A Communication-Optimal N-Body Algorithm for Direct Interactions*

Penporn Koanantakool[1],
Michael Driscoll[1], Evangelos Georganas[1],
Edgar Solomonik[1], Katherine Yelick[1,2]

[1]UC Berkeley, [2]LBNL

DEGAS retreat
June 3, 2013

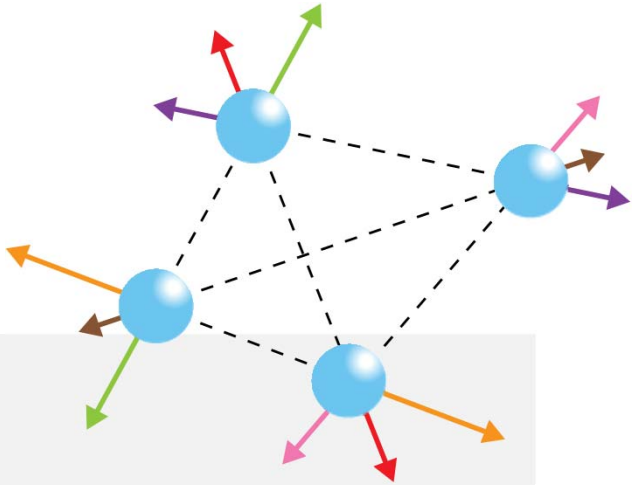
*In proceedings of IPDPS 2013 [Michael et al. 2013]

Outline

- Introduction
  - N-Body
  - Communication Lower Bounds
  - Replication to Improve Scalability
- Communication-Avoiding Algorithm
- Performance Results
  - All-pairs Interactions
  - Finite Cutoff Distance
- Conclusions

Direct N-Body

- n particles
  - Molecules, galaxies, etc.



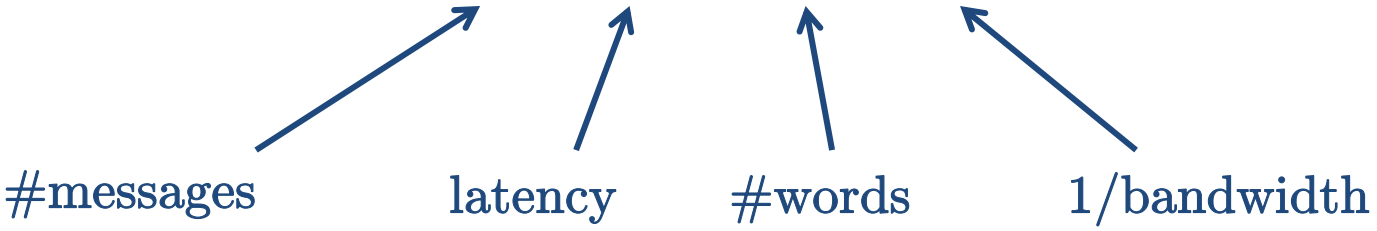
```
for t=1:timesteps
    for i=1:n
        for j=1:n
            force[i] += interact(particles[i], particles[j])
    for i=1:n
        move(particles[i], force[i])
```

- $O(n^2)$ interactions
  - Want to reduce 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
  - M size of fast memory in words
  - H max #flops with M words

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

Communication Lower Bounds

- From Minimizing Communication in Numerical Linear Algebra [Ballard et al. 2011a]:
  - F #flops per processor
  - M size of fast memory in words
  - H max #flops with M words

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

- Generalized in Communication Lower Bounds and Optimal Algorithms for Programs that Reference Arrays [Christ et al. 2013]

Lower Bounds for N-Body

- $\mathbf{n}$ particles, $\mathbf{p}$ processors

Lower Bounds for N-Body

- n particles, p processors
  - F (Flops): $\mathrm{O}(\mathrm{n}^2/\mathrm{p})$
  - #particles in fast mem: $\mathrm{O}(\mathrm{M})$
  - H (max #flops/M words): $\mathrm{O}(\mathrm{M}^2)$

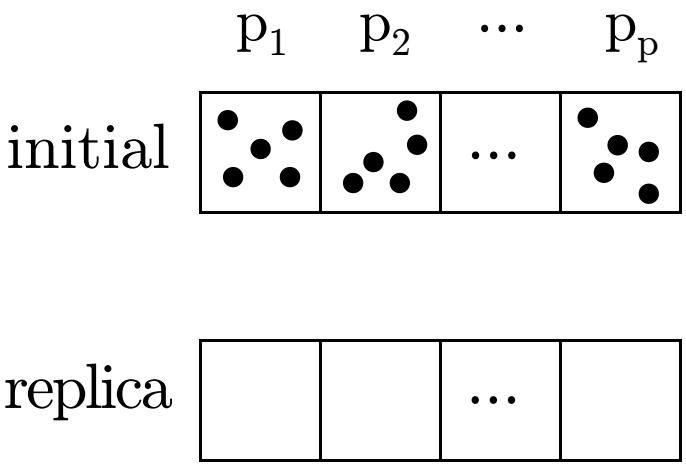
Lower Bounds for N-Body

- n particles, p processors
  - F (Flops): $\mathrm{O}(\mathrm{n}^2/\mathrm{p})$
  - #particles in fast mem: $\mathrm{O}(\mathrm{M})$
  - H (max #flops/M words): $\mathrm{O}(\mathrm{M}^2)$

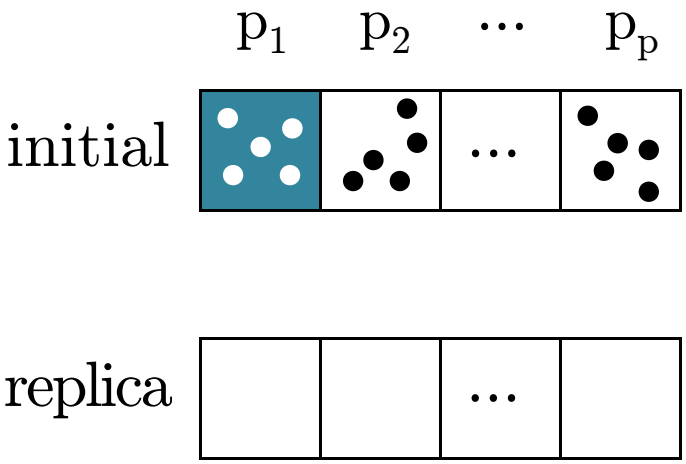
$$S_{\text{N-Body}} = \Omega\left(\frac{F}{H}\right) = \Omega\left(\frac{n^2/p}{M^2}\right)$$

$$W_{\text{N-Body}} = \Omega\left(\frac{F \cdot M}{H}\right) = \Omega\left(\frac{n^2/p}{M}\right)$$

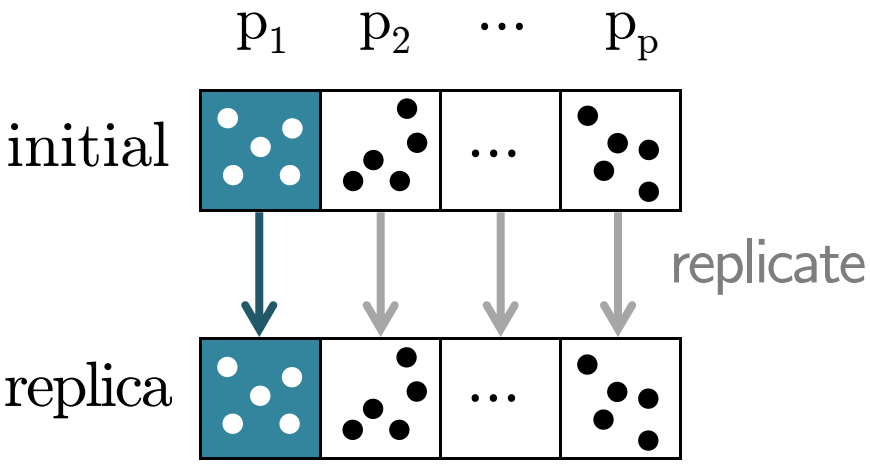
Naïve All-Pairs Interaction Algorithm



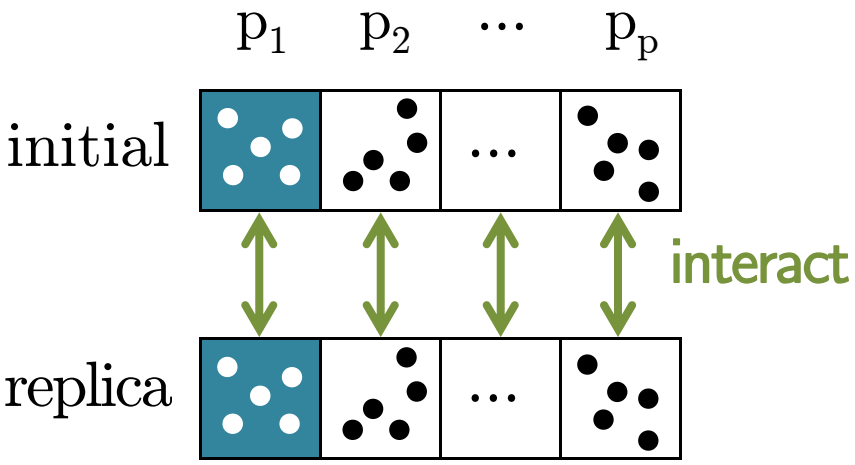
Naïve All-Pairs Interaction Algorithm



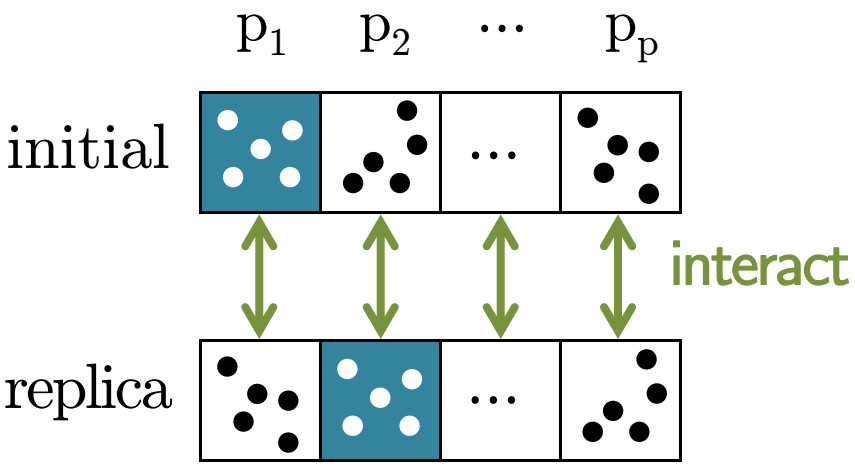
Naïve All-Pairs Interaction Algorithm



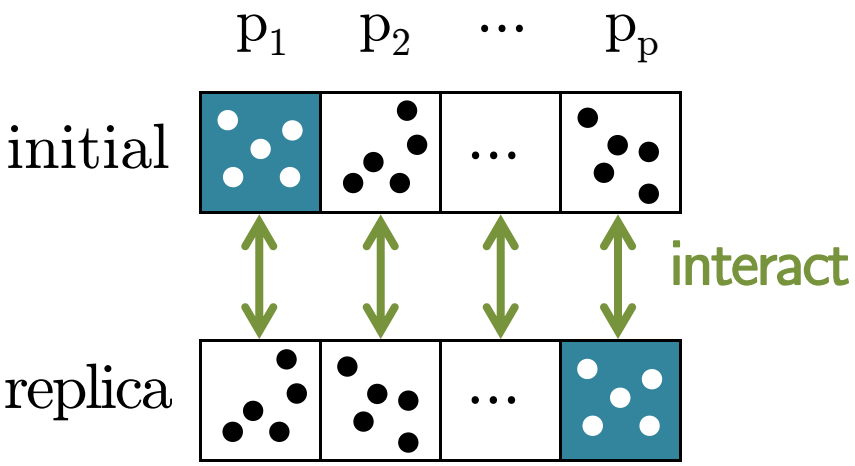
Naïve All-Pairs Interaction Algorithm



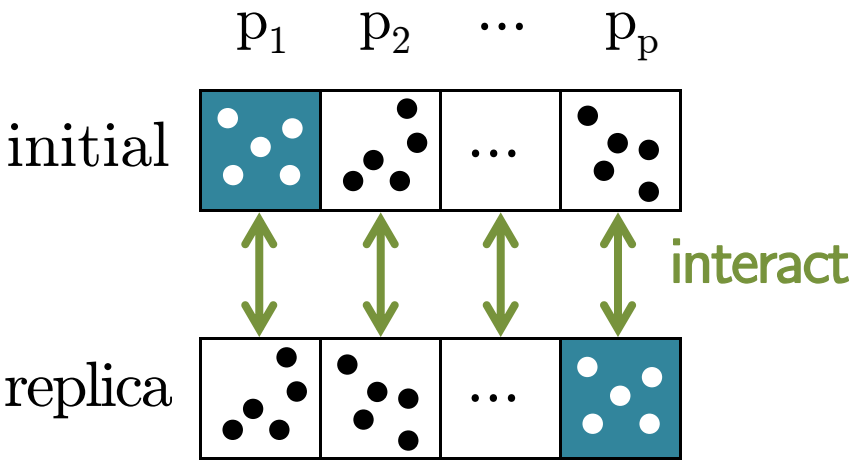
Naïve All-Pairs Interaction Algorithm



Naïve All-Pairs Interaction Algorithm

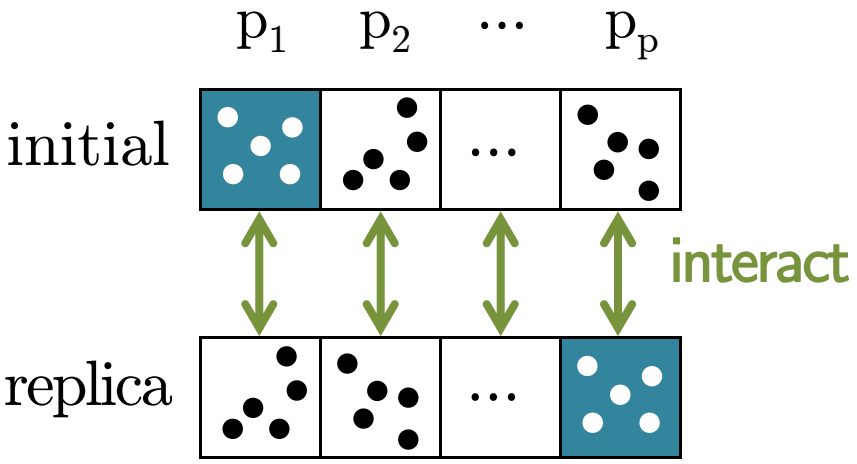


Naïve All-Pairs Interaction Algorithm



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

Naïve All-Pairs Interaction Algorithm



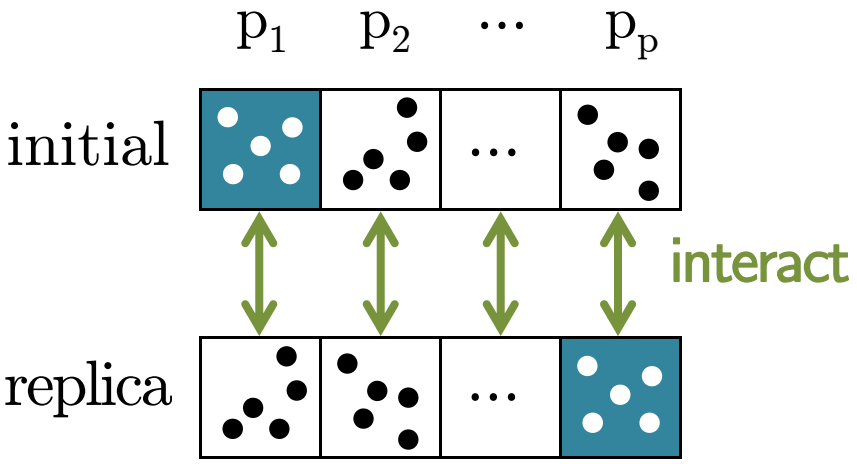
message size: $\mathrm{O(n/p)}$
#messages: $\mathrm{O(p)}$
#words sent: $\mathrm{O(n)}$

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

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

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

Naïve All-Pairs Interaction Algorithm



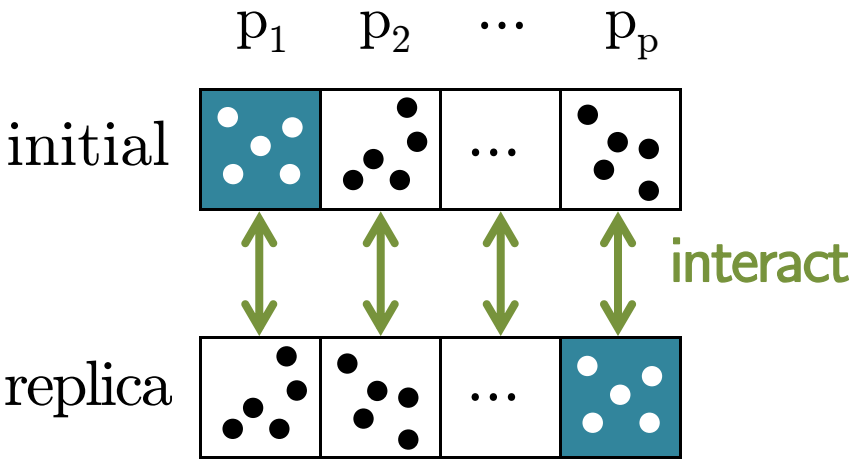
message size: $O(n/p)$
#messages: $O(p)$
#words sent: $O(n)$

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

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

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

Naïve All-Pairs Interaction Algorithm



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

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

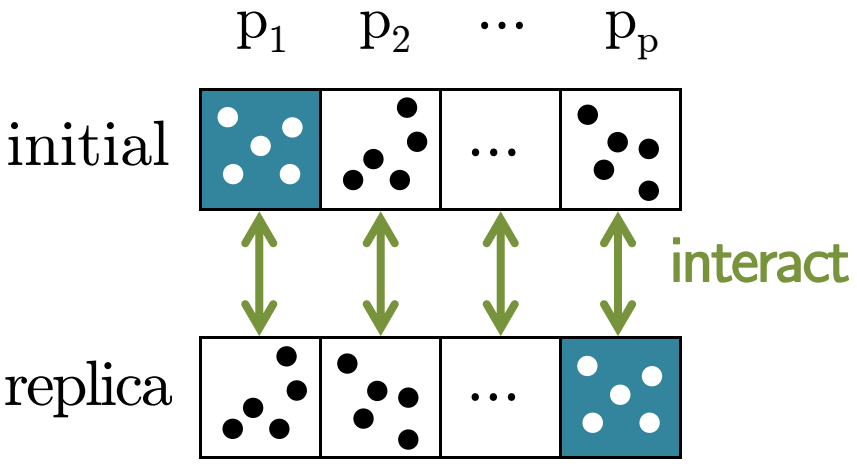
#words sent: $\mathrm{O(n)}$

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

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

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

Naïve All-Pairs Interaction Algorithm



message size: $O(n/p)$

#messages: $O(p)$

#words sent: $O(n)$

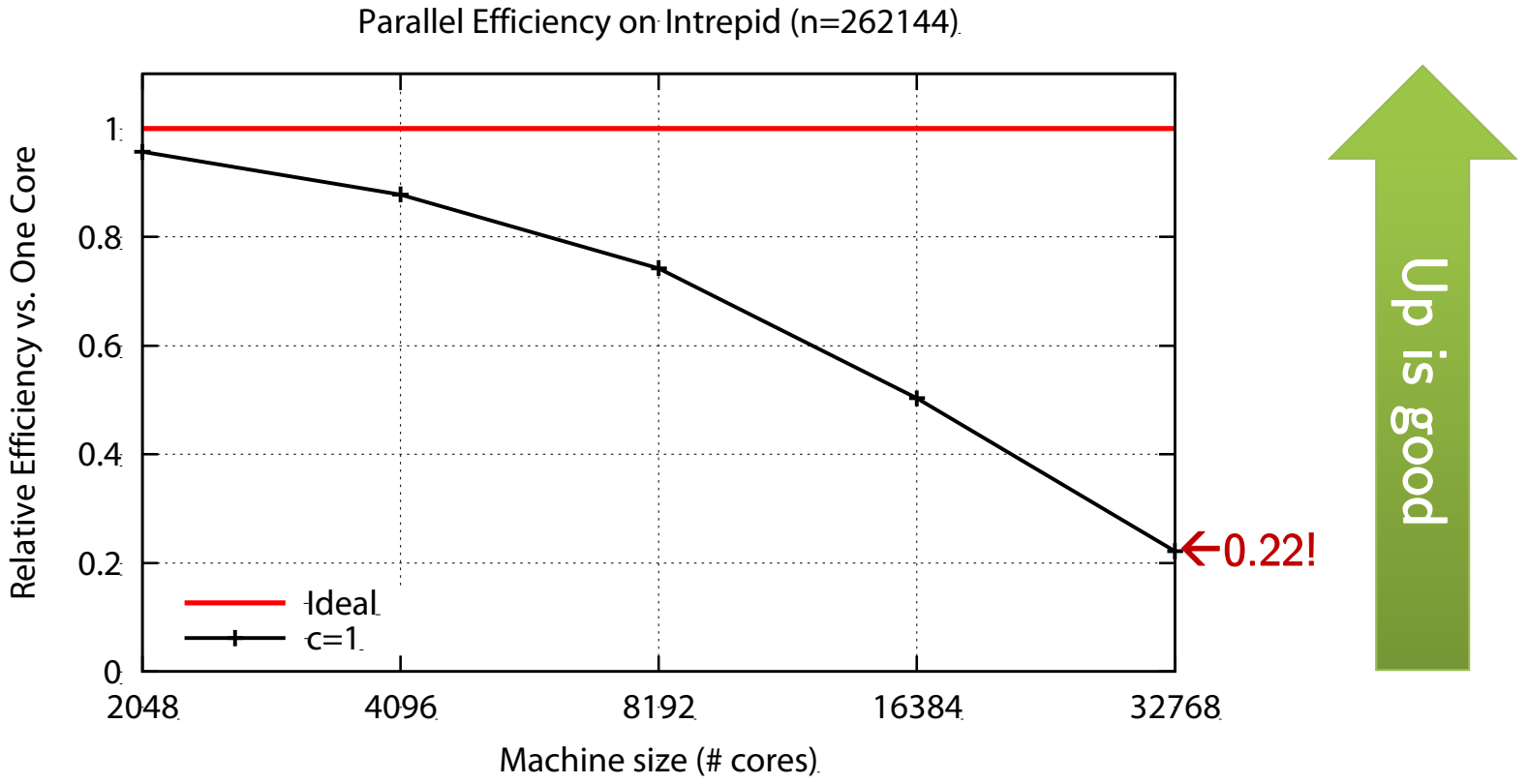
- Lower Bounds for $M=O(n/p)$

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

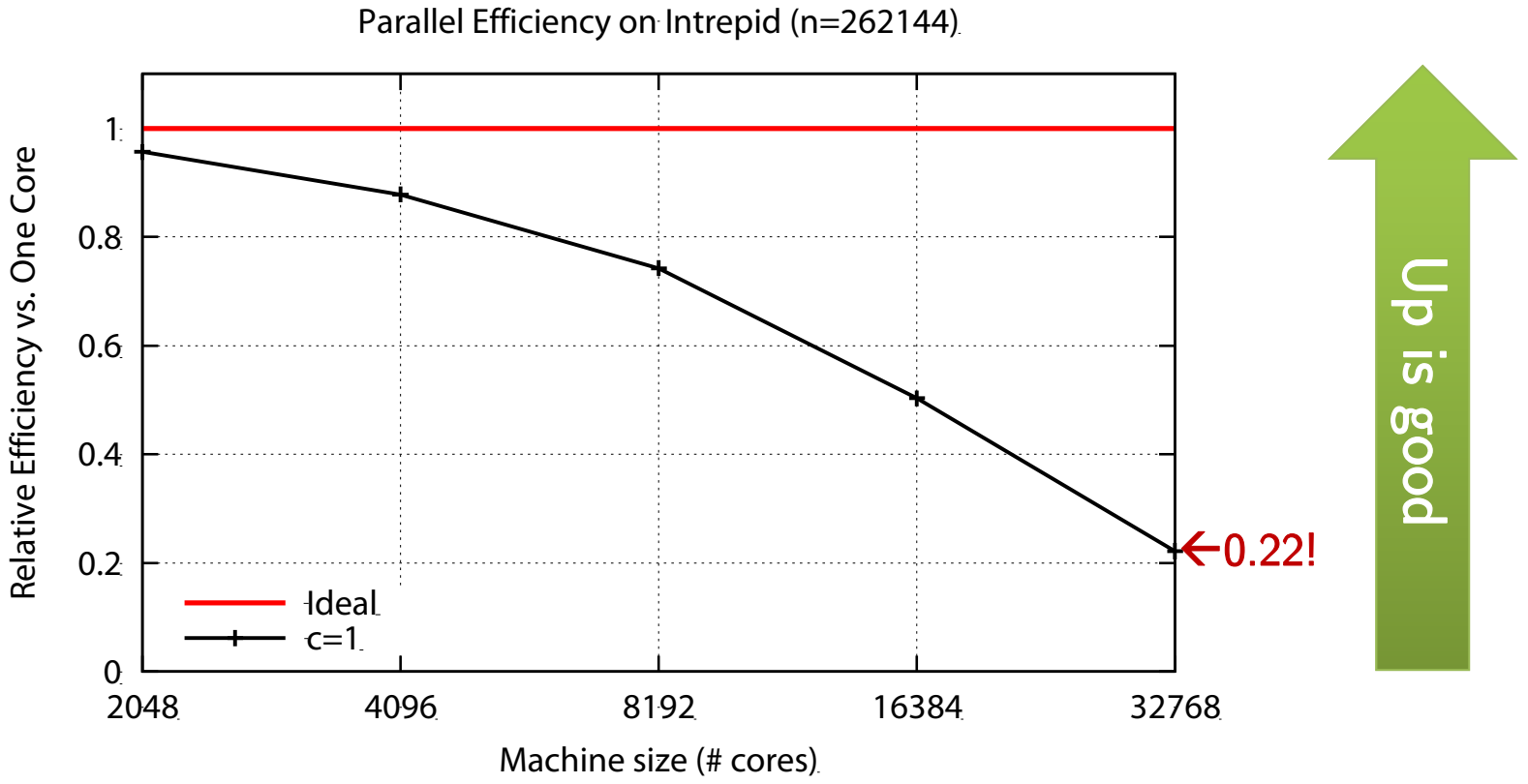
$$W_{\text{allpairs}} = \Omega\left(\frac{n^2/p}{M}\right) = \Omega\left(\frac{n^2/p}{n/p}\right) = \Omega\left(n\right)$$

- Communication-optimal for $M=O(n/p)$

Bad Scalability



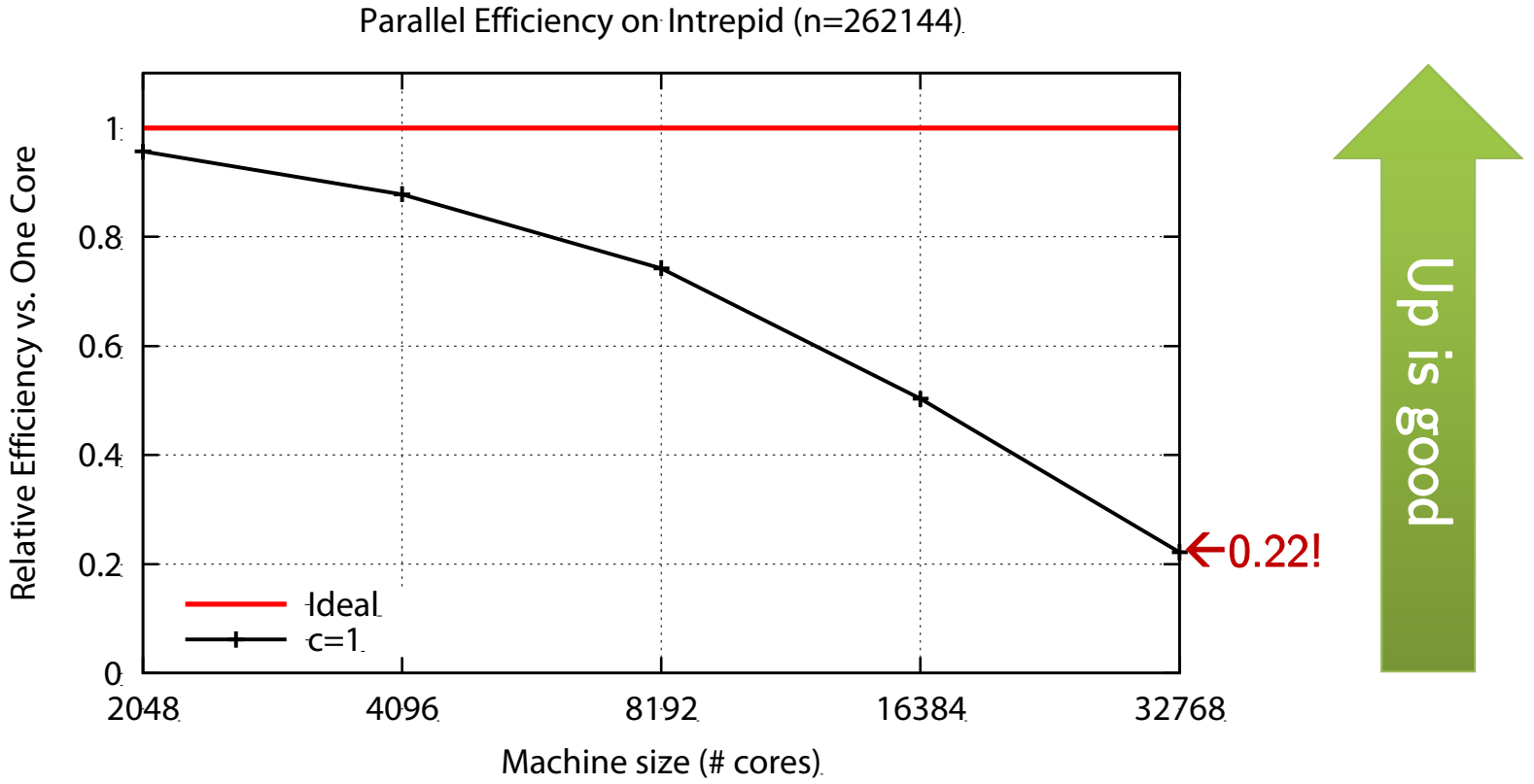
Bad Scalability



- We can do better

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

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 $M_{\mathrm{CA}} = O\left(c \cdot \frac{n}{p}\right) = c \cdot M_{\mathrm{allpairs}}$

Exploiting Extra Memory

- By making c replicas $M_{\mathrm{CA}} = O\left(c \cdot \frac{n}{p}\right) = c \cdot M_{\mathrm{allpairs}}$
- We get $S_{\mathrm{CA}} = \Omega\left(\frac{F}{M_{\mathrm{CA}}^2}\right)$

$$W_{\mathrm{CA}} = \Omega\left(\frac{F}{M_{\mathrm{CA}}}\right)$$

Exploiting Extra Memory

- By making c replicas $M_{\mathrm{CA}} = O\left(c \cdot \frac{n}{p}\right) = c \cdot M_{\mathrm{allpairs}}$
- We get

$$S_{\mathrm{CA}} = \Omega\left(\frac{F}{M_{\mathrm{CA}}^2}\right) = \Omega\left(\frac{F}{c^2 \cdot M_{\mathrm{allpairs}}^2}\right) = \frac{1}{c^2} \cdot S_{\mathrm{allpairs}}$$

$$W_{\mathrm{CA}} = \Omega\left(\frac{F}{M_{\mathrm{CA}}}\right) = \Omega\left(\frac{F}{c \cdot M_{\mathrm{allpairs}}}\right) = \frac{1}{c} \cdot W_{\mathrm{allpairs}}$$

Exploiting Extra Memory

- By making c replicas $M_{\mathrm{CA}} = O\left(c \cdot \frac{n}{p}\right) = c \cdot M_{\mathrm{allpairs}}$
- We get

$$S_{\mathrm{CA}} = \Omega\left(\frac{F}{M_{\mathrm{CA}}^2}\right) = \Omega\left(\frac{F}{c^2 \cdot M_{\mathrm{allpairs}}^2}\right) = \frac{1}{c^2} \cdot S_{\mathrm{allpairs}}$$

$$W_{\mathrm{CA}} = \Omega\left(\frac{F}{M_{\mathrm{CA}}}\right) = \Omega\left(\frac{F}{c \cdot M_{\mathrm{allpairs}}}\right) = \frac{1}{c} \cdot W_{\mathrm{allpairs}}$$

- Existing algorithms only use $c = \sqrt{p}$
  - [Plimpton 1995], [Snir 2004], etc.

Exploiting Extra Memory

- By making c replicas $M_{\text{CA}} = O\left(c \cdot \frac{n}{p}\right) = c \cdot M_{\text{allpairs}}$
- We get

$$S_{\text{CA}} = \Omega\left(\frac{F}{M_{\text{CA}}^2}\right) = \Omega\left(\frac{F}{c^2 \cdot M_{\text{allpairs}}^2}\right) = \frac{1}{c^2} \cdot S_{\text{allpairs}}$$

$$W_{\text{CA}} = \Omega\left(\frac{F}{M_{\text{CA}}}\right) = \Omega\left(\frac{F}{c \cdot M_{\text{allpairs}}}\right) = \frac{1}{c} \cdot W_{\text{allpairs}}$$

- Existing algorithms only use $c = \sqrt{p}$
  - [Plimpton 1995], [Snir 2004], etc.
- Our CA algorithm allows $1 \leq c \leq \sqrt{p}$ *

*Formal proof in HBL paper

Exploiting Extra Memory

- By making c replicas $M_{\text{CA}} = O\left(c \cdot \frac{n}{p}\right) = c \cdot M_{\text{allpairs}}$
- We get

$$S_{\text{CA}} = \Omega\left(\frac{F}{M_{\text{CA}}^2}\right) = \Omega\left(\frac{F}{c^2 \cdot M_{\text{allpairs}}^2}\right) = \frac{1}{c^2} \cdot S_{\text{allpairs}}$$

$$W_{\text{CA}} = \Omega\left(\frac{F}{M_{\text{CA}}}\right) = \Omega\left(\frac{F}{c \cdot M_{\text{allpairs}}}\right) = \frac{1}{c} \cdot W_{\text{allpairs}}$$

- Existing algorithms only use $c = \sqrt{p}$
  - [Plimpton 1995], [Snir 2004], etc.
- Our CA algorithm allows $1 \leq c \leq \sqrt{p}$ *
  - Support machines with limited memory

*Formal proof in HBL paper

Exploiting Extra Memory

- By making c replicas $M_{\text{CA}} = O\left(c \cdot \frac{n}{p}\right) = c \cdot M_{\text{allpairs}}$
- We get $S_{\text{CA}} = \Omega\left(\frac{F}{M_{\text{CA}}^2}\right) = \Omega\left(\frac{F}{c^2 \cdot M_{\text{allpairs}}^2}\right) = \frac{1}{c^2} \cdot S_{\text{allpairs}}$

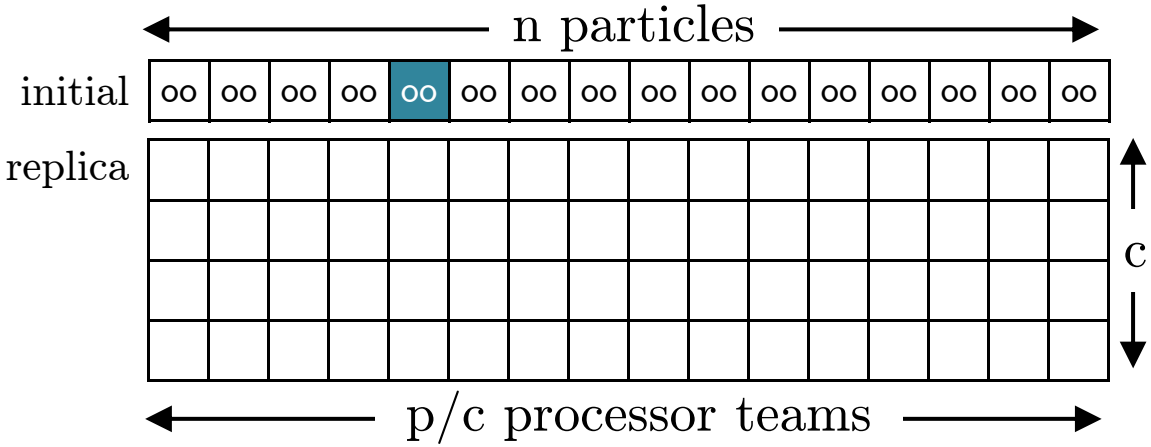
$$W_{\text{CA}} = \Omega\left(\frac{F}{M_{\text{CA}}}\right) = \Omega\left(\frac{F}{c \cdot M_{\text{allpairs}}}\right) = \frac{1}{c} \cdot W_{\text{allpairs}}$$

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

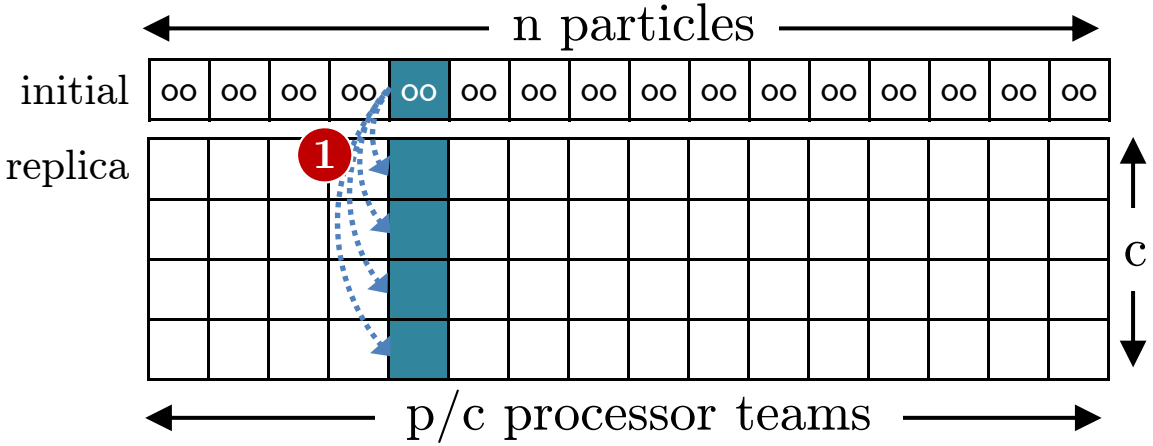
*Formal proof in HBL paper

The Communication-Avoiding All-pairs Interaction Algorithm

The CA-All-Pairs Algorithm

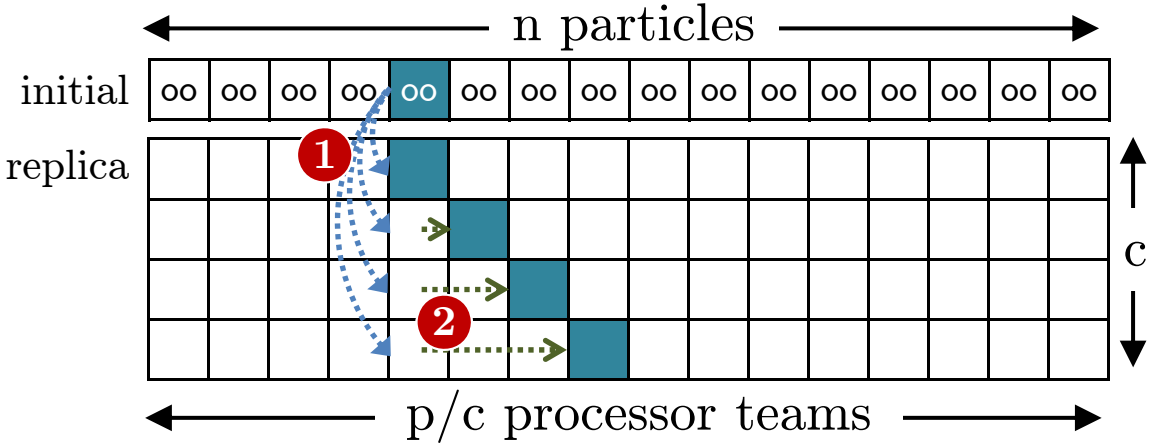


The CA-All-Pairs Algorithm



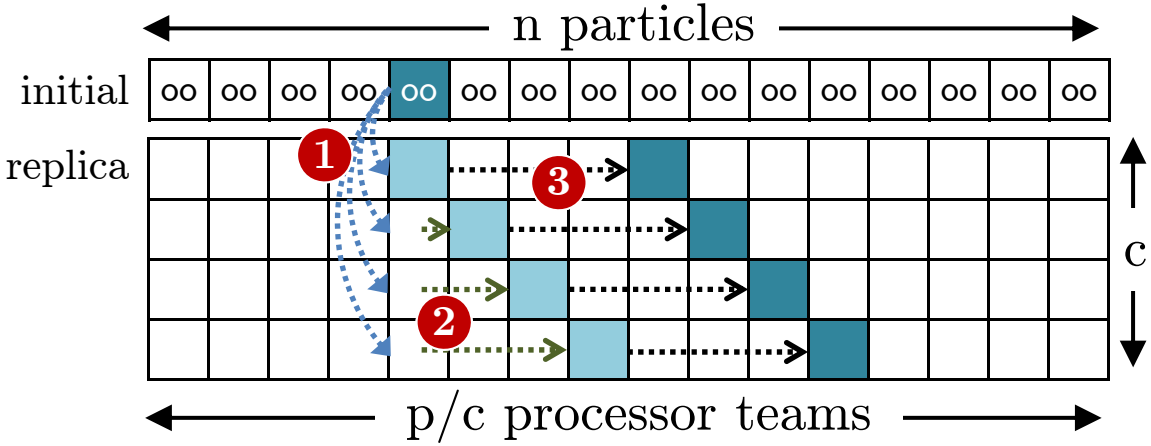
1. Broadcast to team members

The CA-All-Pairs Algorithm



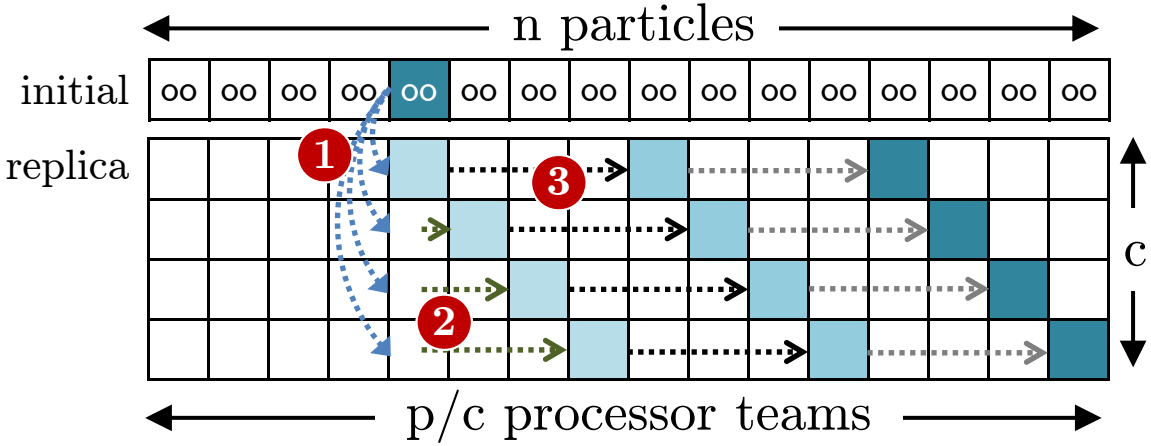
2. Shift by row ID,
   and calculate interactions

The CA-All-Pairs Algorithm



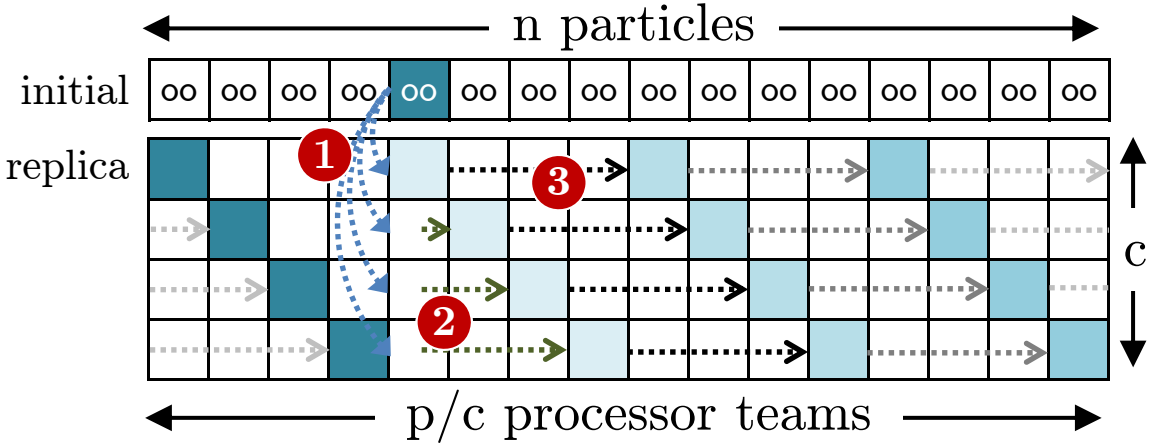
3. Shift by c,
   and calculate interactions

The CA-All-Pairs Algorithm



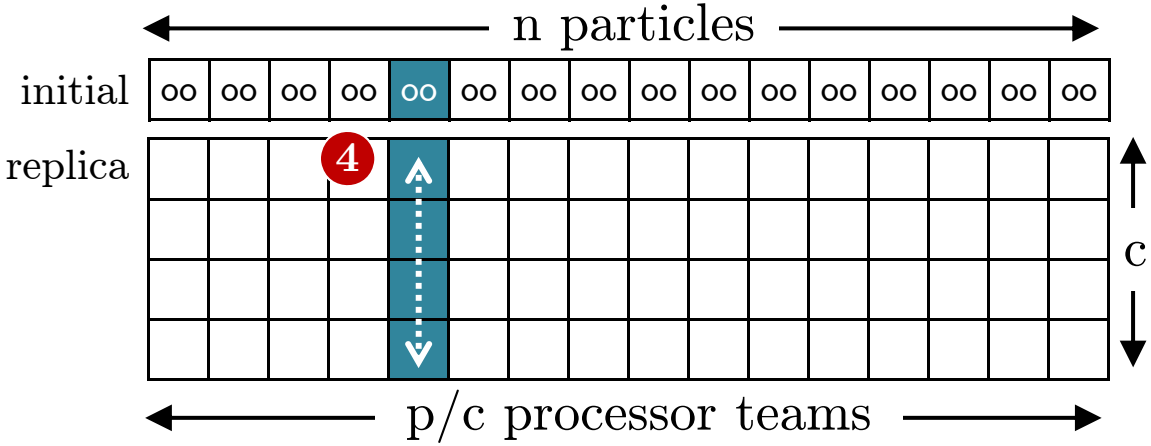
3. Shift by c,
   and calculate interactions

The CA-All-Pairs Algorithm



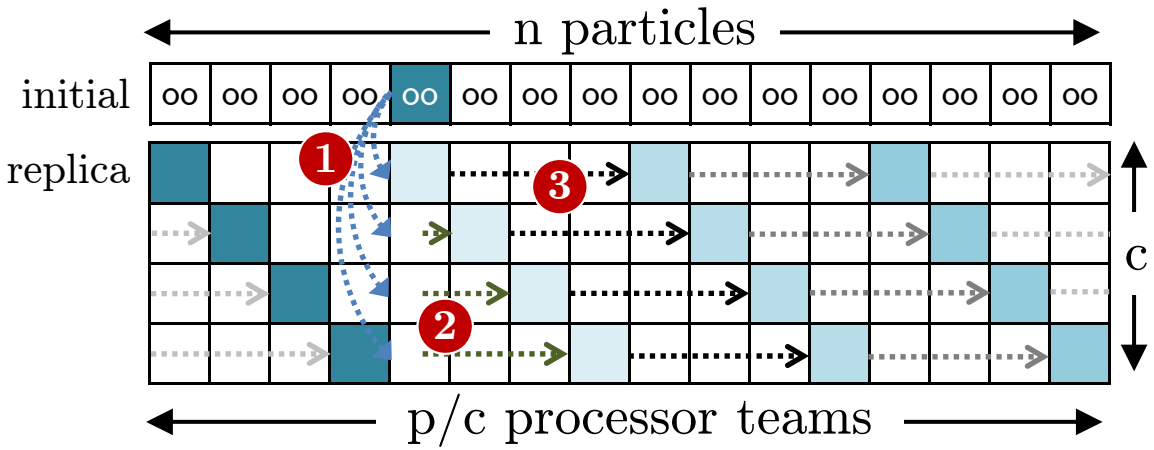
3. Shift by c,
   and calculate interactions

The CA-All-Pairs Algorithm



4. Reduce all interactions calculated by all team members

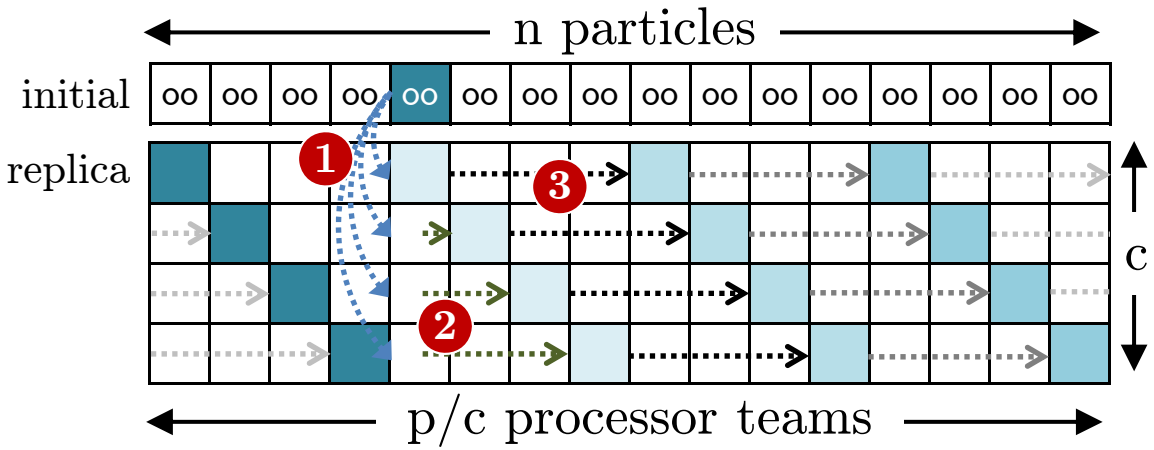
Communication Optimality



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

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

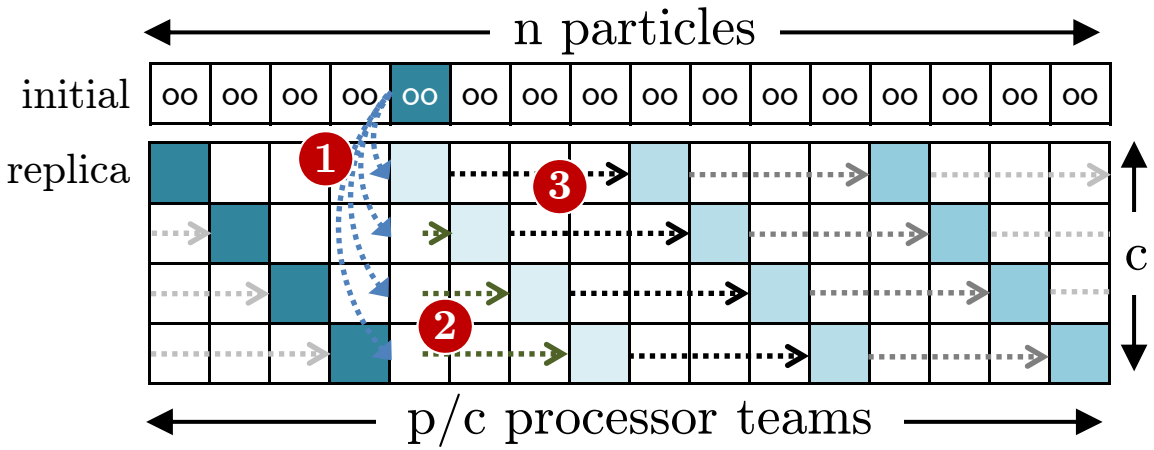
Communication Optimality



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

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

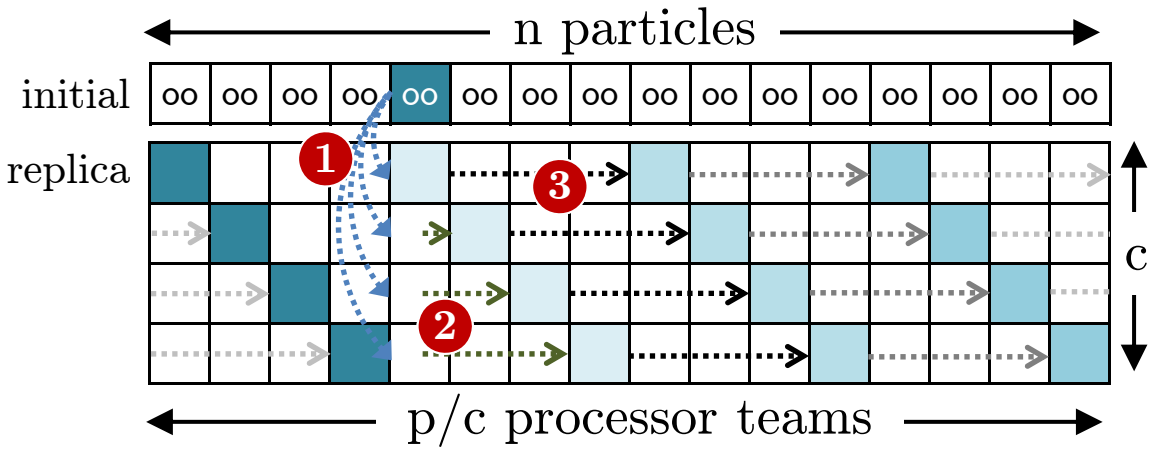
Communication Optimality



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

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

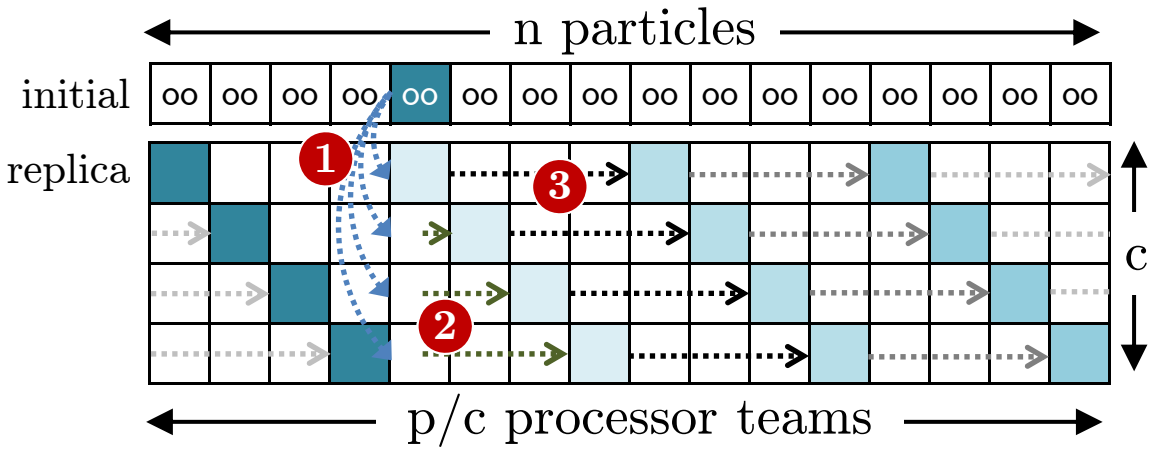
Communication Optimality



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

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

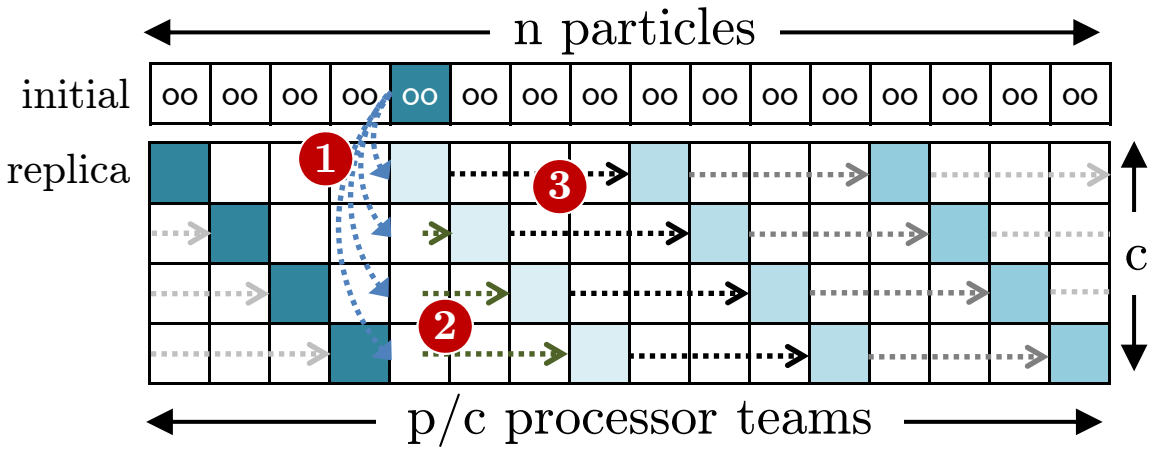
Communication Optimality



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

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

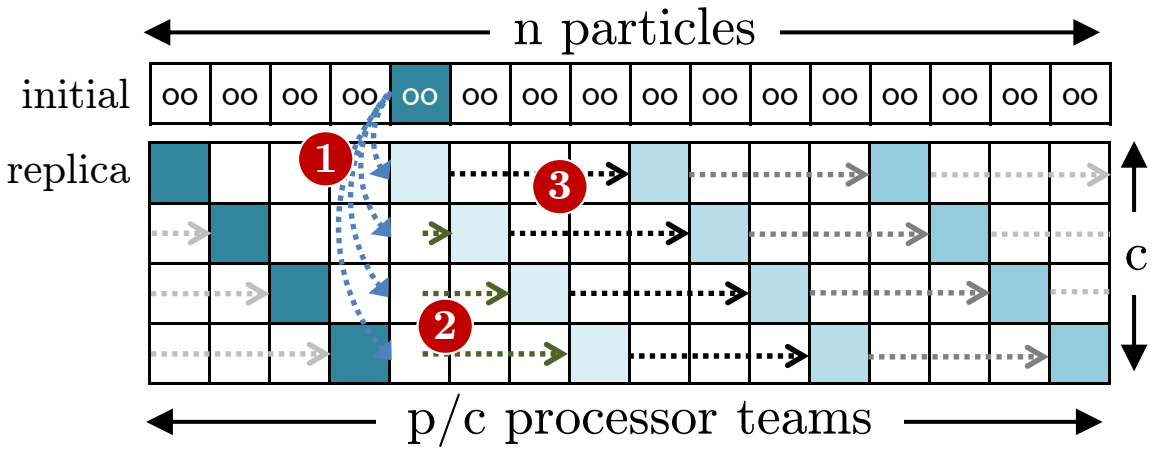
Communication Optimality



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

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

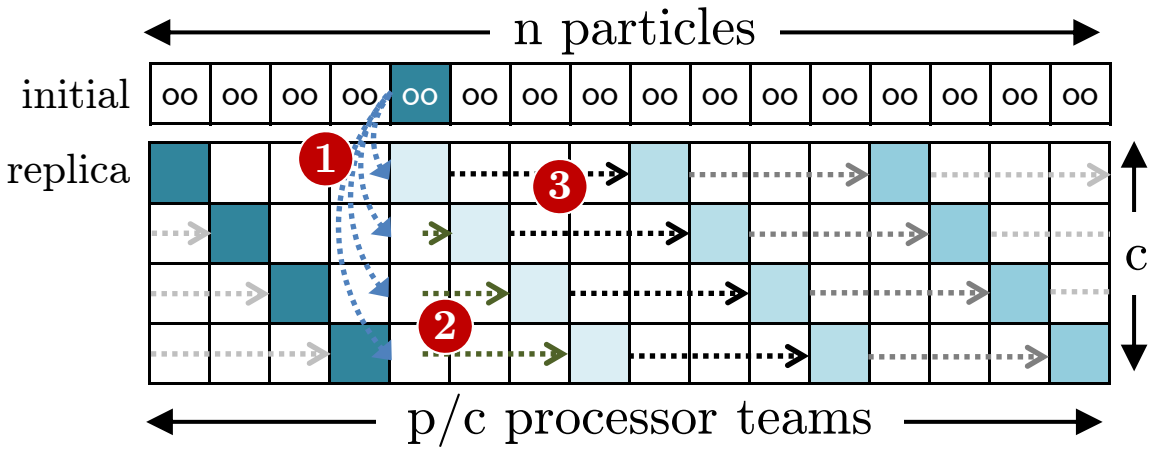
Communication Optimality



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

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

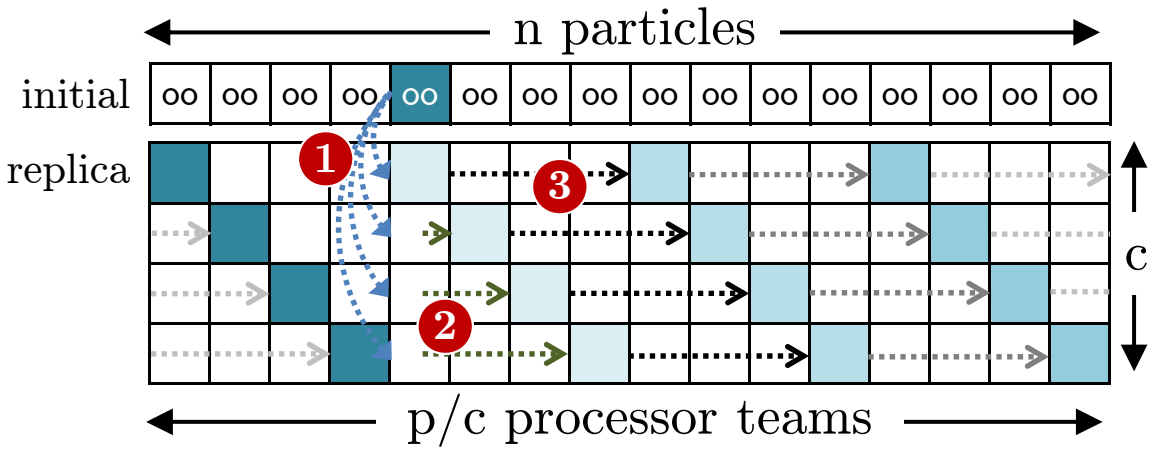
Communication Optimality



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

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

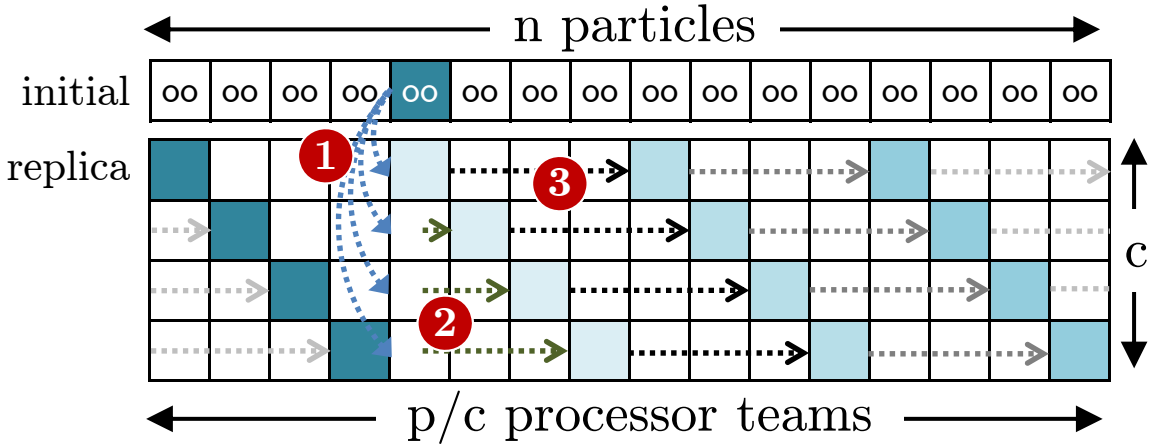
Communication Optimality



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

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

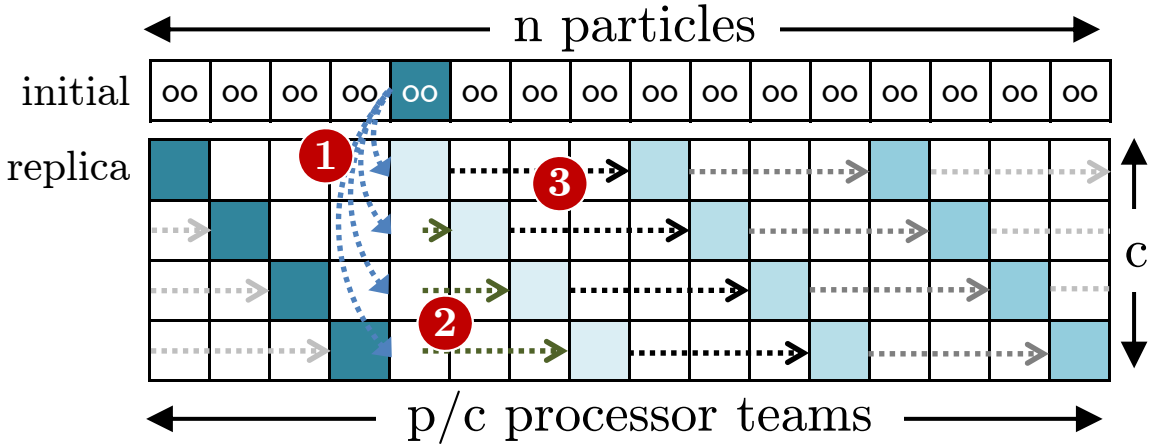
Communication Optimality



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

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

Communication Optimality



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

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

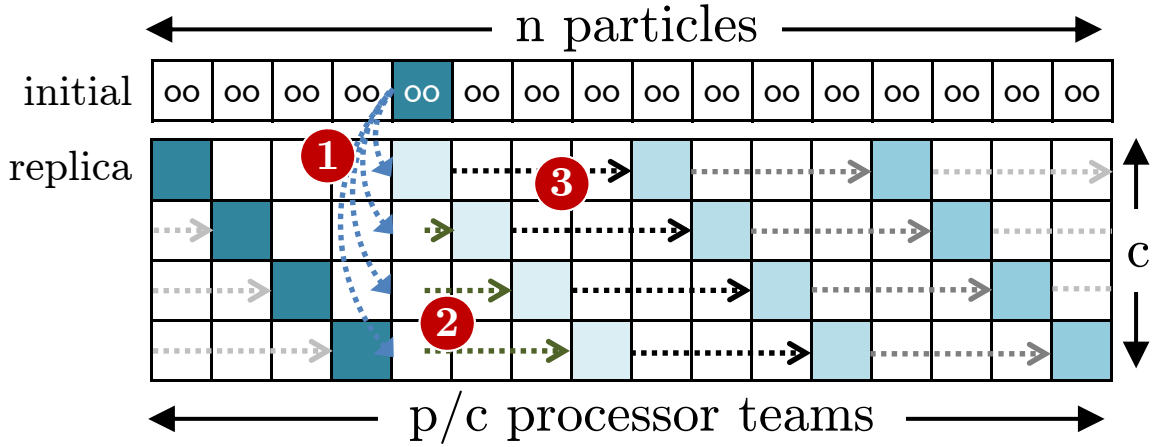
Dominant factor

Bounds

$$S_{\mathrm{CA}} = \frac{1}{c^2} \cdot S_{\mathrm{allpairs}}$$

$$W_{\mathrm{CA}} = \frac{1}{c} \cdot W_{\mathrm{allpairs}}$$

Communication Optimality



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

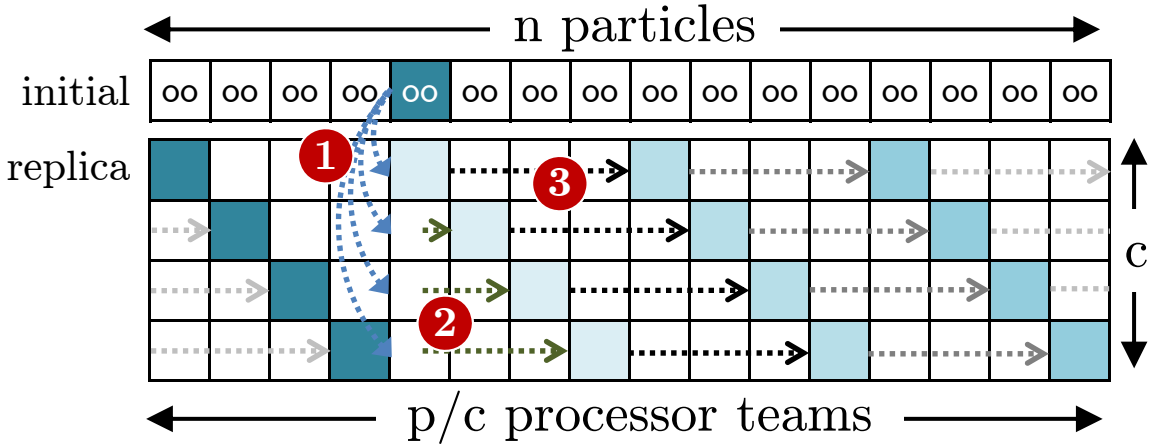
	Message Size	#messages (S)	#words (W)
1. Broadcast	$O\left(\frac{n}{p/c}\right)$	$O(\log c)$	$O\left(\frac{nc\log c}{p}\right)$
2. Skew	$O\left(\frac{n}{p/c}\right)$	$O(1)$	$O\left(\frac{nc}{p}\right)$
3. Shift	$O\left(\frac{n}{p/c}\right)$	$O\left(\frac{p/c}{c}\right) = \boldsymbol{O\left(\frac{p}{c^2}\right)}$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

Dominant factor

Bounds

$$S_{\text{CA}} = \frac{1}{c^2}\cdot S_{\text{allpairs}}$$
$$= \frac{1}{c^2}\cdot\Omega(p) = \Omega\left(\frac{p}{c^2}\right)$$
$$W_{\text{CA}} = \frac{1}{c}\cdot W_{\text{allpairs}}$$

Communication Optimality



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

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

Bounds

$$S_{\text{CA}} = \frac{1}{c^2}\cdot S_{\text{allpairs}} = \frac{1}{c^2}\cdot\Omega(p) = \Omega\left(\frac{p}{c^2}\right)$$

$$W_{\text{CA}} = \frac{1}{c}\cdot W_{\text{allpairs}} = \frac{1}{c}\cdot\Omega(n) = \Omega\left(\frac{n}{c}\right)$$

Test Environment

- N-Body code
  - Flat MPI
  - 56-byte particles
  - Repulsive force $\propto 1/\mathrm{r}^2$
  - Reflective boundary conditions

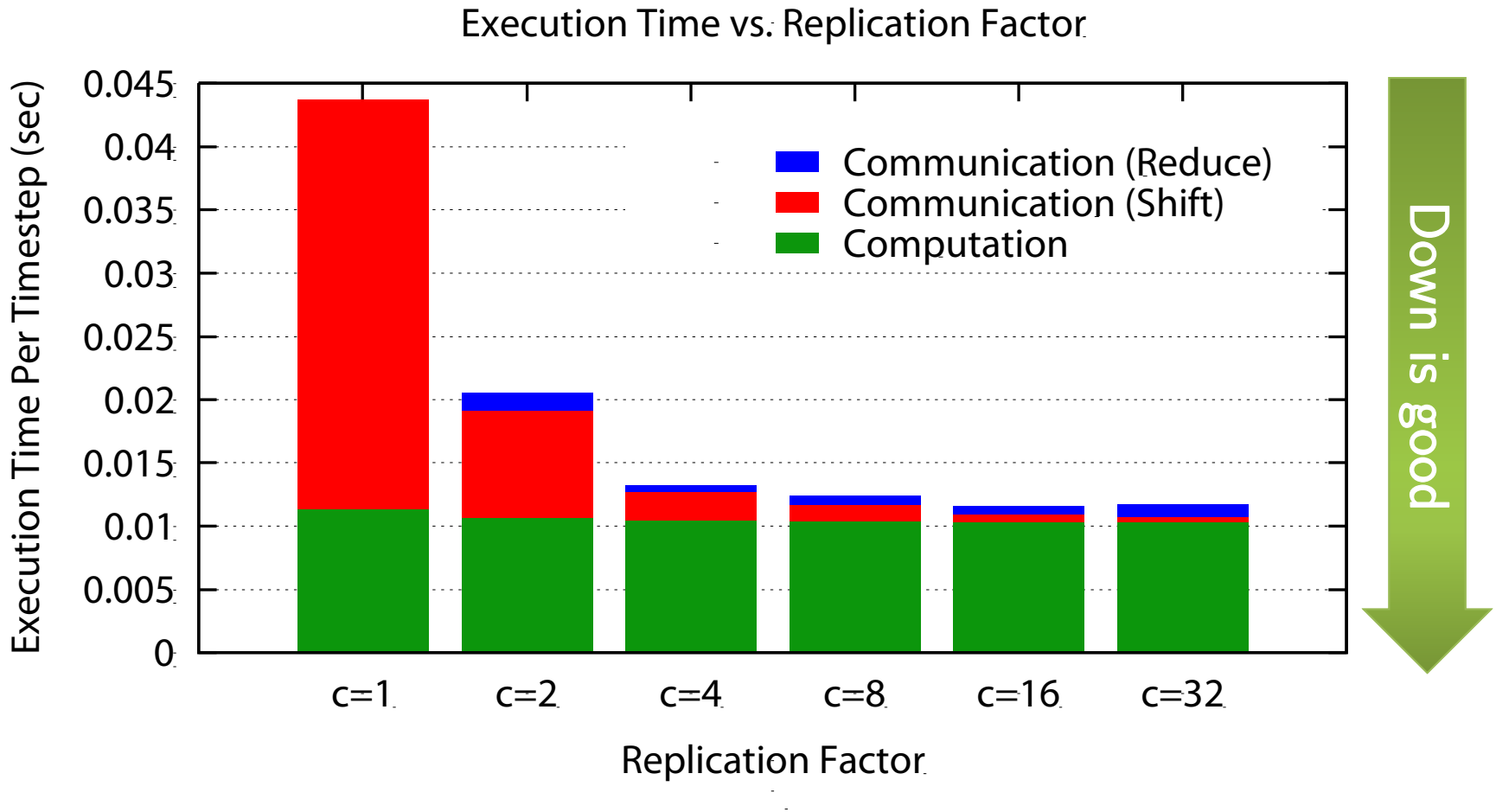
Test Environment

- N-Body code
  - Flat MPI
  - 56-byte particles
  - Repulsive force $\propto 1/r^2$
  - 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

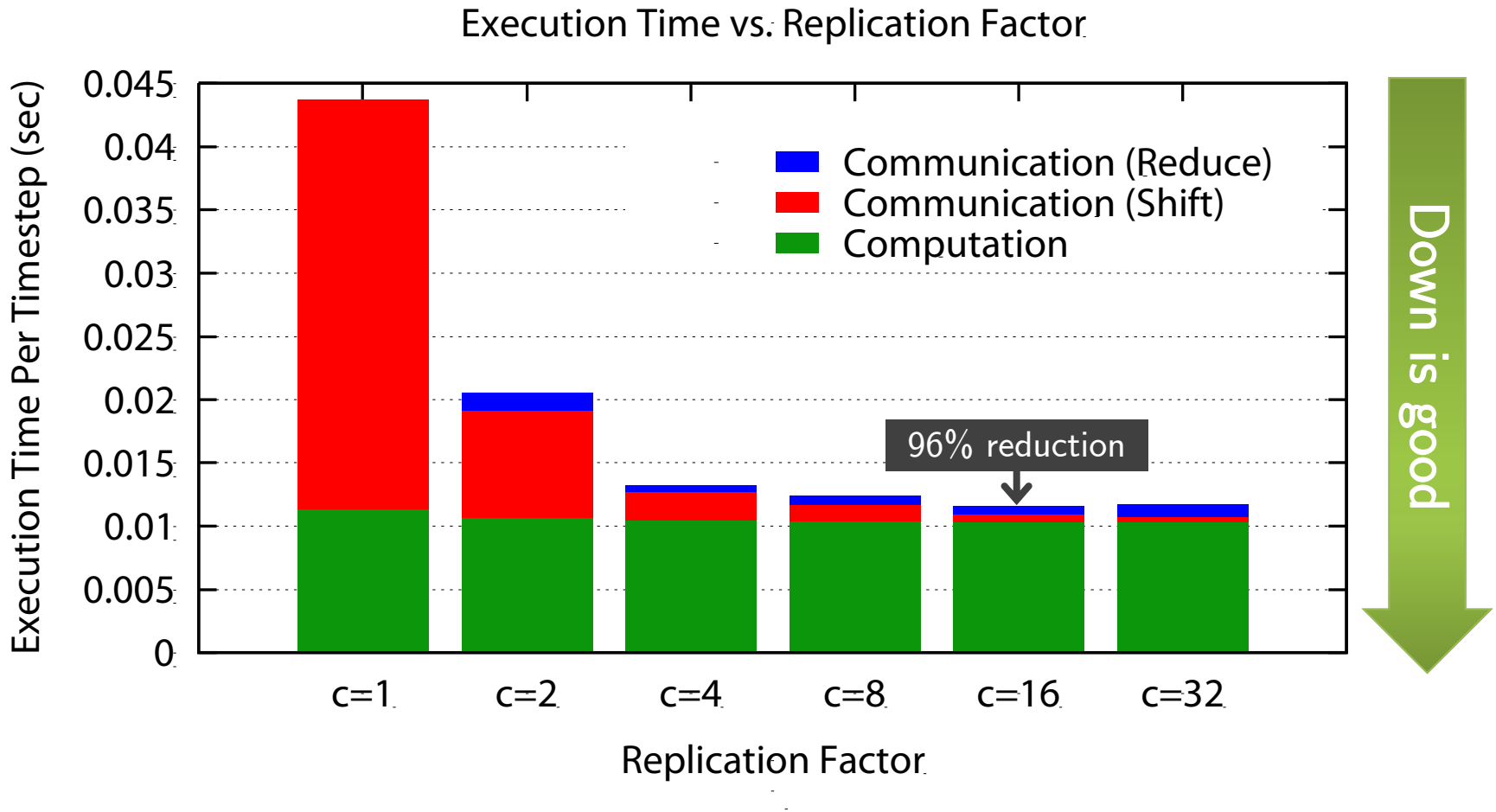
Communication Decreases!

- Hopper; $n=24K$ particles, $p=6K$ cores



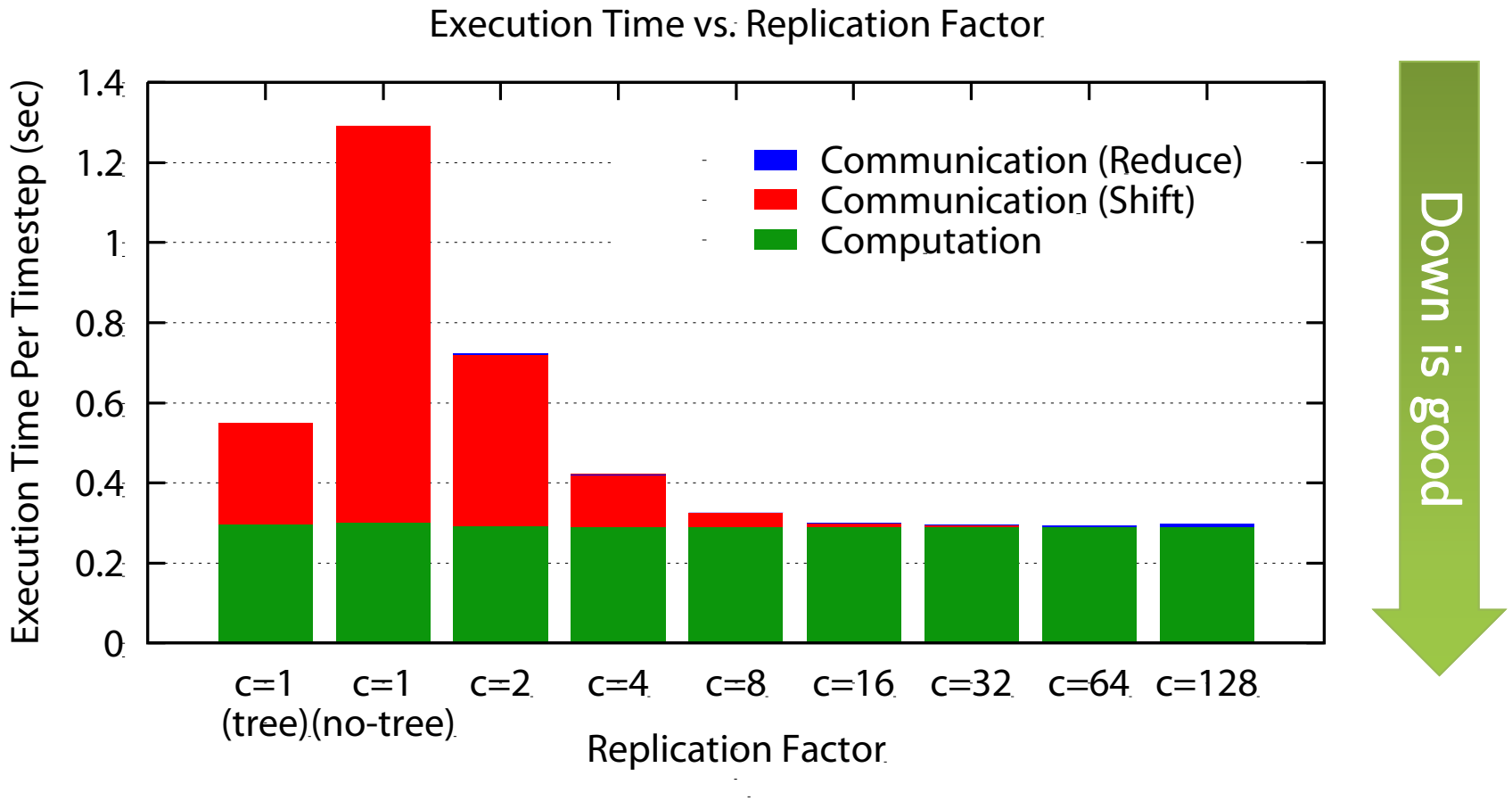
Communication Decreases!

- Hopper; $n=24K$ particles, $p=6K$ cores



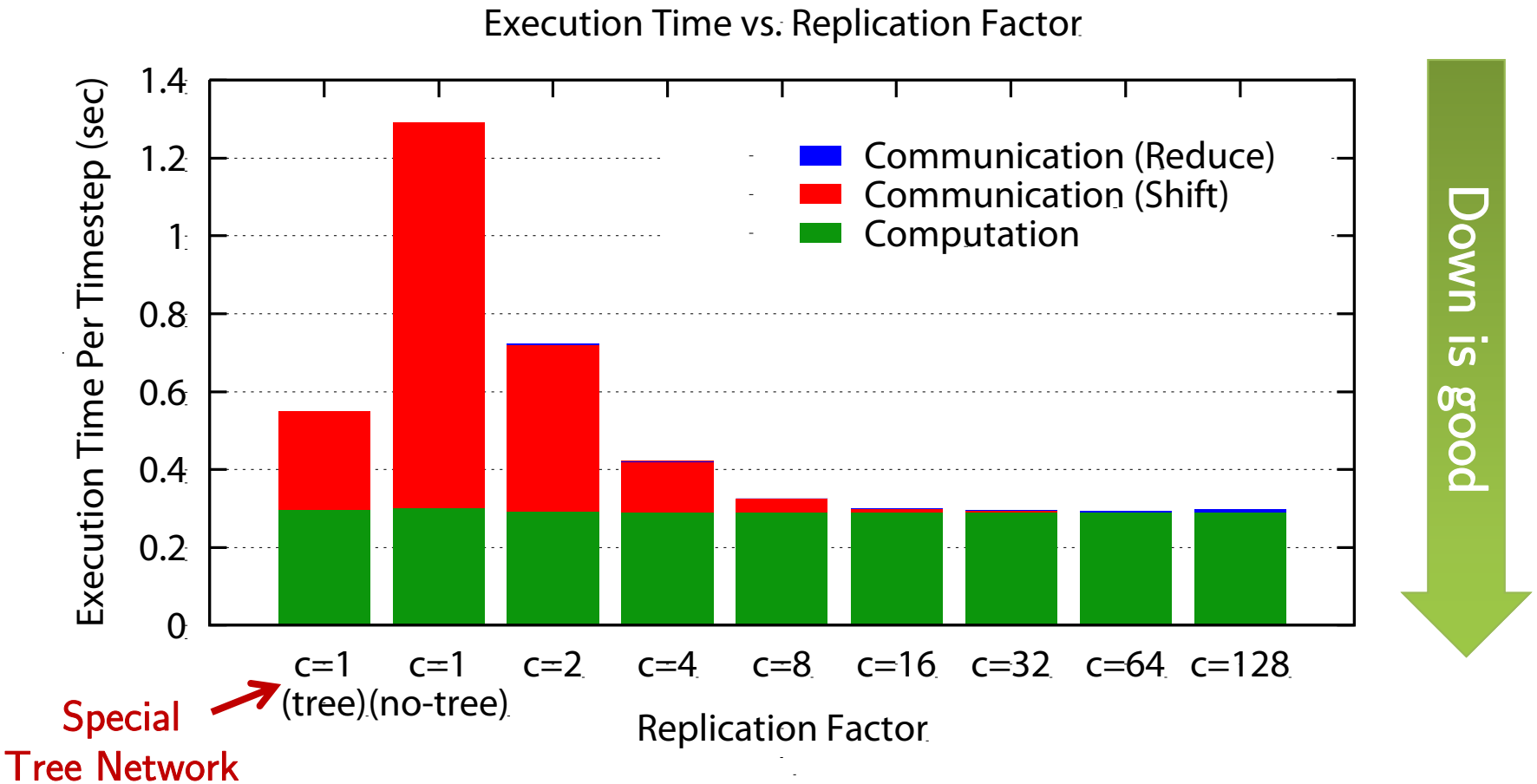
Often, $\sqrt{p}$ isn't the best c

- Intrepid; $\mathrm{n}=262\mathrm{K}$ particles, $\mathrm{p}=32\mathrm{K}$ cores



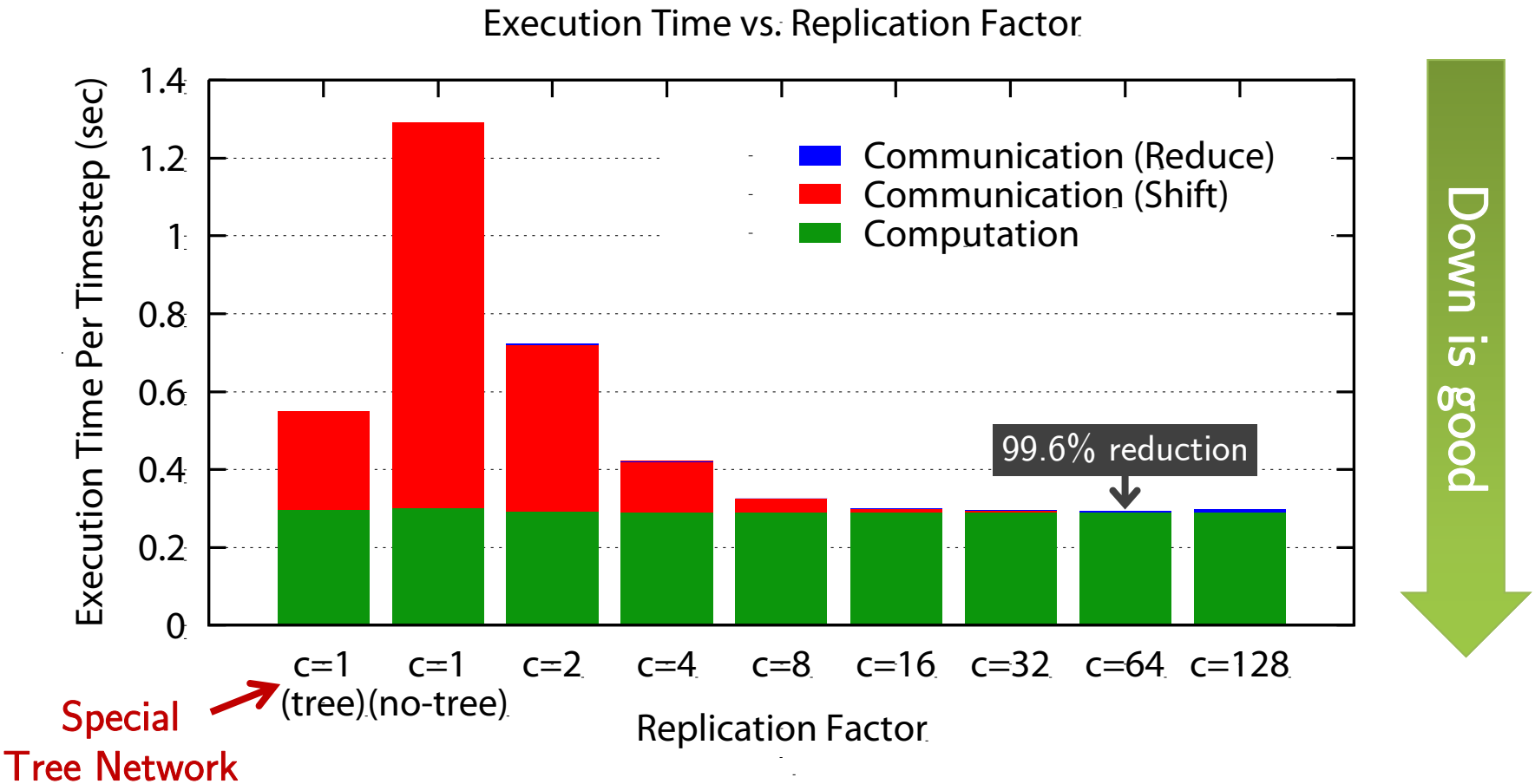
Often, $\sqrt{p}$ isn't the best c

- Intrepid; $\mathrm{n}=262\mathrm{K}$ particles, $\mathrm{p}=32\mathrm{K}$ cores

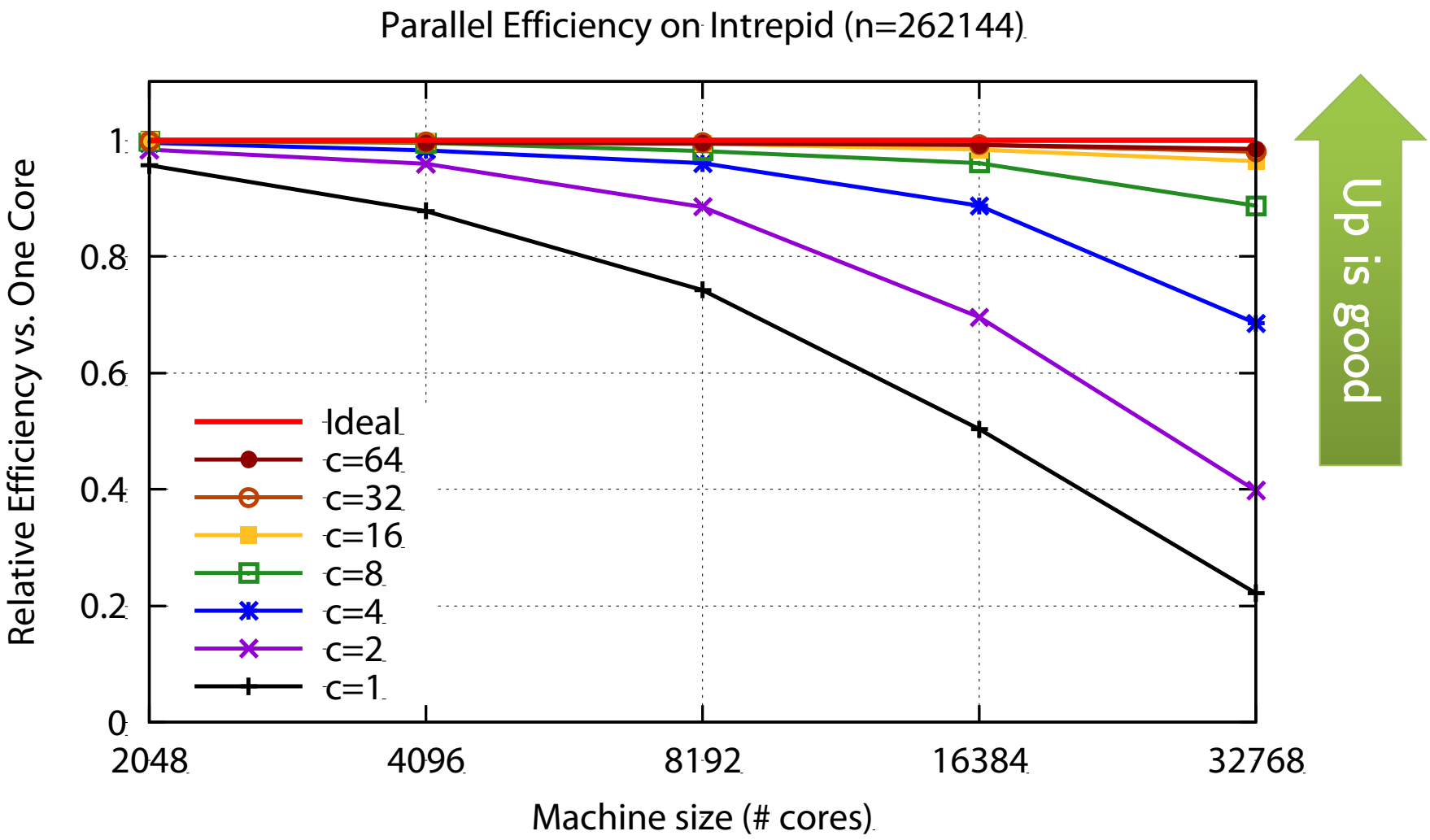


Often, $\sqrt{p}$ isn't the best c

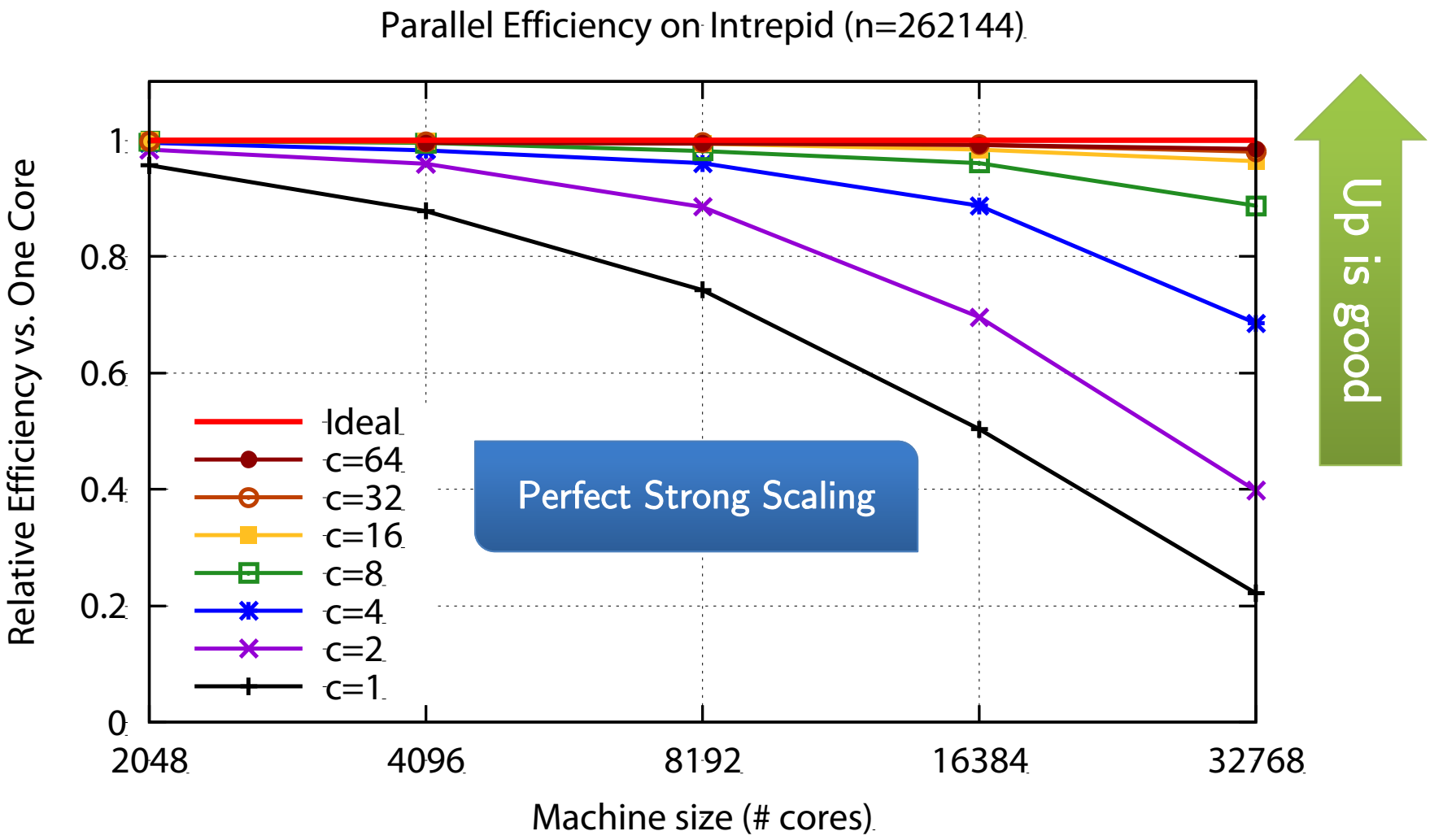
- Intrepid; $n=262K$ particles, $p=32K$ cores



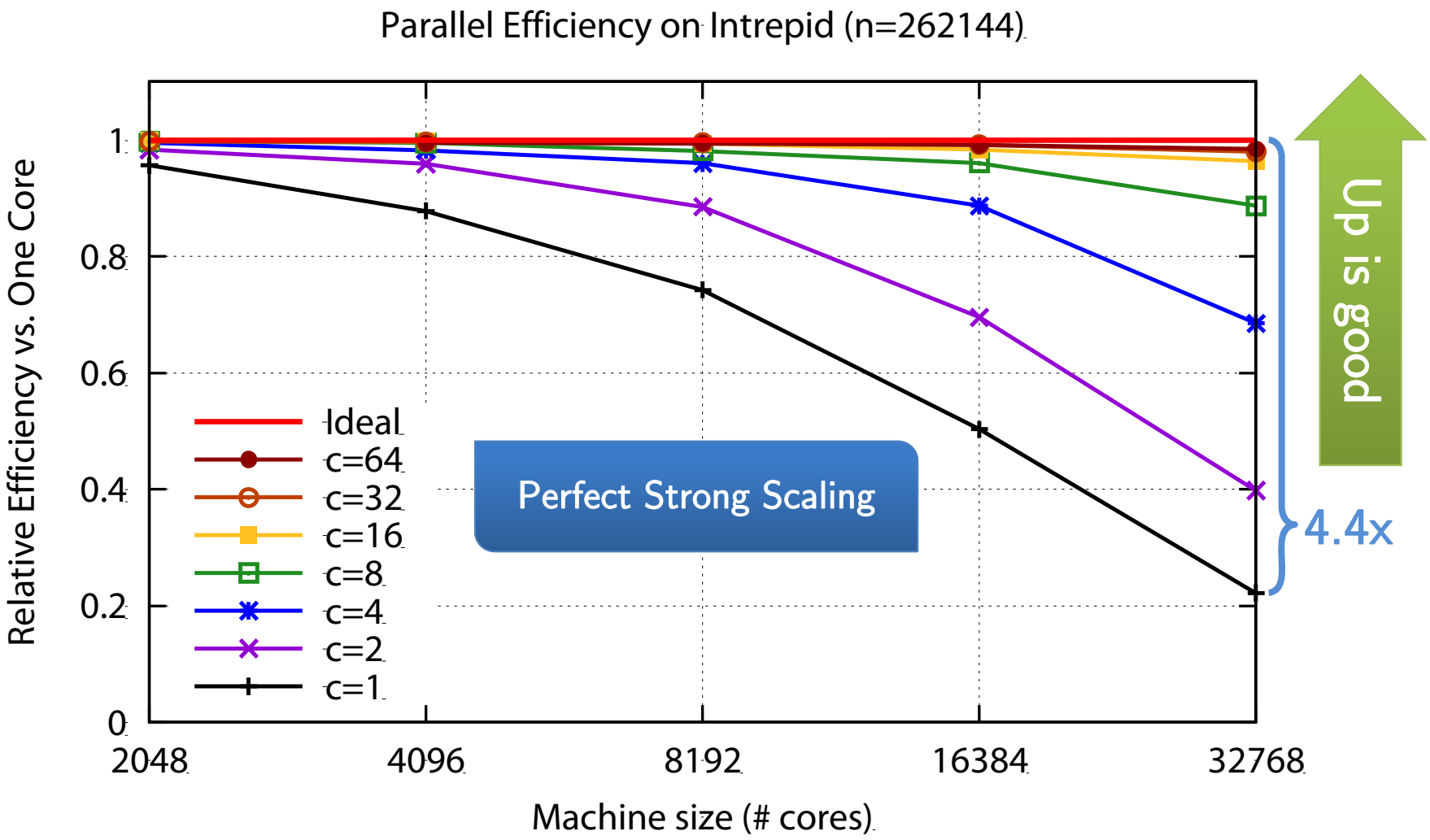
Near-Perfect Scalability



Near-Perfect Scalability



Near-Perfect Scalability

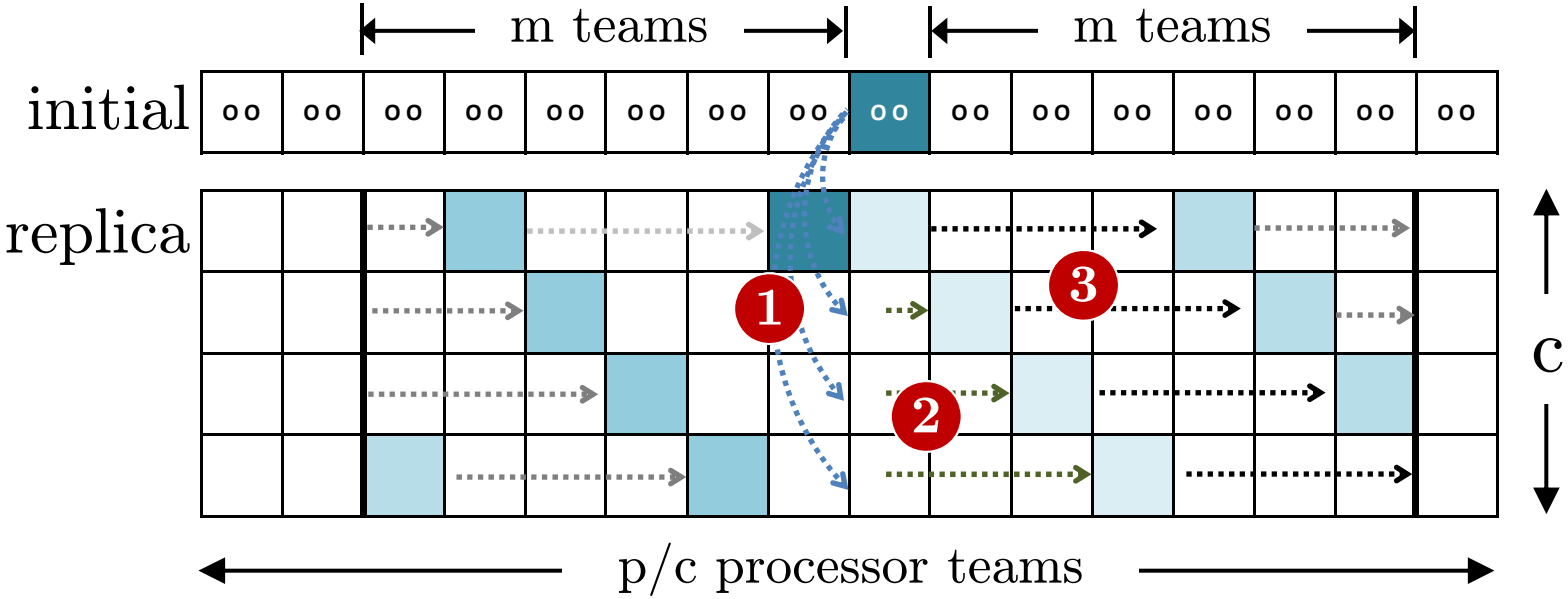


Extension for Finite Cutoff Distance

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 paper

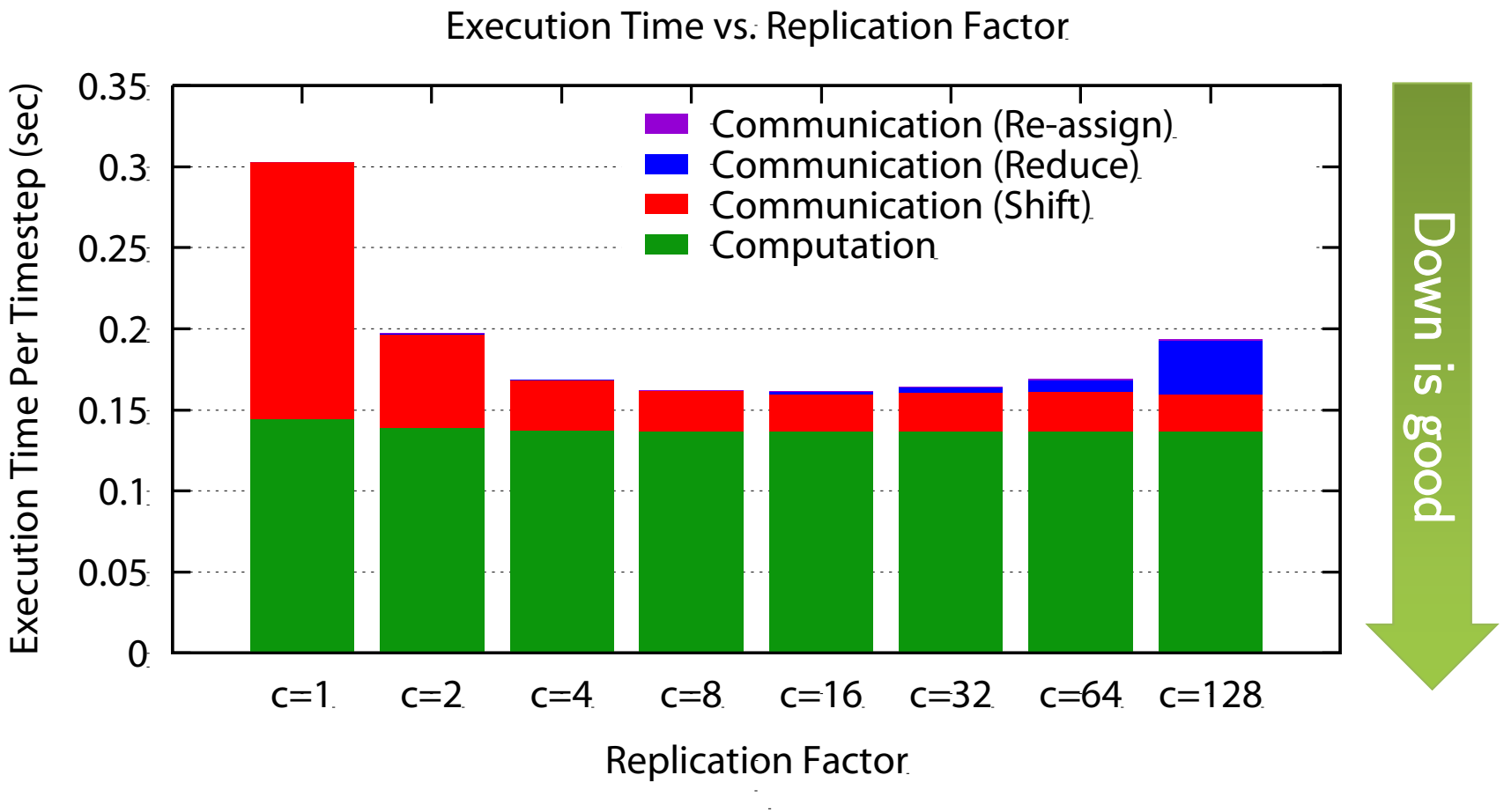
1D-Space Cutoff Algorithm



- Shift-modulo (wrap-around)

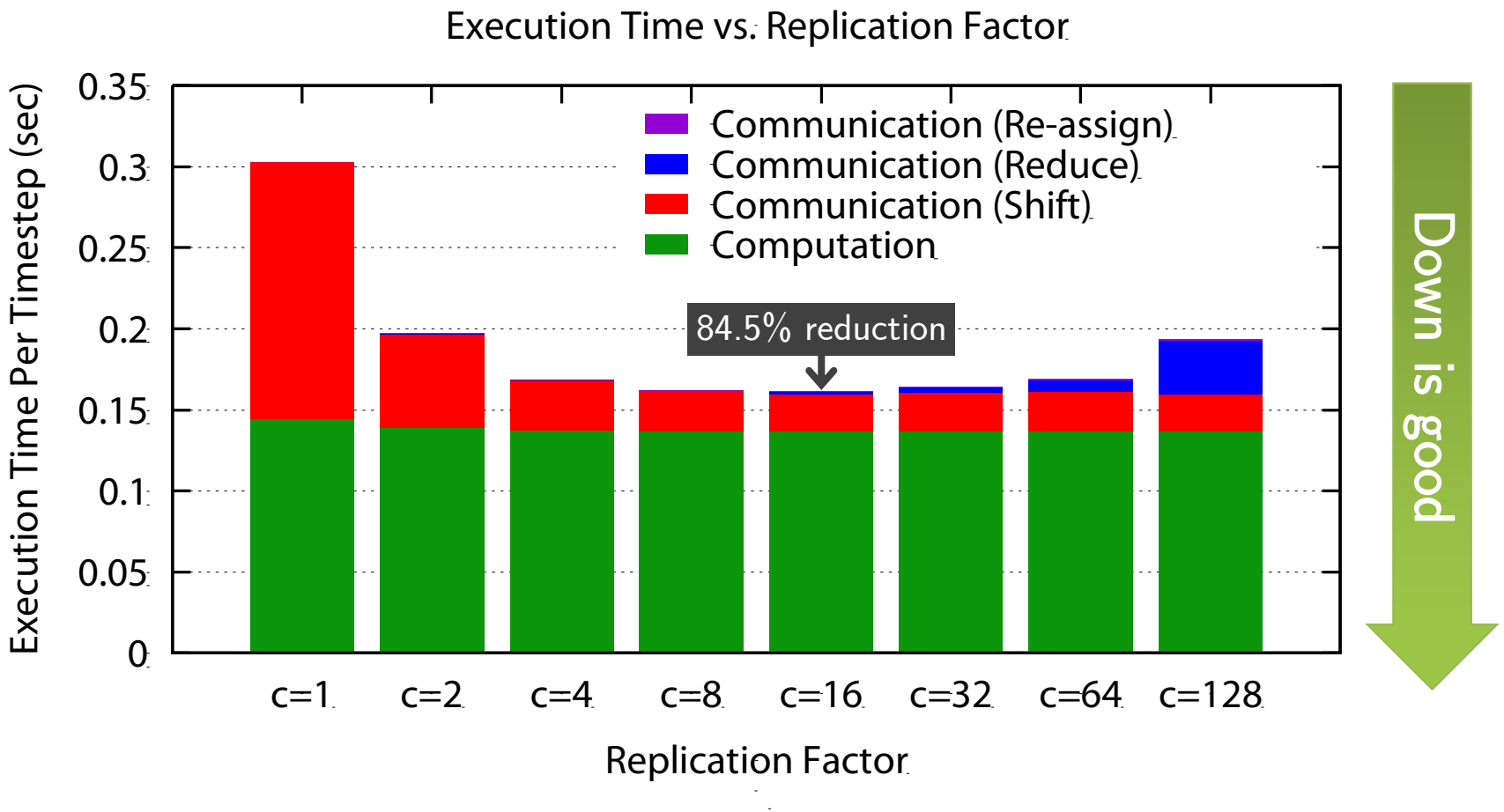
CA1D: Time Breakdown

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

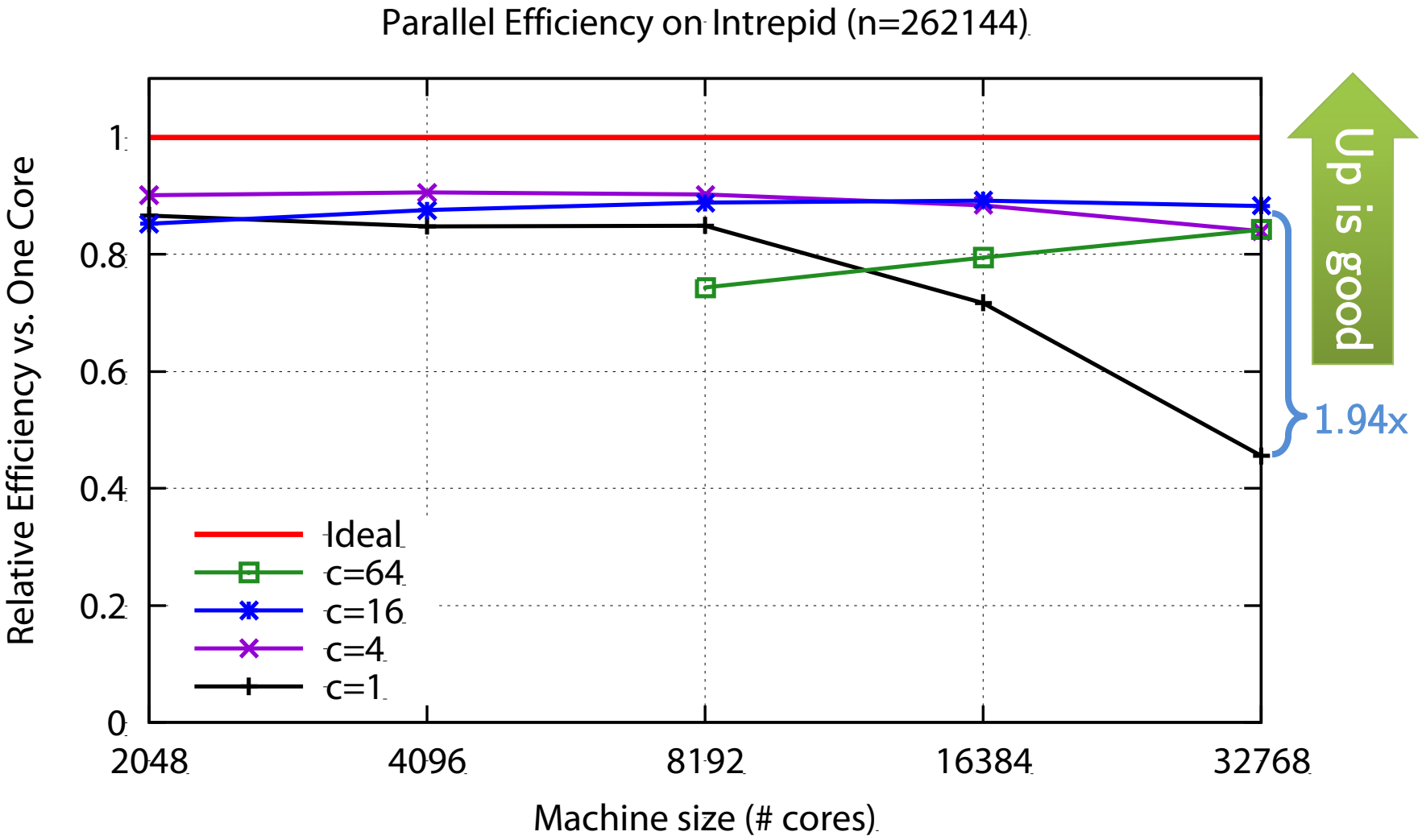


CA1D: Time Breakdown

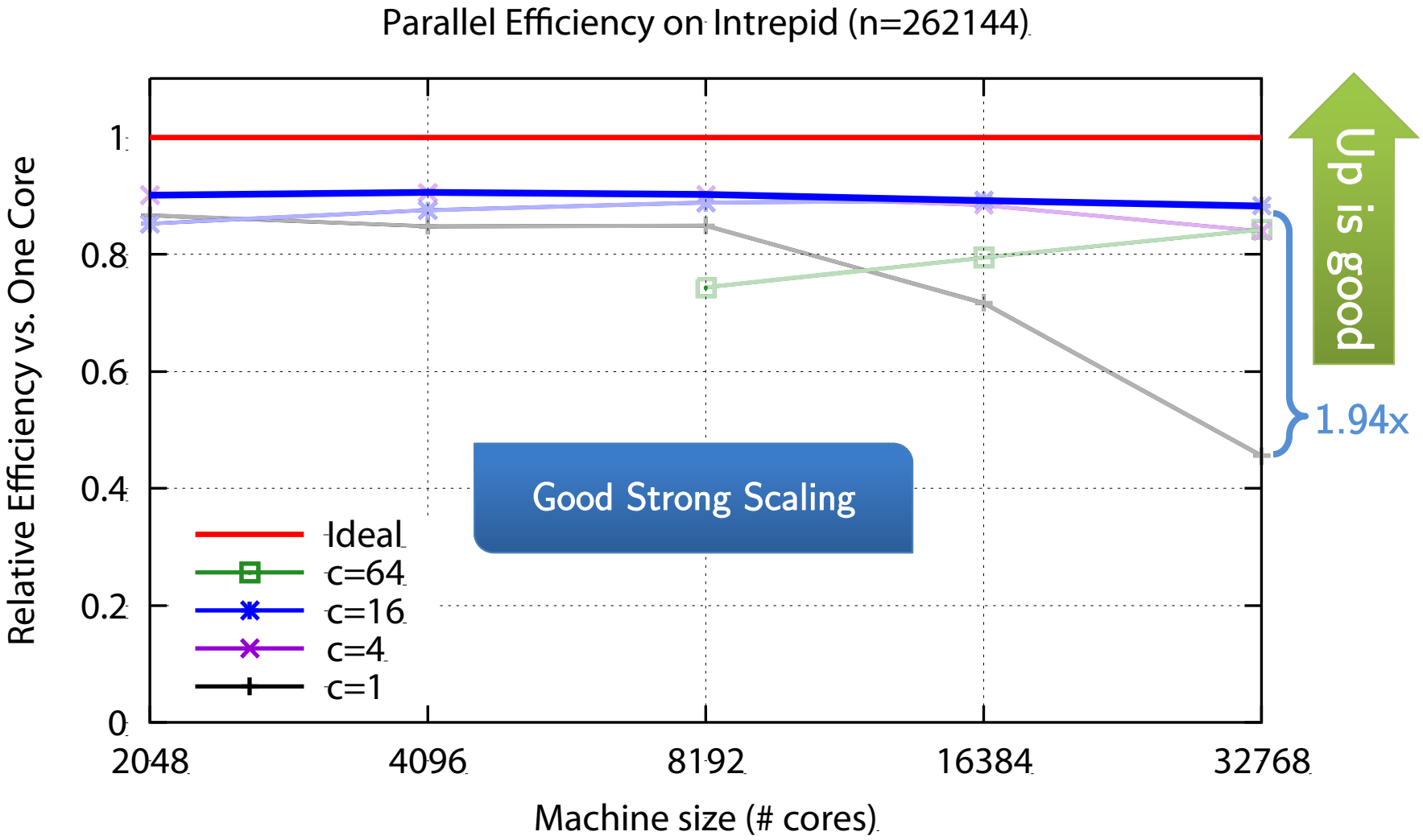
- Intrepid; $n=262K$ particles, $p=32K$ cores



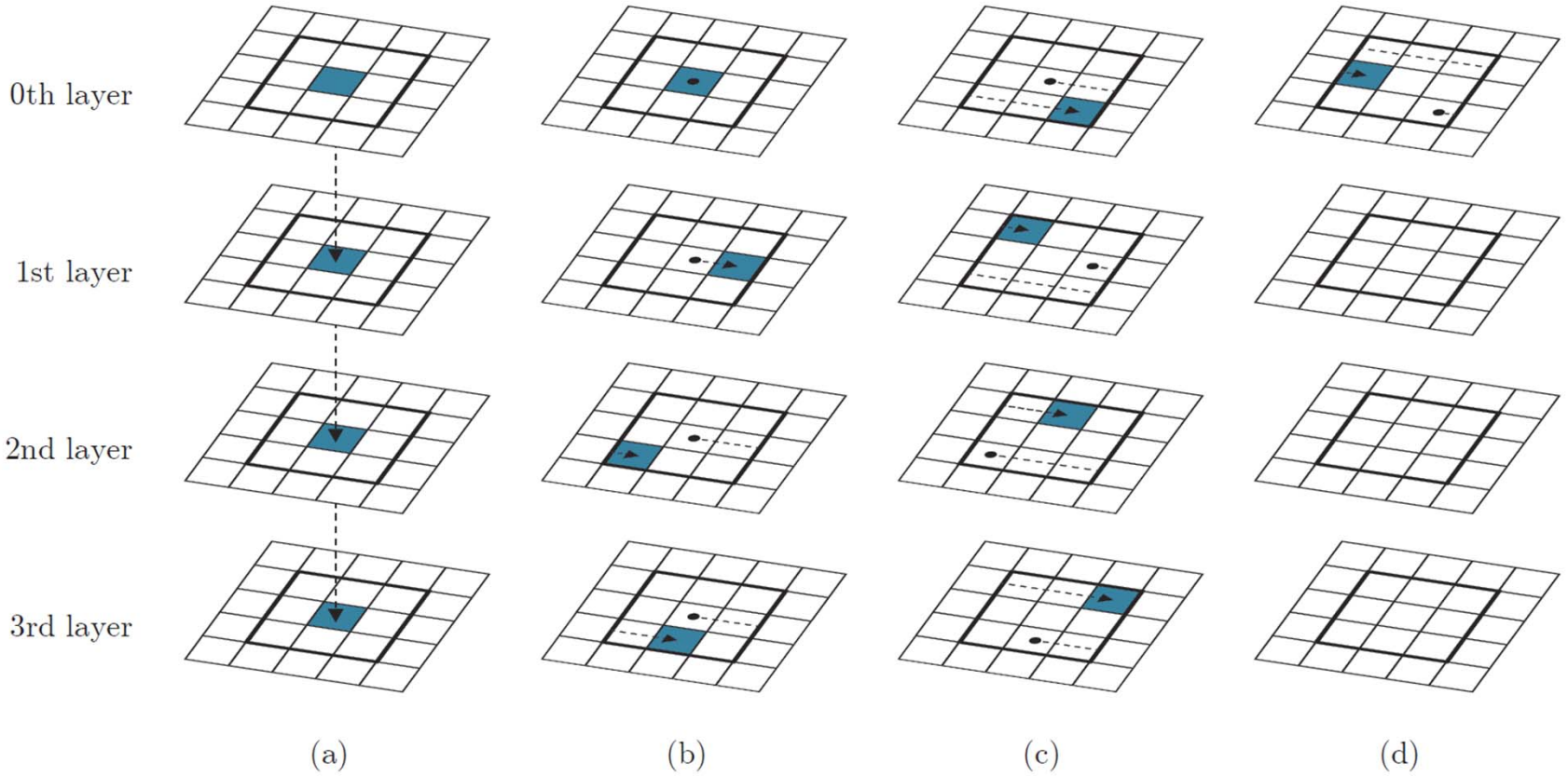
CA1D: Better Scalability



CA1D: Better Scalability

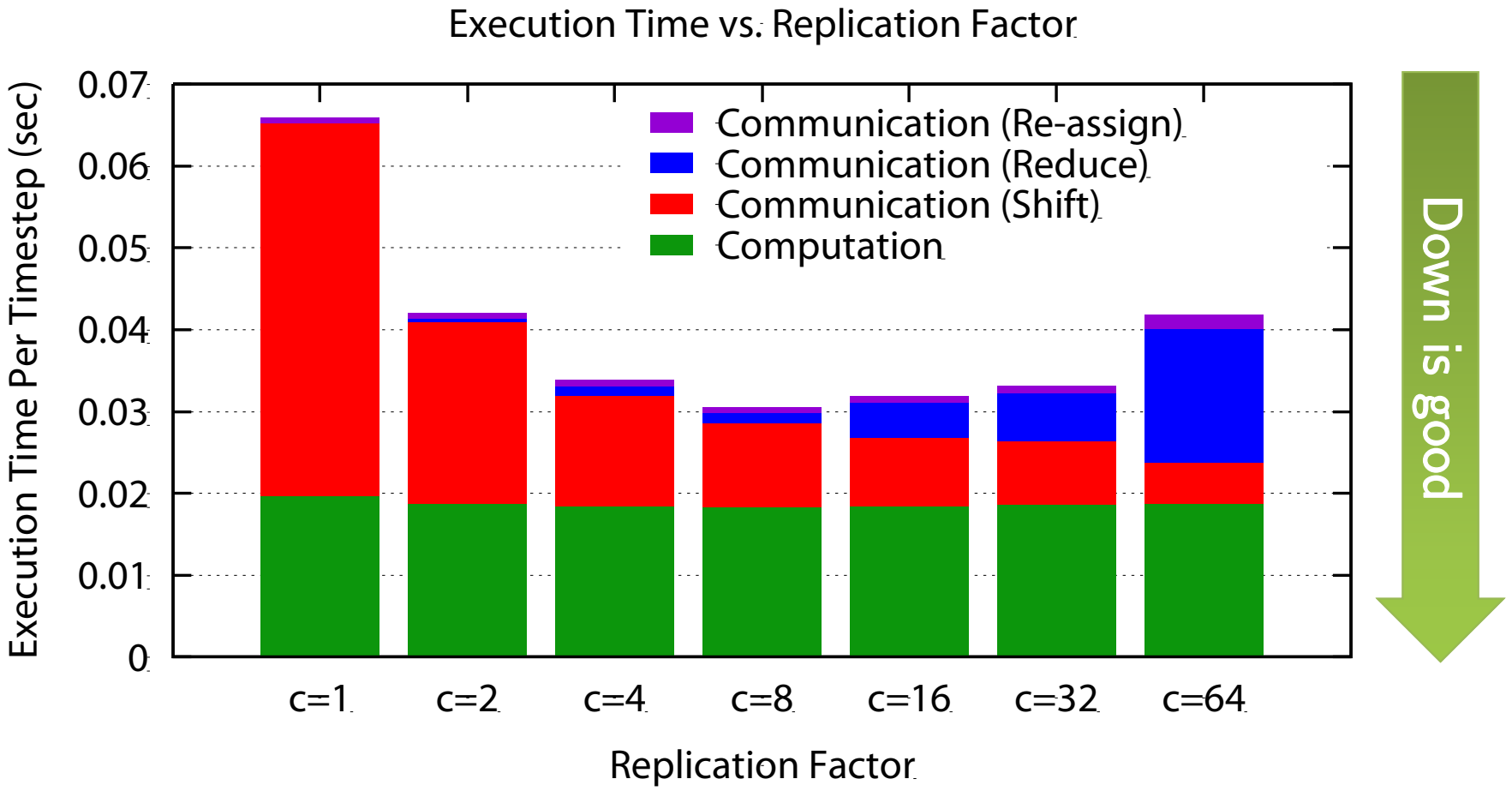


2D-space Cutoff Algorithm



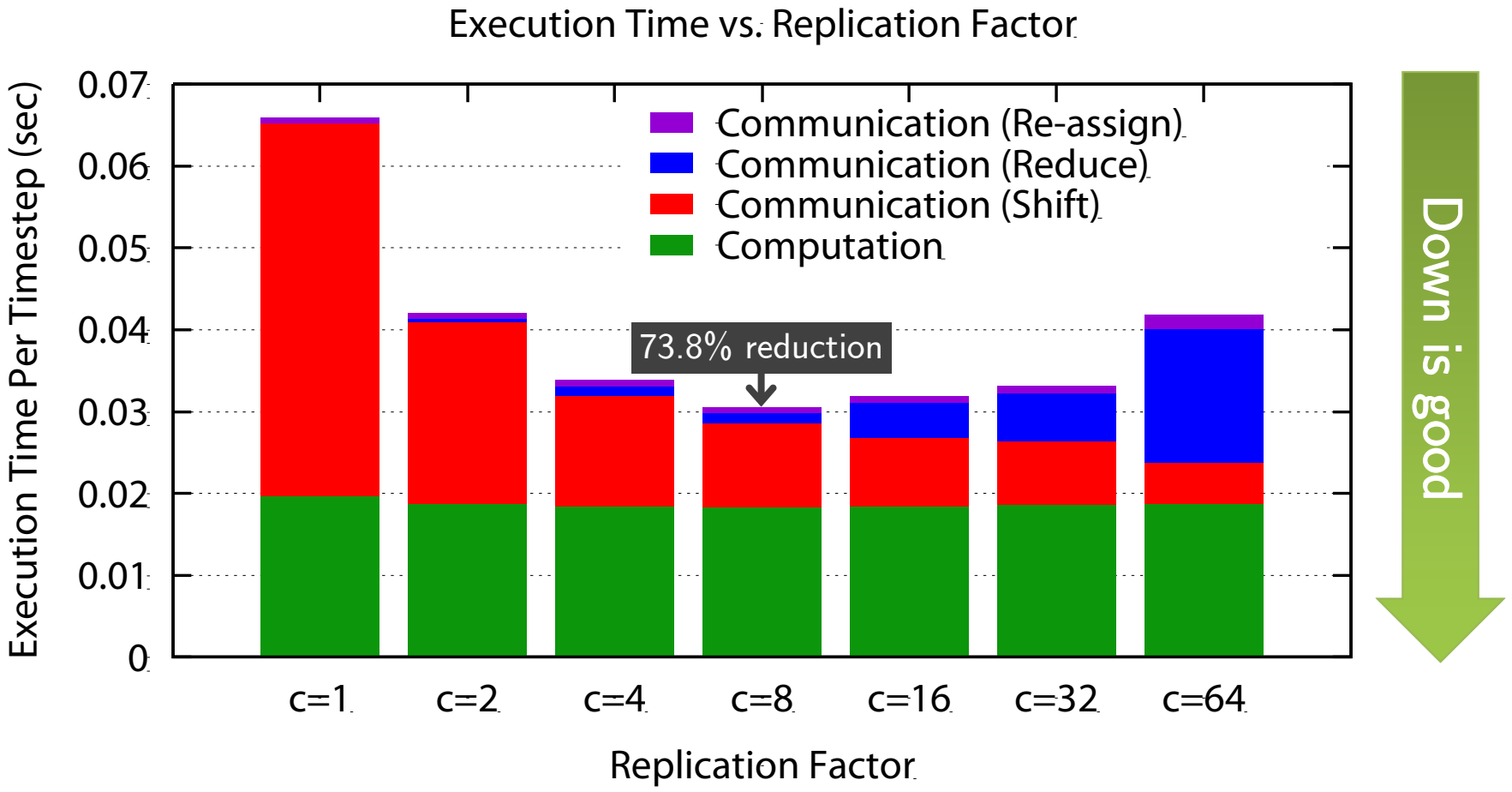
CA2D: Time Breakdown

- Hopper; $\mathrm{n}{=}196\mathrm{K}$ particles, $\mathrm{p}{=}24\mathrm{K}$ cores

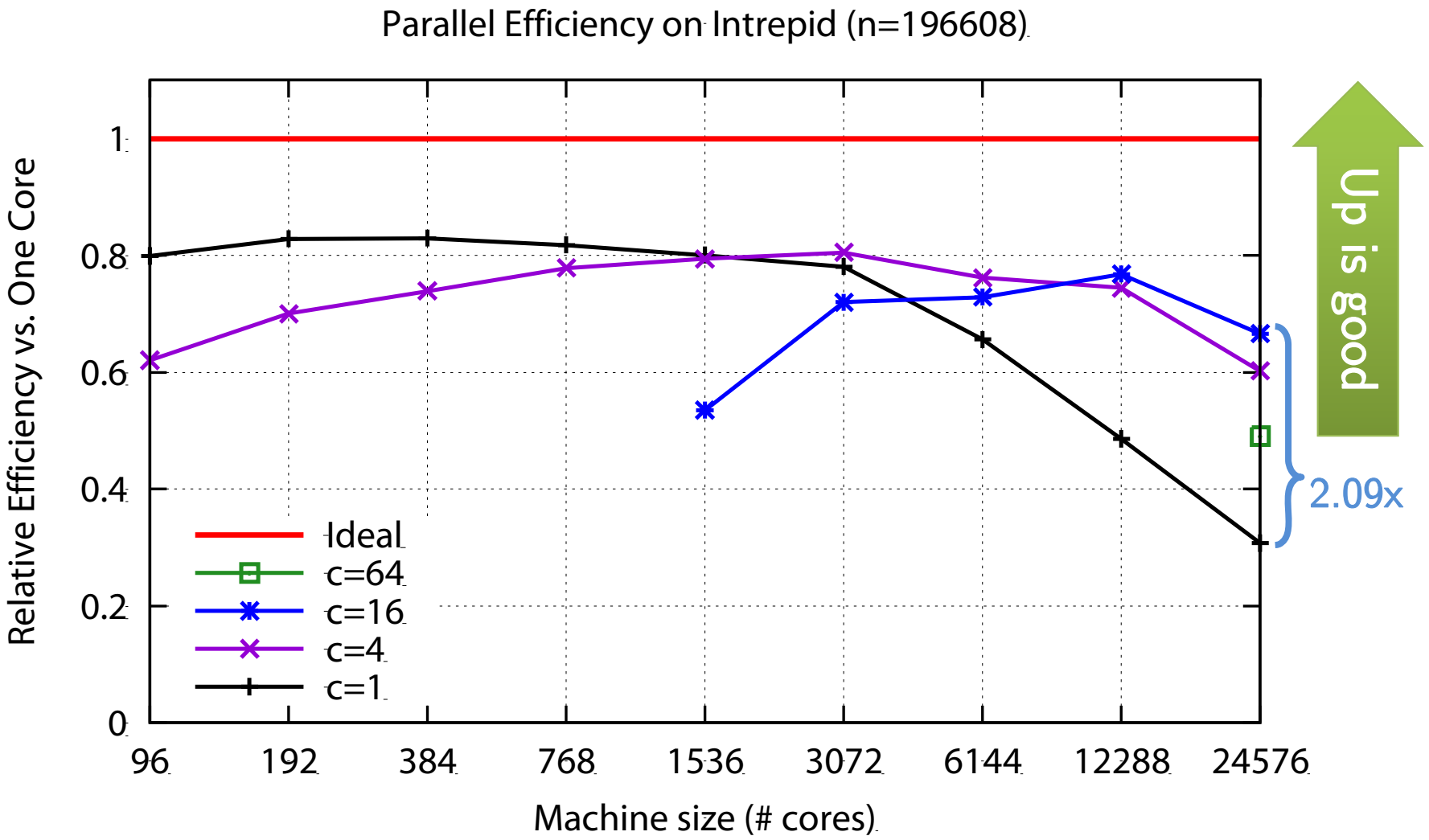


CA2D: Time Breakdown

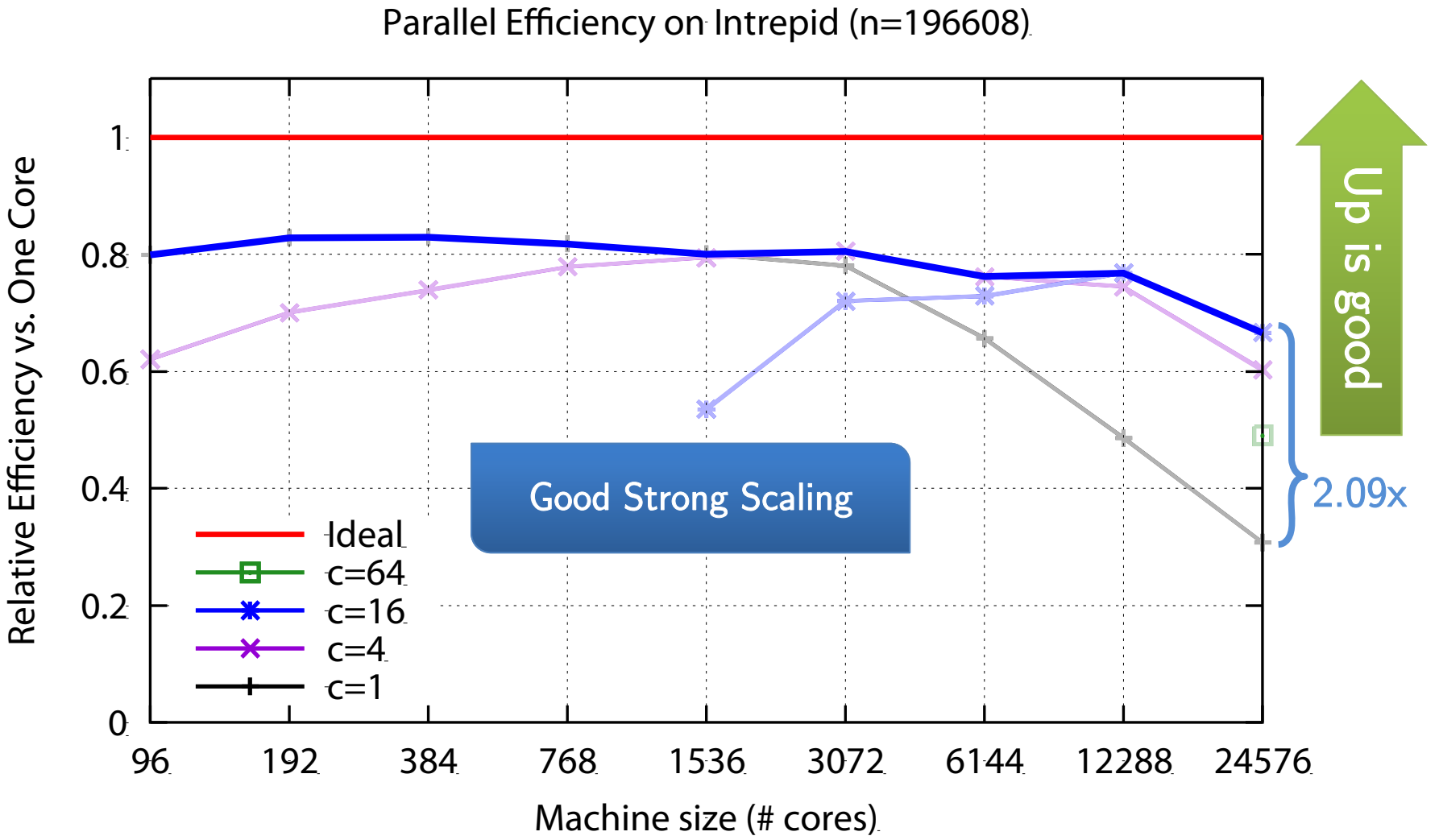
- Hopper; $\mathrm{n}=196\mathrm{K}$ particles, $\mathrm{p}=24\mathrm{K}$ cores



CA2D: Better Scalability



CA2D: Better Scalability



Conclusions

- Using c replications reduces
  - #**messages** sent by a factor of c^2
  - #**words** sent by a factor of c

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; $1 \leq c \leq \sqrt{p}$

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.

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
  - Database joins
  - Collision detection algorithms

References

- M. Driscoll, E. Georganas, P. Koanantakool, E. Solomonik, K. Yelick, **A Communication-Optimal N-Body Algorithm for Direct Interactions.** In proceedings of the IPDPS 2013
- 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.