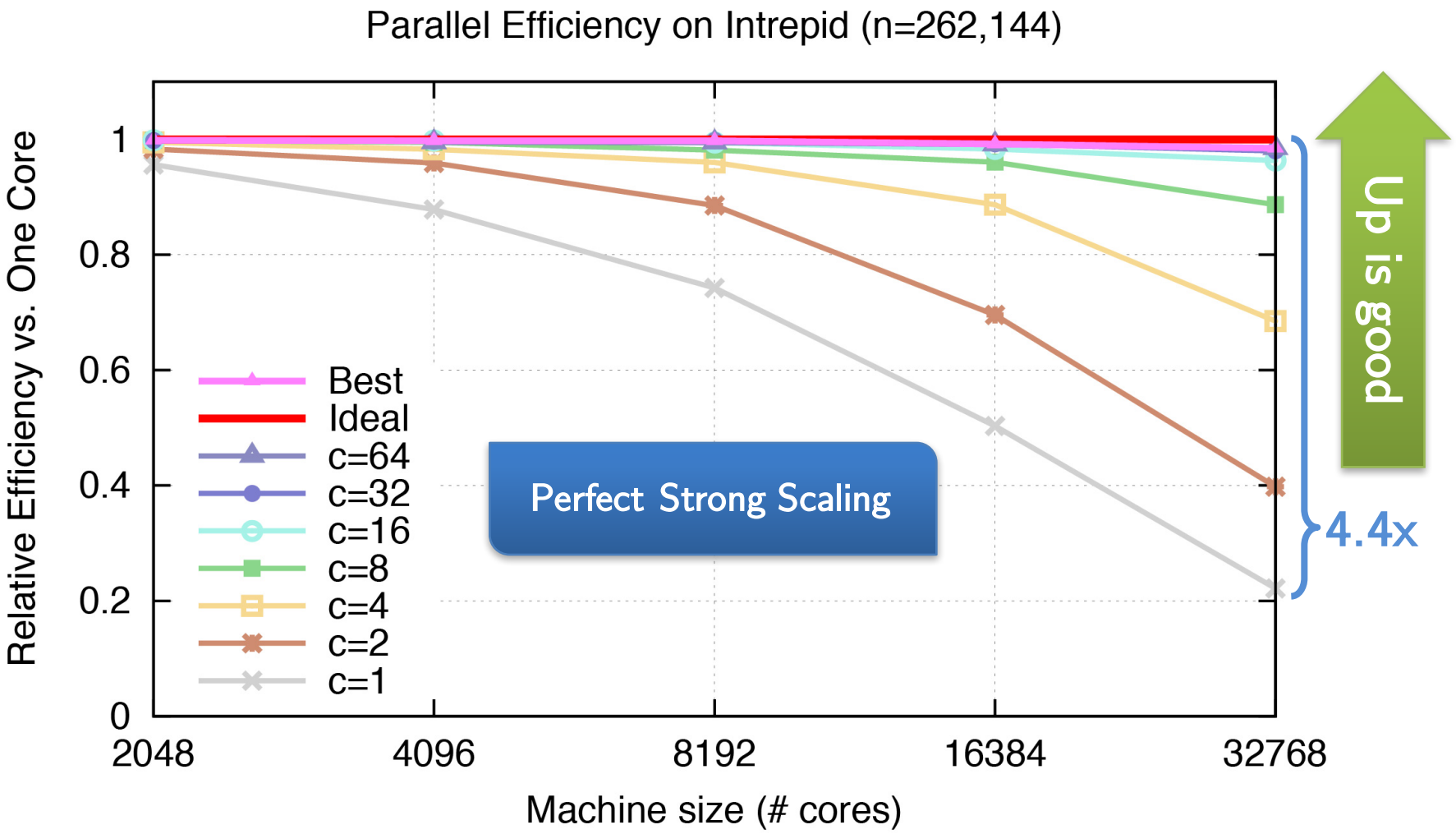
- Hopper; $\mathrm{n}{=}24\mathrm{K}$ particles, $\mathrm{p}{=}6\mathrm{K}$ cores



Near-Perfect Scalability



Near-Perfect Scalability



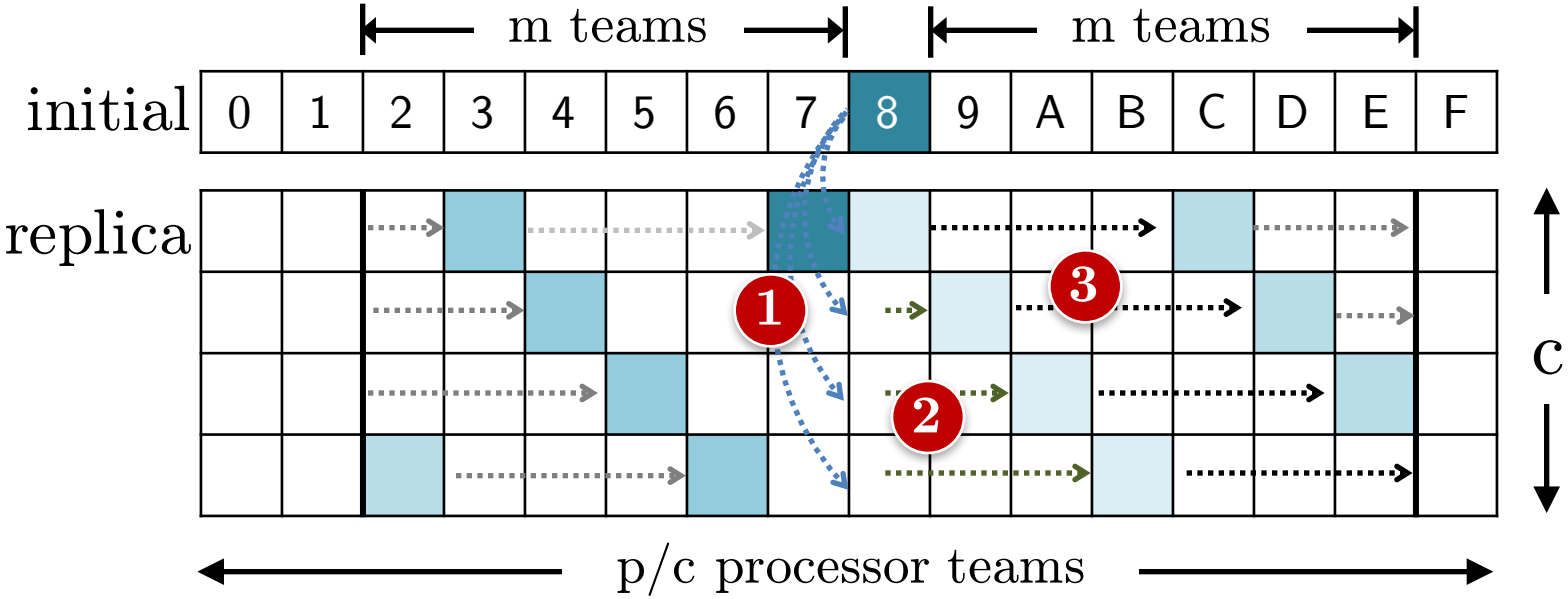
Outline

- Introduction
  - N-Body
  - Communication Lower Bounds
- Communication-Avoiding Algorithm
- Extension for cutoff distance
- Symmetries and k-way N-Body
- Applications

Interactions with Cutoff Radius

- Don't interact if distance ≥ cutoff radius
- Suggests spatial decomposition
- Assumptions
  - Uniform distribution
  - Cutoff distance spans multiple processor areas
- Simple extension
  - Shifts within only cutoff area
  - Works for any dimensionality of simulation space
  - Still communication-optimal
- More details in the paper

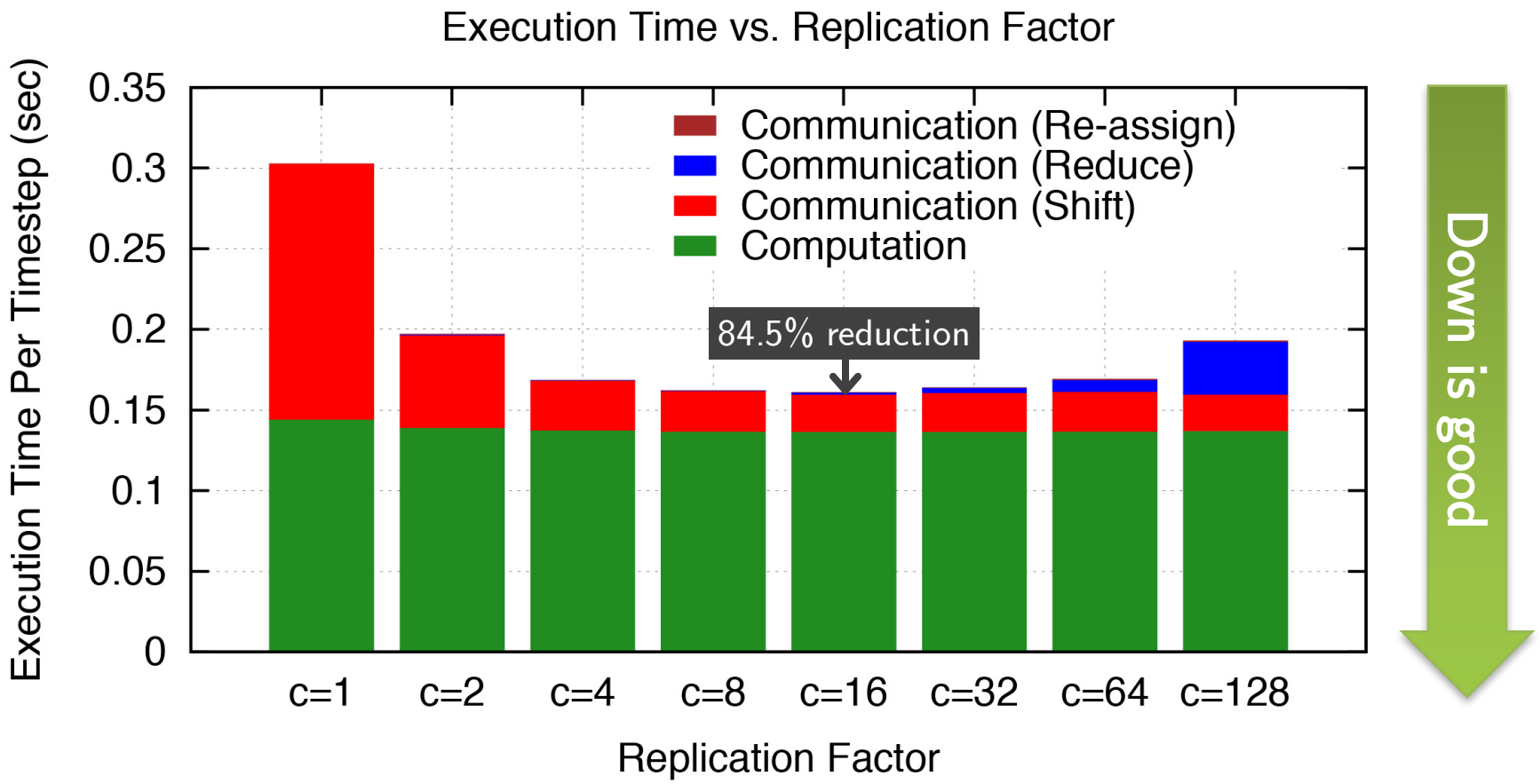
1D-Space Cutoff Algorithm



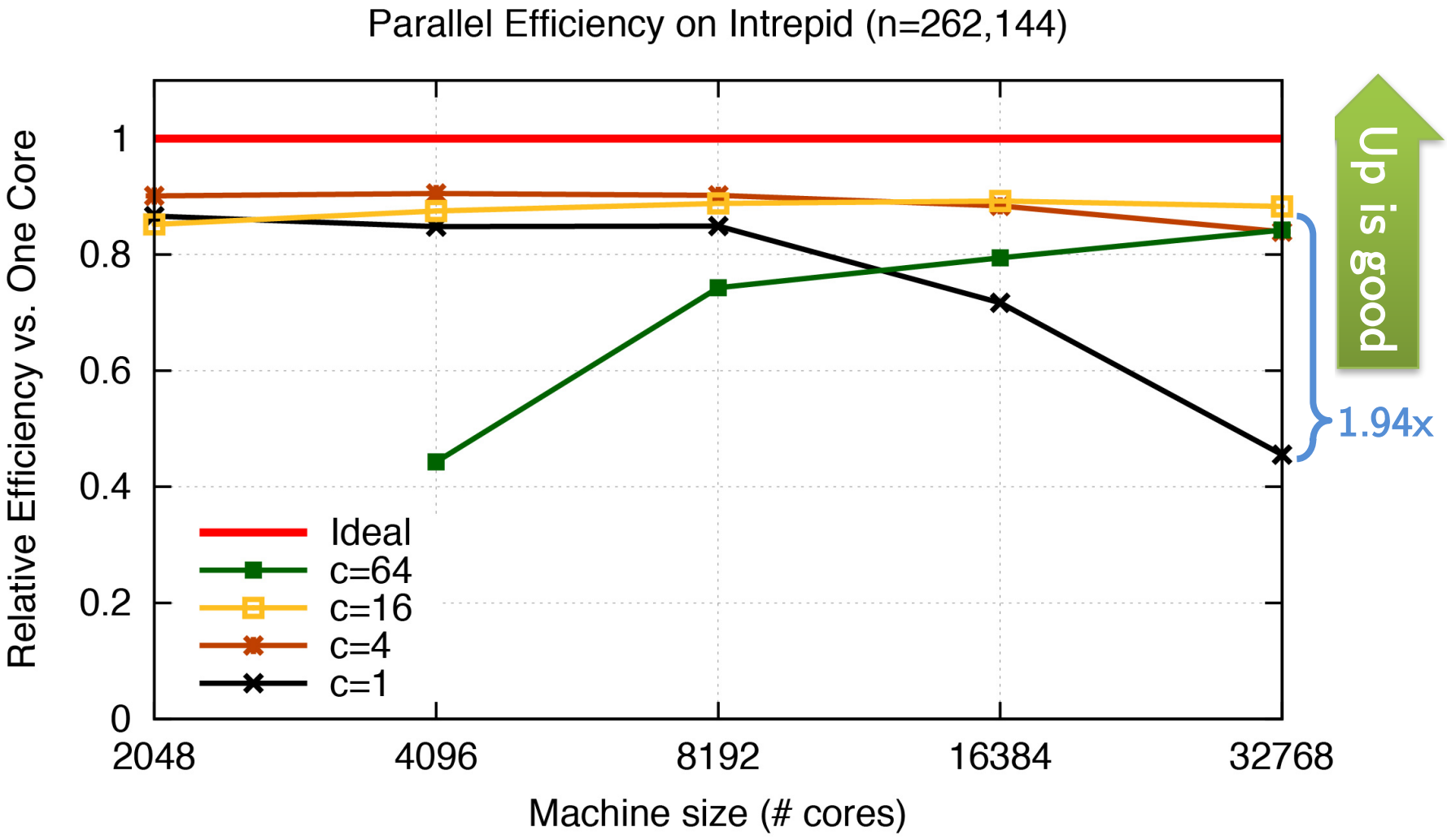
- Shift-modulo (wrap-around)

CA1D: Time Breakdown

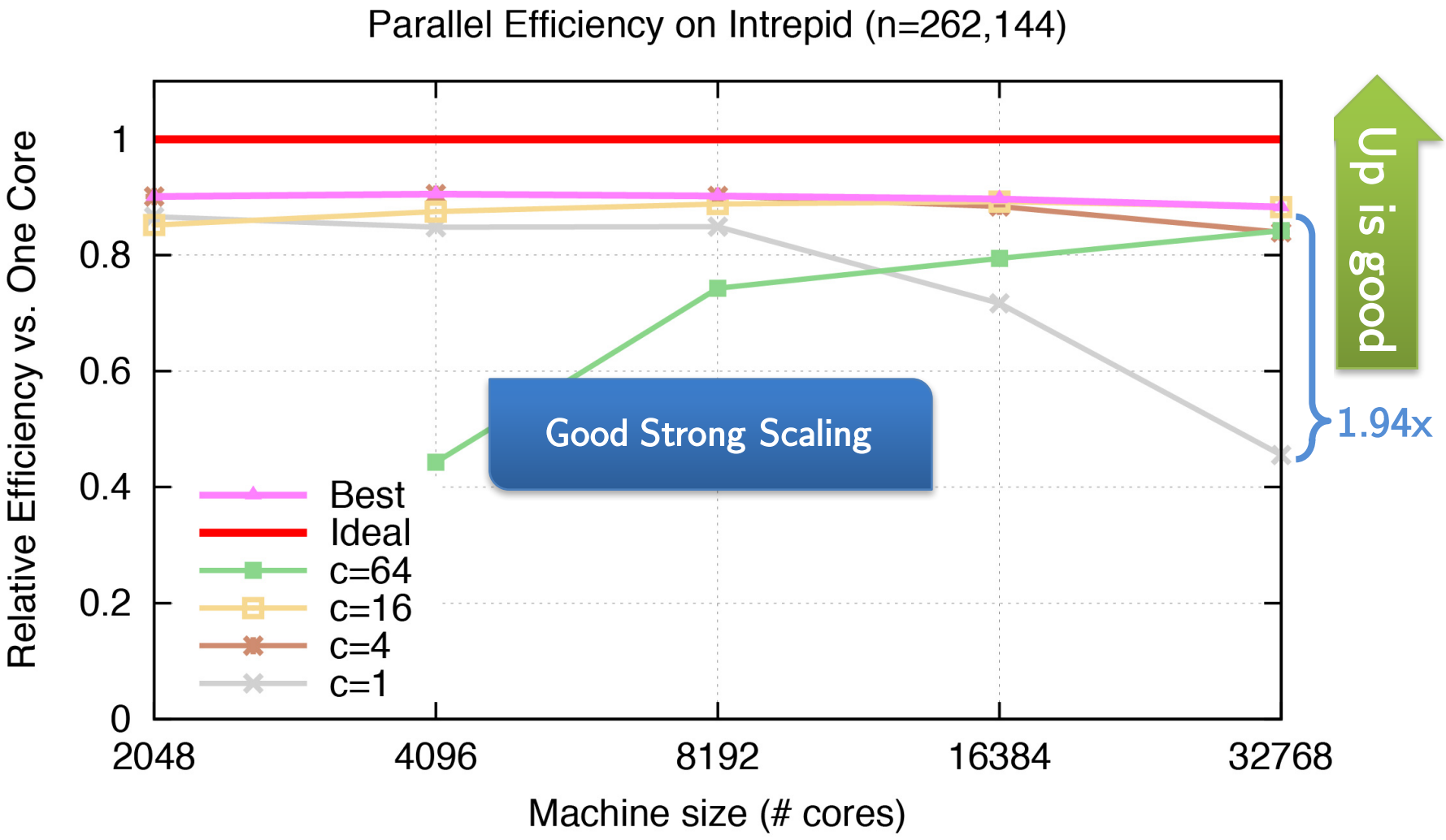
- Intrepid; $n=262K$ particles, $p=32K$ cores



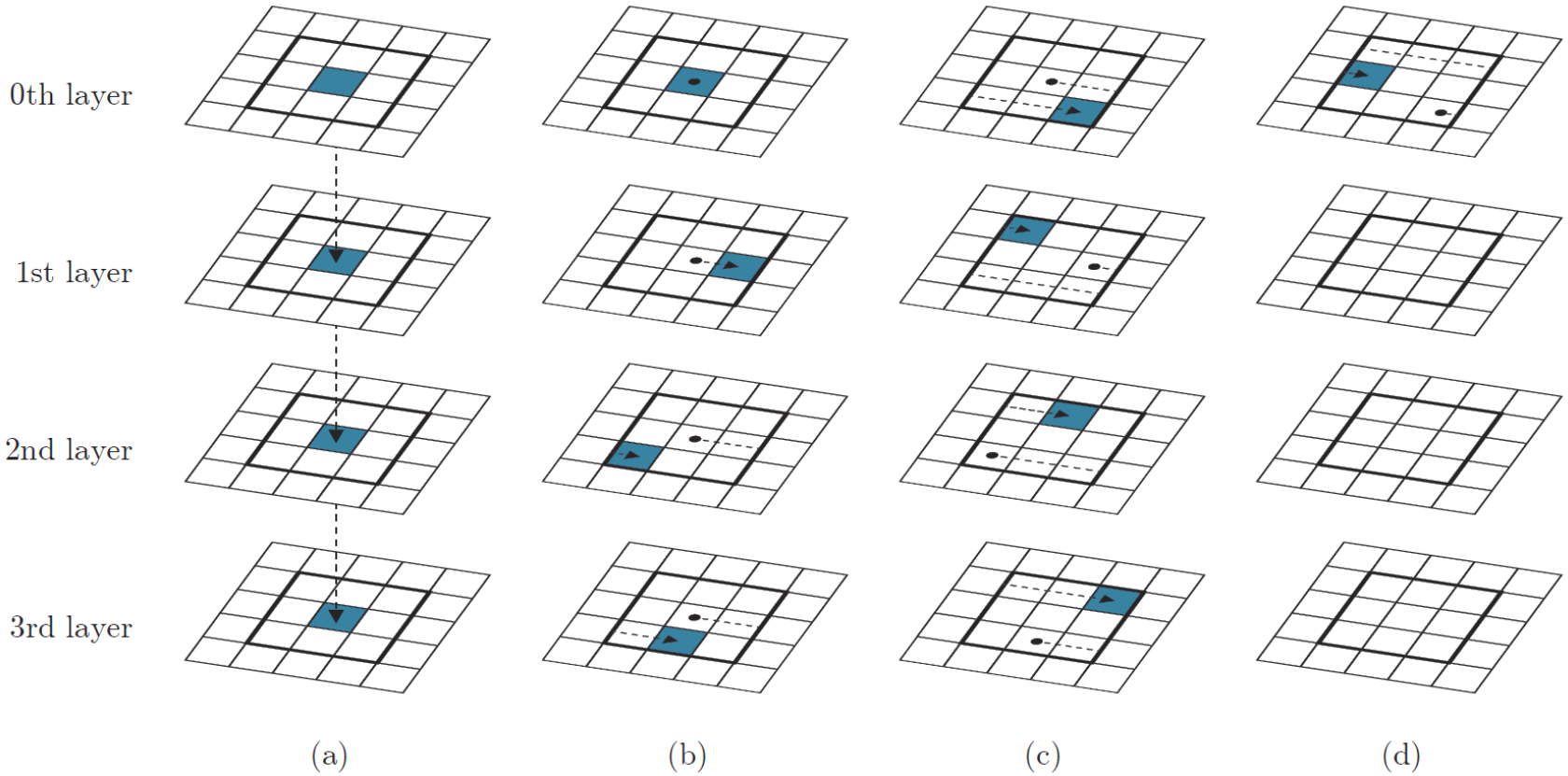
CA1D: Better Scalability



CA1D: Better Scalability



2D-space Cutoff Algorithm

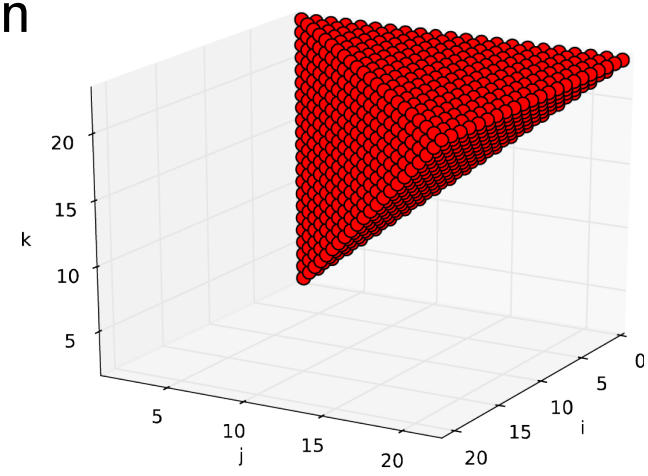


Outline

- Introduction
  - N-Body
  - Communication Lower Bounds
- Communication-Avoiding Algorithm
- Extension for cutoff distance
- Applications
- Conclusions

'Other' N-Body Problems

- Bottom solver of hierarchical N-Body
- k-way interaction N-Body
  - Should make use of force symmetries
  - Same communication-avoiding technique
  - All-triplets 3-Body
    - 99.98% communication reduction
    - Observed up to 42x speedup on tens of thousands of cores [Koanantakool and Yelick 2014]



$O(n^2)$ Database Join

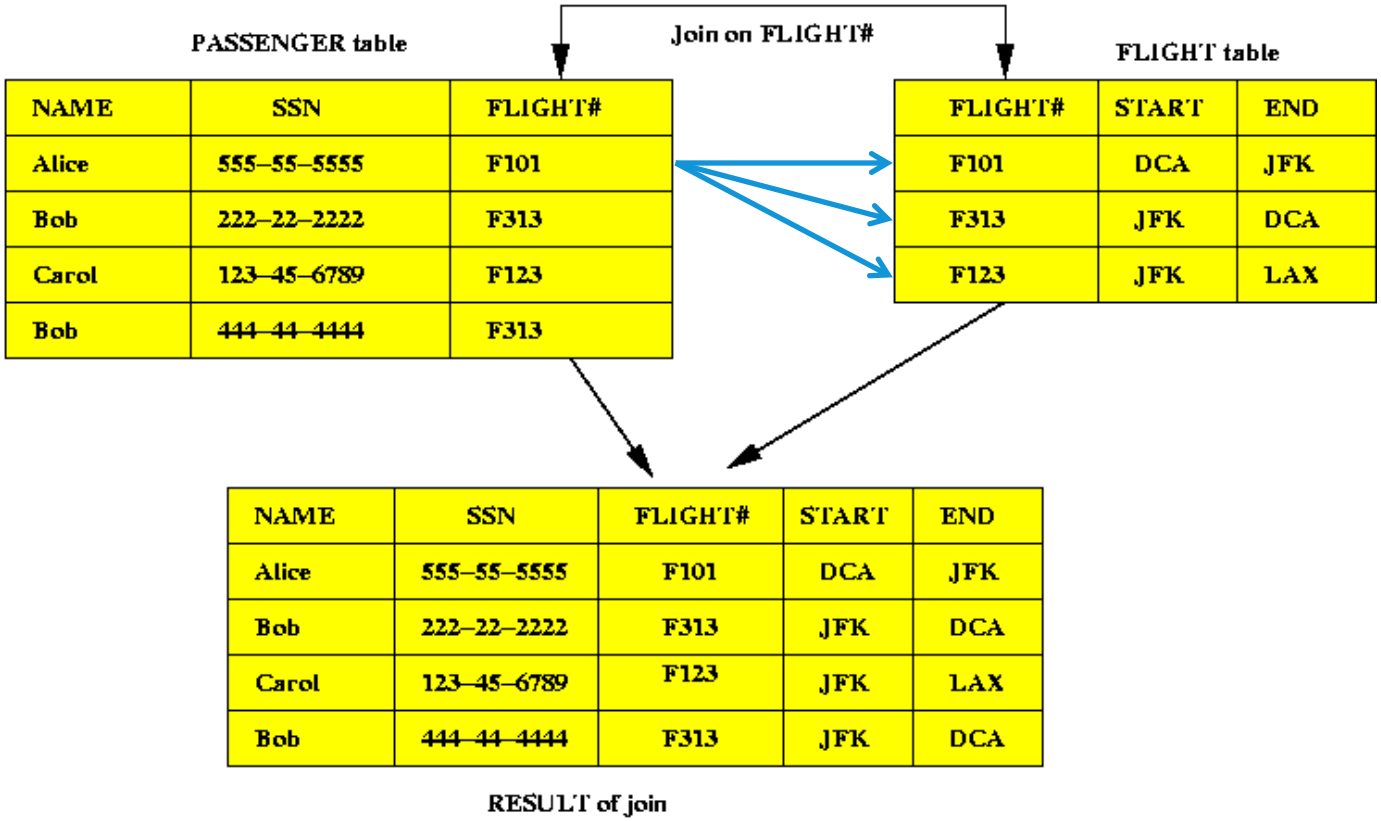


Image source:
https://www.seas.gwu.edu/~simhaweb/cs177/computinglecture/part3.html

CG: $O(n^2)$ Collision Detection

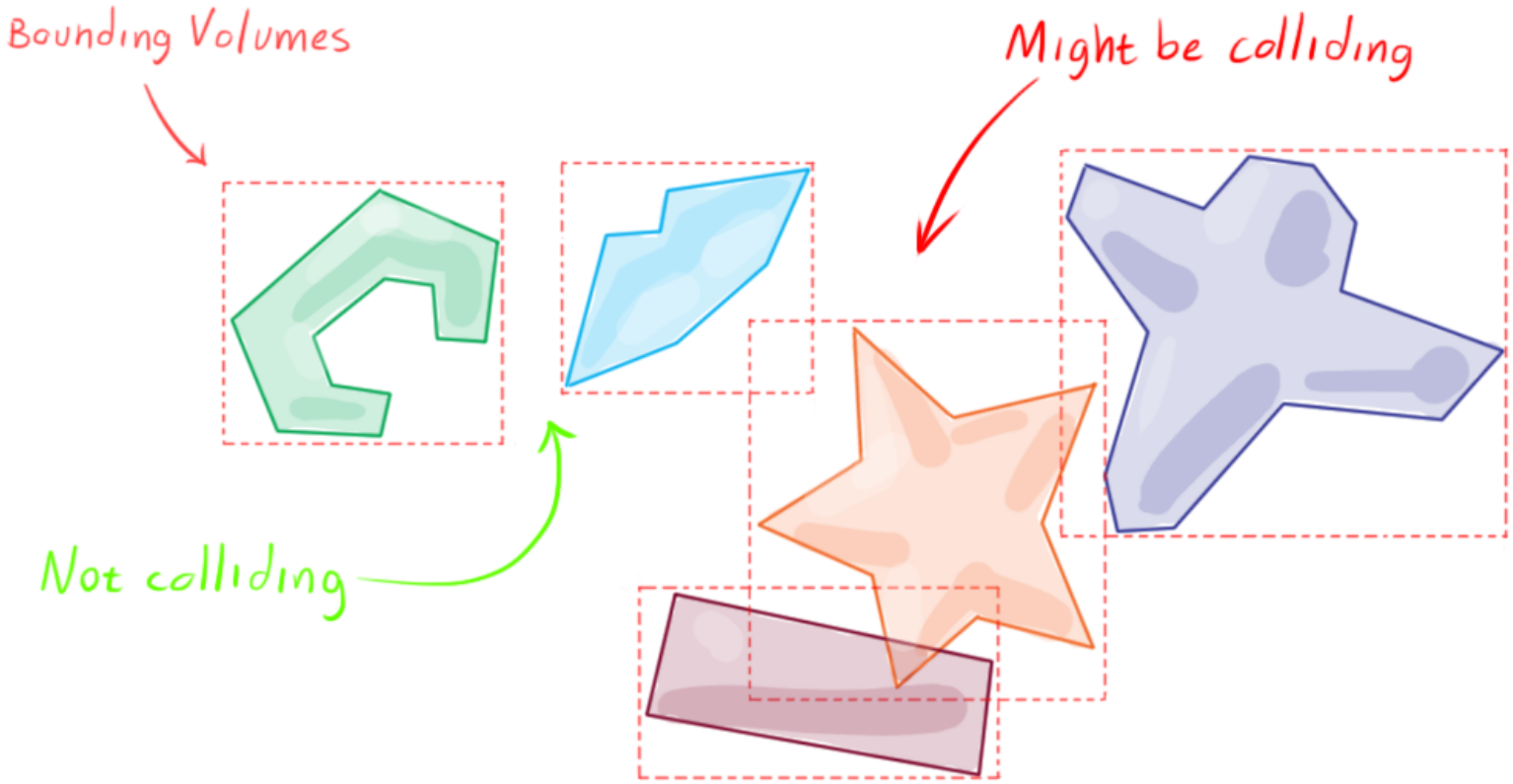


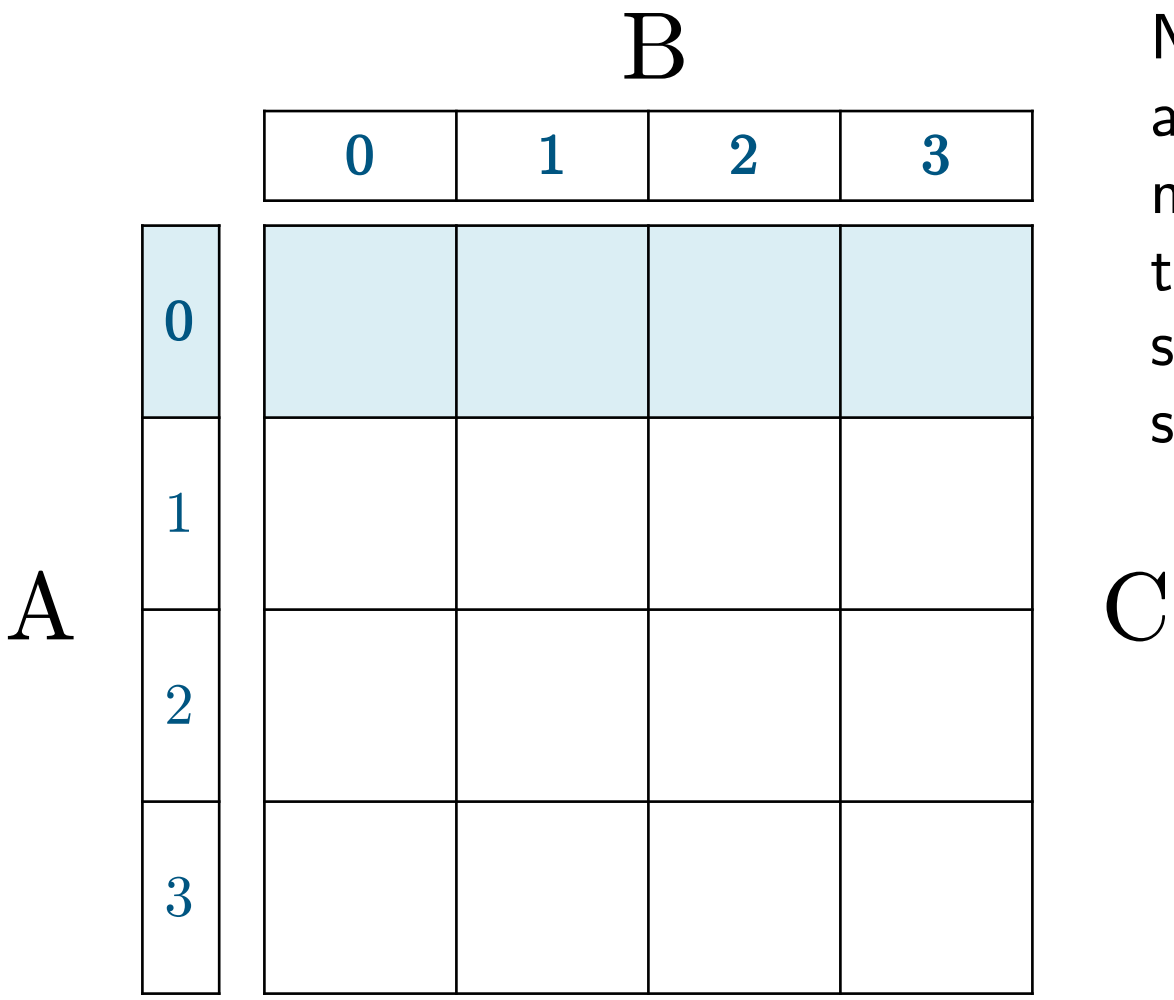
Image source:
https://www.toptal.com/game/video-game-physics-part-ii-collision-detection-for-solid-objects

$O(n^2)$ Sort

- $\text{interact}(x,\ y)$ is just $(x > y)$
- Sum of all interactions $=$ rank after sorted

	13	4	2	7	11	3	rank
13	0	1	1	1	1	1	5
4	0	0	1	0	0	1	2
2	0	0	0	0	0	0	0
7	0	1	1	0	0	1	3
11	0	1	1	1	0	1	4
3	0	0	1	0	0	0	1

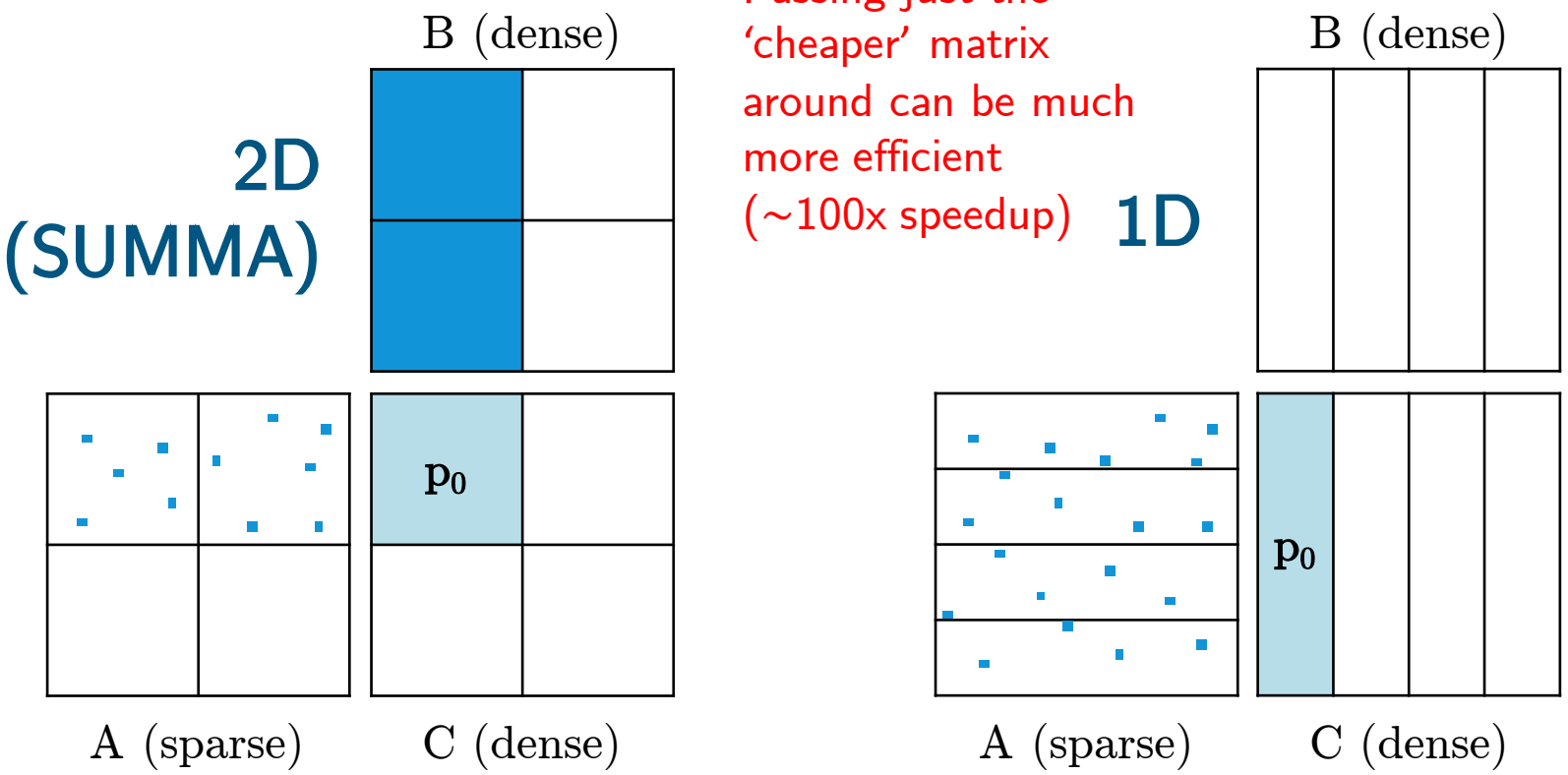
Matrix Multiplication



N-Body (1.5D) algorithm is more suitable than 2.5D for some matrix shapes.

Sparse-Dense Matmul

- Or other matmuls with unequal message sizes for $\mathrm{A}/\mathrm{B}/\mathrm{C}$. [Koanantakool et al. 2016]



Conclusions

- Using c replications reduces
  - #**messages** sent by a factor of $\mathbf{c^2}$
  - #**words** sent by a factor of $\mathbf{c}$
- $\mathbf{c}$ is a tunable parameter; $1 \leq c \leq \sqrt{p}$
- Reduced up to 99.6% communication time. Observed up to 11.8x speedup.
- Applications
  - Bottom solvers in hierarchical N-Body
  - k-way N-Body problem (42x speedup for 3-Body)
  - Database join, collision detection in CG, etc
  - Matrix-matrix multiplication (up to 100x speedup)

References I

- G. Ballard, J. Demmel, O. Holtz, and O. Schwartz.
  Minimizing communication in linear algebra.
  SIAM J. Mat. Anal. Appl., vol. 32, no. 3, 2011.
- M. Christ, J. Demmel, N. Knight, T. Scanlon, and K. A. Yelick.
  Communication lower bounds and optimal algorithms for programs that reference arrays - part 1.
  Technical Report UCB/EECS-2013-61, EECS Department, University of California, Berkeley, May 2013.
- S. Plimpton.
  Fast parallel algorithms for short-range molecular dynamics.
  J. Comput. Phys., vol. 117, no. 1, pp. 1–19, Mar. 1995. [Online]. Available: http://dx.doi.org/10.1006/jcph.1995.1039
- M. Snir.
  A note on n-body computations with cutoffs.
  Theory of Computing Systems, vol. 37, pp. 295–318, 2004.

References II

- M. Driscoll, E. Georganas, P. Koanantakool, E. Solomonik, K. Yelick.
  A Communication-Optimal N-Body Algorithm for Direct Interactions.
  In IPDPS 2013.
- P. Koanantakool, K. Yelick.
  A Computation- and Communication-Optimal Parallel Direct 3-Body Algorithm.
  In ACM/IEEE SC'14
- P. Koanantakool, A. Azad, A. Buluç, D. Morozov, S. Oh, L. Oliker, K. Yelick.
  Communication-Avoiding Parallel Sparse-Dense Matrix-Matrix Multiplication.
  In IPDPS 2016.