A Communication-Optimal N-Body Algorithm for Direct Interactions

Michael Driscoll, Evangelos Georganas, Penporn Koanantakool, Edgar Solomonik, and Katherine Yelick

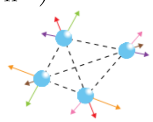
Motivation

- N-Body requires n^2 interactions = lots of communication
- Lower bounds from [Ballard, Demmel, Holtz, S. 2011a]

$$S = \Omega\left(\frac{F}{H}\right), \qquad W = \Omega(S \cdot M) = \Omega\left(\frac{M \cdot F}{H}\right)$$

- N-Body: at most M^2 force evaluations

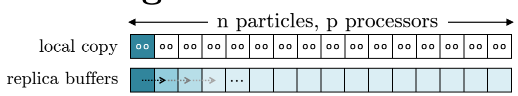
$$S_{\rm NB} = \Omega\left(\frac{F}{M^2}\right), \qquad W_{\rm NB} = \Omega\left(\frac{F}{M}\right)$$



- M in the denominator = using extra memory can decrease lower bound.

No cutoff (all-pairs)

Naïve Algorithm

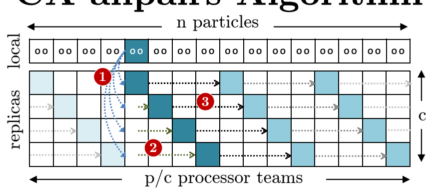


$$S_{\rm allpairs} = \Omega\left(\frac{n^2/p}{M^2}\right) = \Omega\left(\frac{n^2/p}{(n/p)^2}\right) = \Omega(p)$$

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

- Each processor sends p messages of size n/p
- Communication-optimal.

CA-allpairs Algorithm



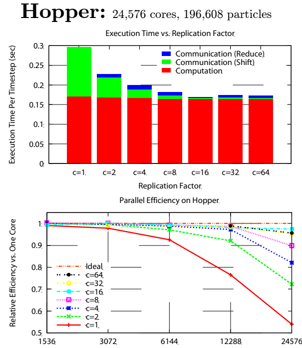
- Replicate over c layers
- Shift team particles along diagonal
- For p/c^2 steps:
  - Shift replicas by c teams
  - Compute interactions
- Reduce across column

$$S_{\rm CA} = \Omega\left(\frac{\frac{n^2}{p}}{\left(\frac{n}{p/c}\right)^2}\right) = \Omega\left(\frac{p}{c^2}\right)$$

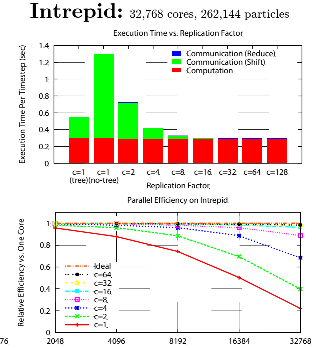
$$W_{\rm CA} = \Omega\left(\frac{\frac{n^2}{p}}{\frac{n}{p/c}}\right) = \Omega\left(\frac{n}{c}\right)$$

	Message Size	Latency ($S_{\rm CA}$)	Bandwidth ($W_{\rm CA}$)
1. Broadcast	$O\left(\frac{n}{p/c}\right)$	$O(\log c)$	$O\left(\frac{n}{p/c}\cdot \log c\right) = O\left(\frac{nc \log c}{p}\right)$
2. Skew	$O\left(\frac{n}{p/c}\right)$	$O(1)$	$O\left(\frac{n}{p/c}\cdot 1\right) = O\left(\frac{nc}{p}\right)$
3. Shift	$O\left(\frac{n}{p/c}\right)$	$O\left(\frac{p/c}{c}\right) = \boldsymbol{O\left(\frac{p}{c^2}\right)}$	$O\left(\frac{n}{p/c}\cdot \frac{p}{c^2}\right) = \boldsymbol{O\left(\frac{n}{c}\right)}$

Hopper: 24,576 cores, 196,608 particles



Intrepid: 32,768 cores, 262,144 particles

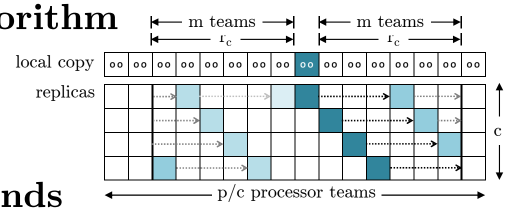


Cutoff (1D)

Assumptions

- Particles are uniformly distributed.
- Cutoff distance (r_c) spans multiple processor areas.
- Particles have 1D coordinates

Algorithm



Bounds

- Let k be #interactions per particle. $k = \frac{m}{p/c} \cdot n = \frac{mnc}{p}$

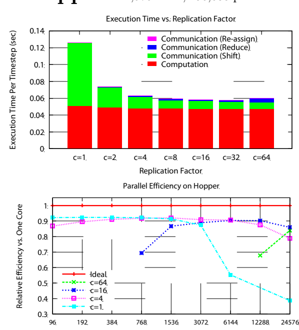
$$S_{\rm cutoff} = \Omega\left(\frac{nk/p}{M^2}\right) = \Omega\left(\frac{\frac{n}{p}\cdot\left(\frac{mnc}{p}\right)}{(nc/p)^2}\right) = \Omega\left(\frac{m}{c}\right)$$

$$W_{\rm cutoff} = \Omega\left(\frac{nk/p}{M}\right) = \Omega\left(\frac{\frac{n}{p}\cdot\left(\frac{mnc}{p}\right)}{nc/p}\right) = \Omega\left(\frac{mn}{p}\right)$$

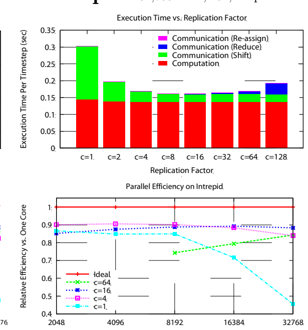
	Message Size	Latency ($S_{\rm 1D}$)	Bandwidth ($W_{\rm 1D}$)
1. Broadcast	$O\left(\frac{n}{p/c}\right)$	$O(\log c)$	$O\left(\frac{n}{p/c}\cdot \log c\right) = O\left(\frac{nc \log c}{p}\right)$
2. Skew	$O\left(\frac{n}{p/c}\right)$	$O(1)$	$O\left(\frac{n}{p/c}\cdot 1\right) = O\left(\frac{nc}{p}\right)$
3. Shift	$O\left(\frac{n}{p/c}\right)$	$\boldsymbol{O\left(\frac{m}{c}\right)}$	$O\left(\frac{n}{p/c}\cdot \frac{m}{c}\right) = \boldsymbol{O\left(\frac{mn}{p}\right)}$

Performance

Hopper: 24,576 cores, 196,608 particles



Intrepid: 32,768 cores, 262,144 particles

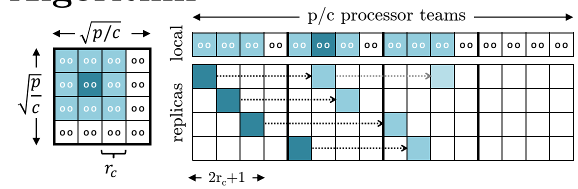


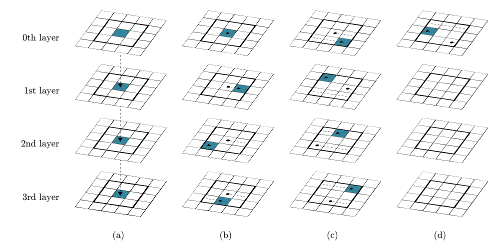
Cutoff (2D)

Assumptions

- Particles are uniformly distributed.
- Cutoff distance (r_c) spans multiple processor areas.
- Particles have 2D coordinates

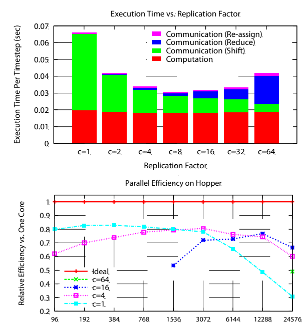
Algorithm



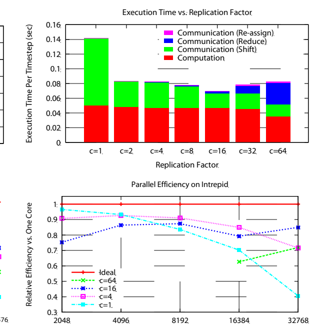


Performance

Hopper: 24,576 cores, 196,608 particles



Intrepid: 32,768 cores, 262,144 particles



Conclusions

- Using extra memory (**c** copies) reduces
  - Latency by a factor of $\mathbf{c^2}$
  - Bandwidth by a factor of **c**.
- Communication avoidance decreases overall execution time for **communication-bound** problems.
- Observed up to **11.8×** speedup over the non-CA version.
- Negligible benefits on compute-bound problems.
- **Tunable replication factor 'c'** in cases where cost of reduction comprises most of the communication cost.