Dynamic Load Balancing for Particle Methods

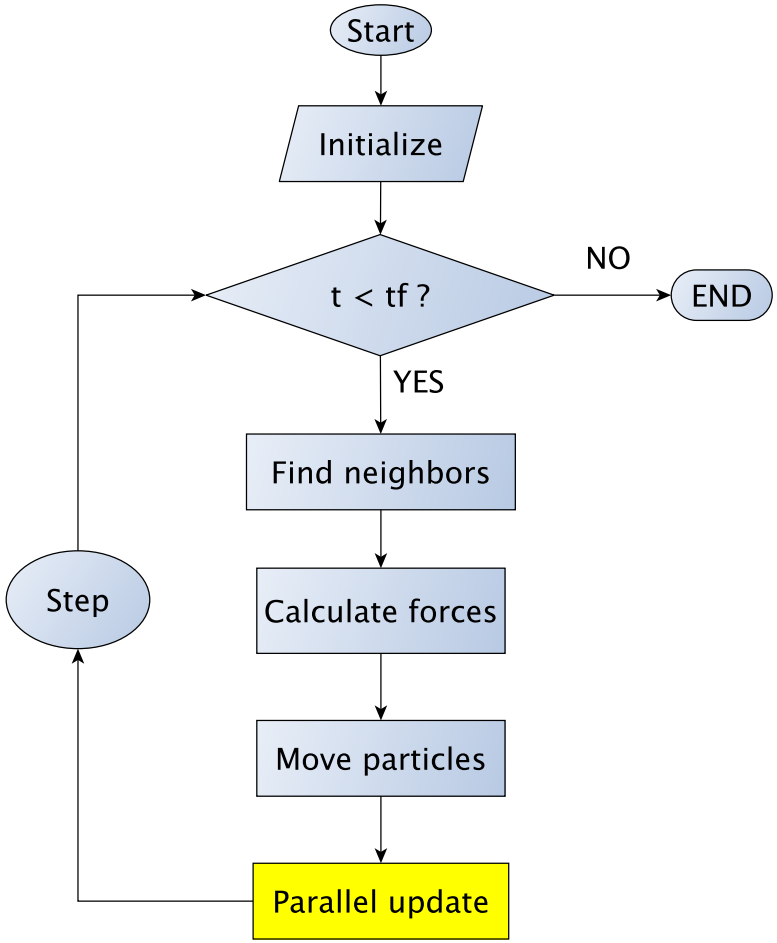
K. Puri, P. Ramachandran and P. Godbole

Department of Aerospace Engineering,
IIT Bombay

Particles 2013, Stuttgart, Germany

Outline

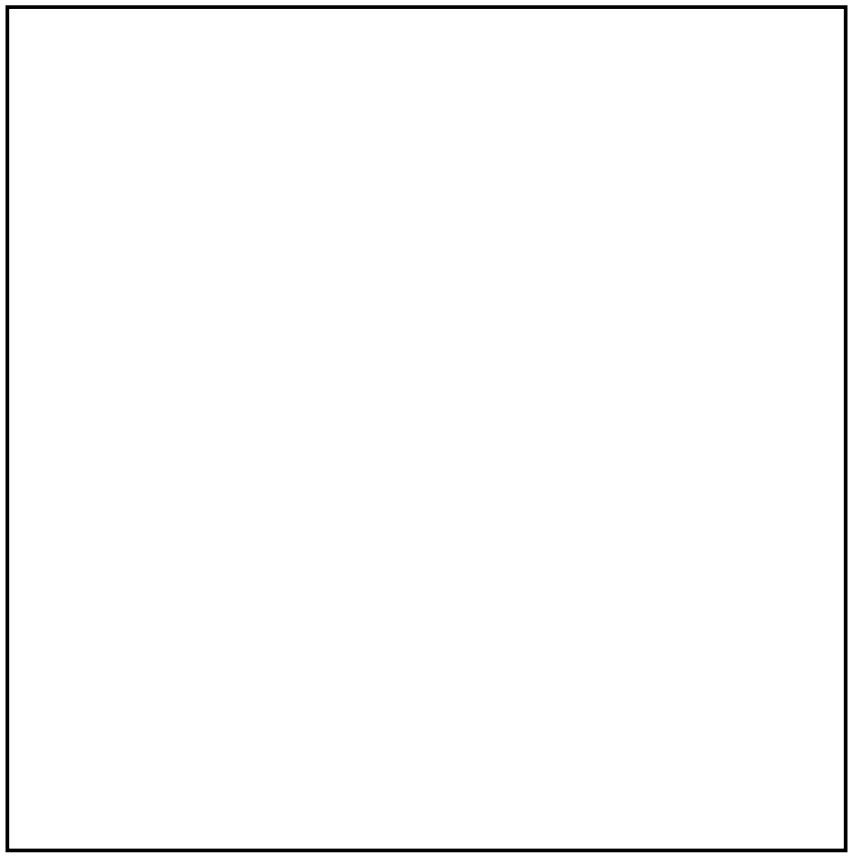
PySPH


Start
Initialize
t < tf ?
NO
END
YES
Find neighbors
Calculate forces
Move particles
Parallel update
Step

Serial algorithm in a nutshell

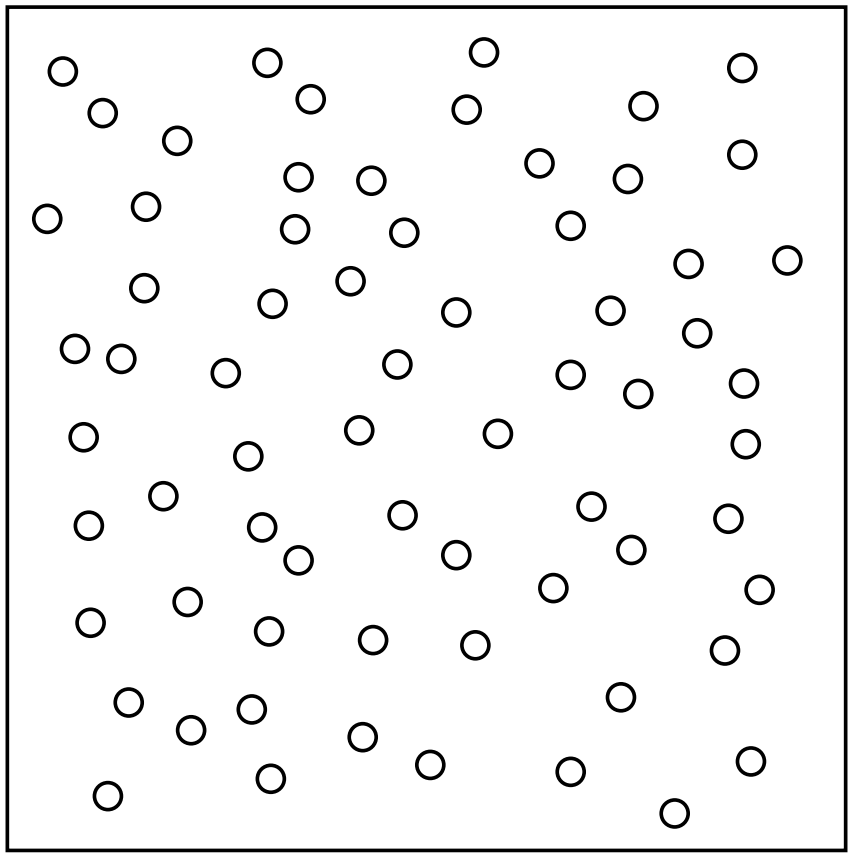
Serial algorithm in a nutshell

Given a domain...



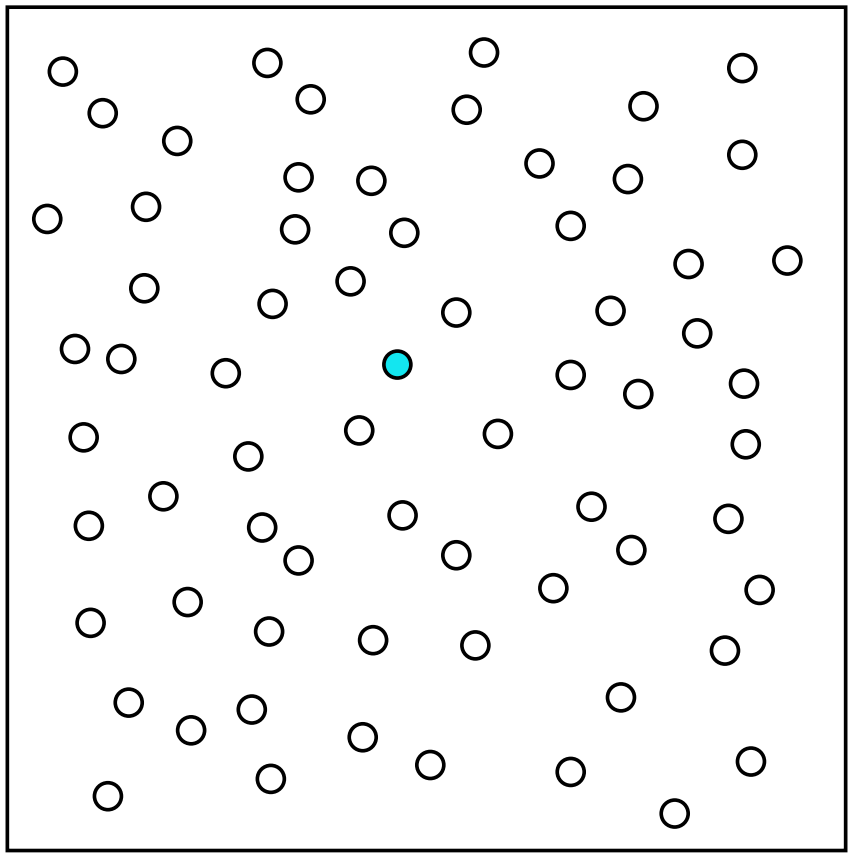
Serial algorithm

Discretized with *Particles*



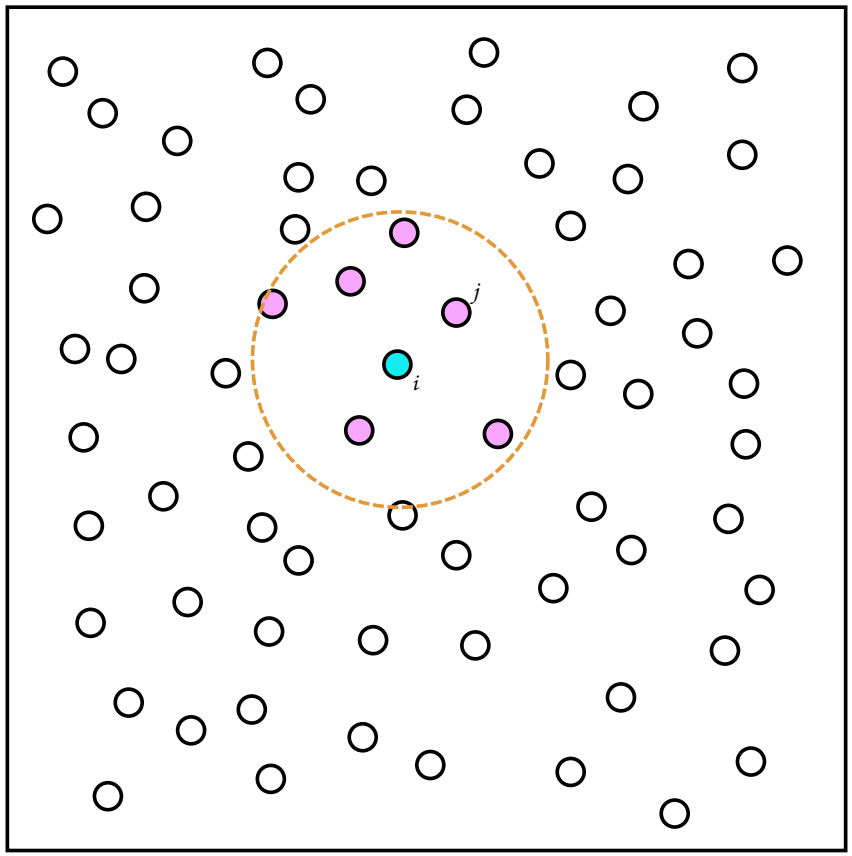
Serial algorithm

For every particle...



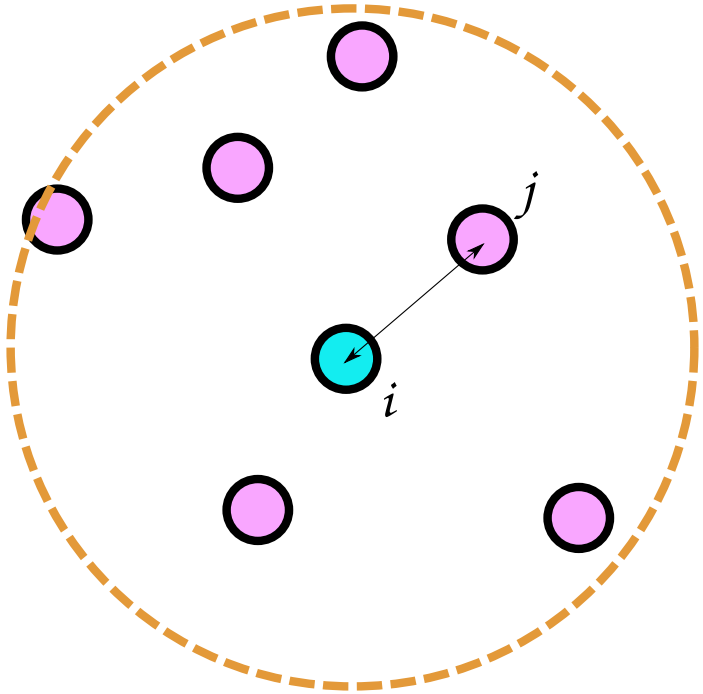
Serial algorithm

Find nearest neighbors...



Serial algorithm

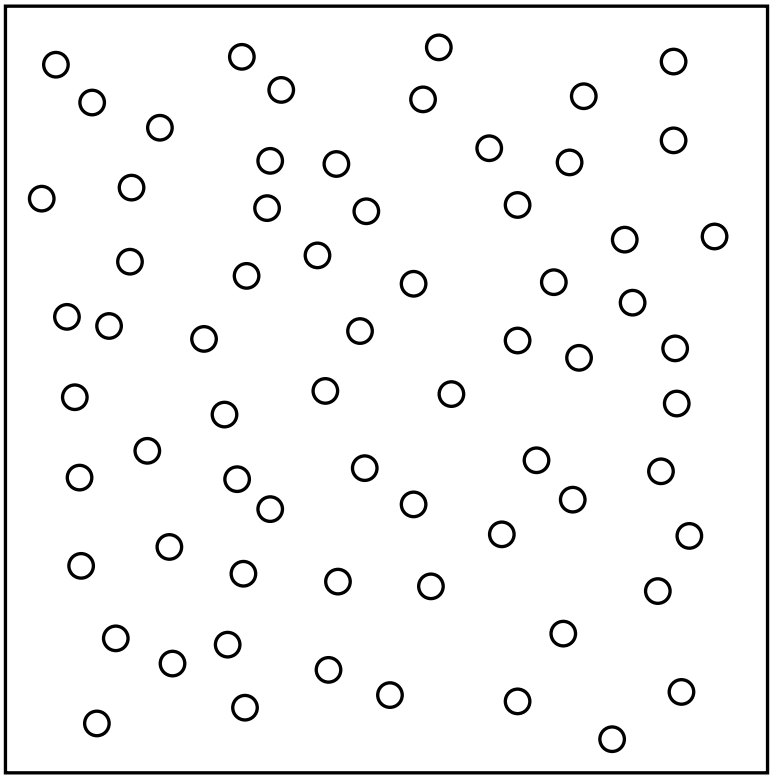
Compute interactions..



$$\frac{dU_i}{dt} = - \sum_{j \in \mathcal{N}(i)} m_j \mathcal{F}_{ij} \nabla_i W_{ij}$$

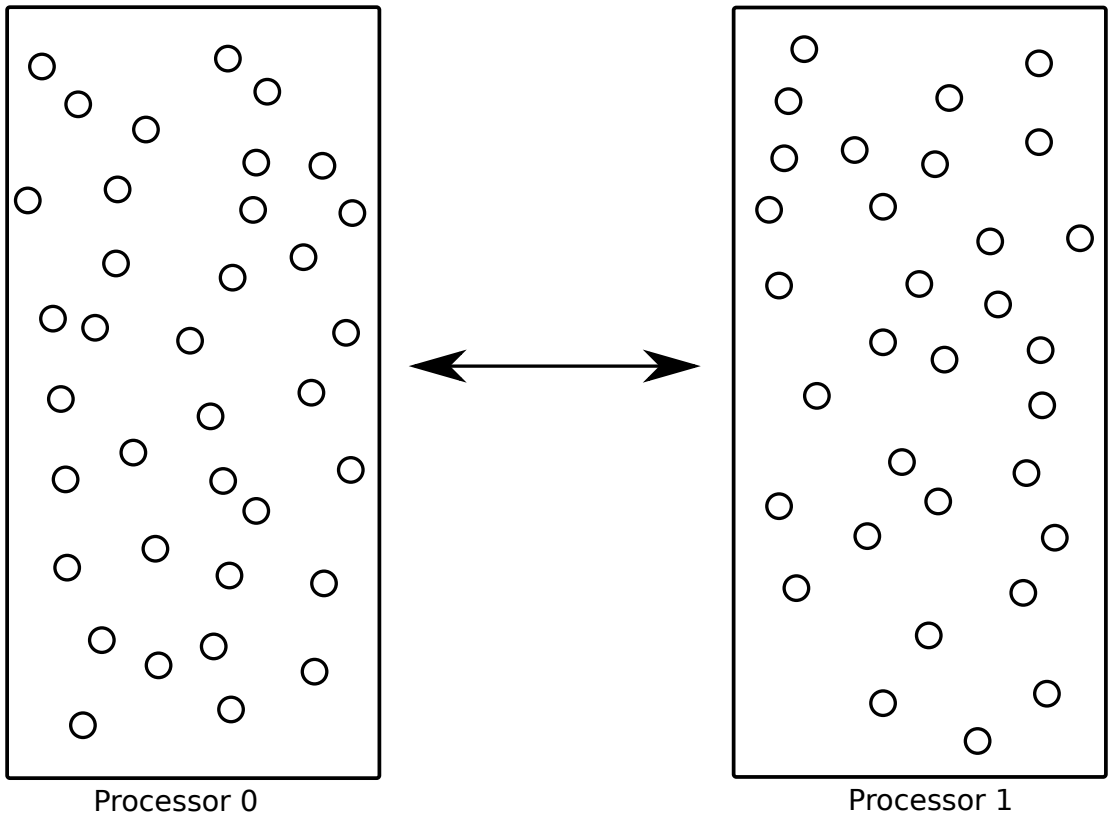
Doing it in Parallel

Given the domain discretized with *Particles*..



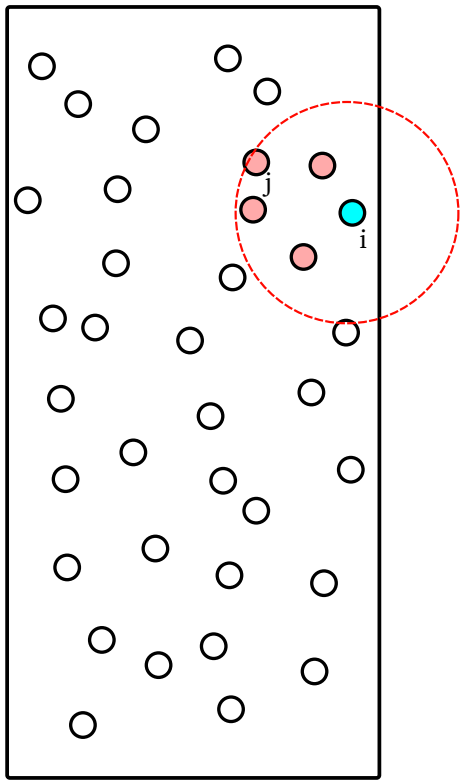
Doing it in Parallel

Partition it across processors : *Load-balancing*

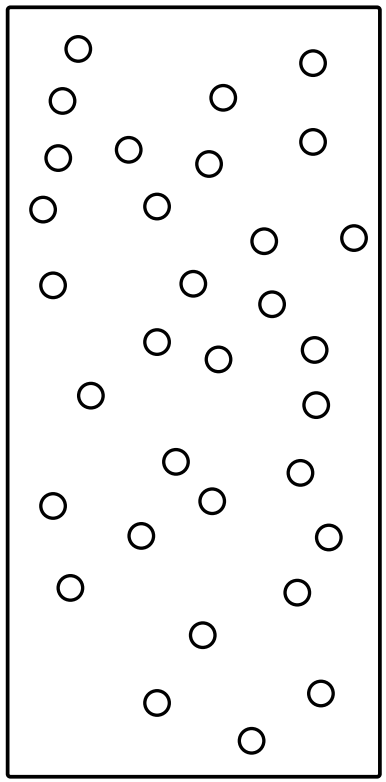


Doing it in Parallel

For every particle, find neighbors.



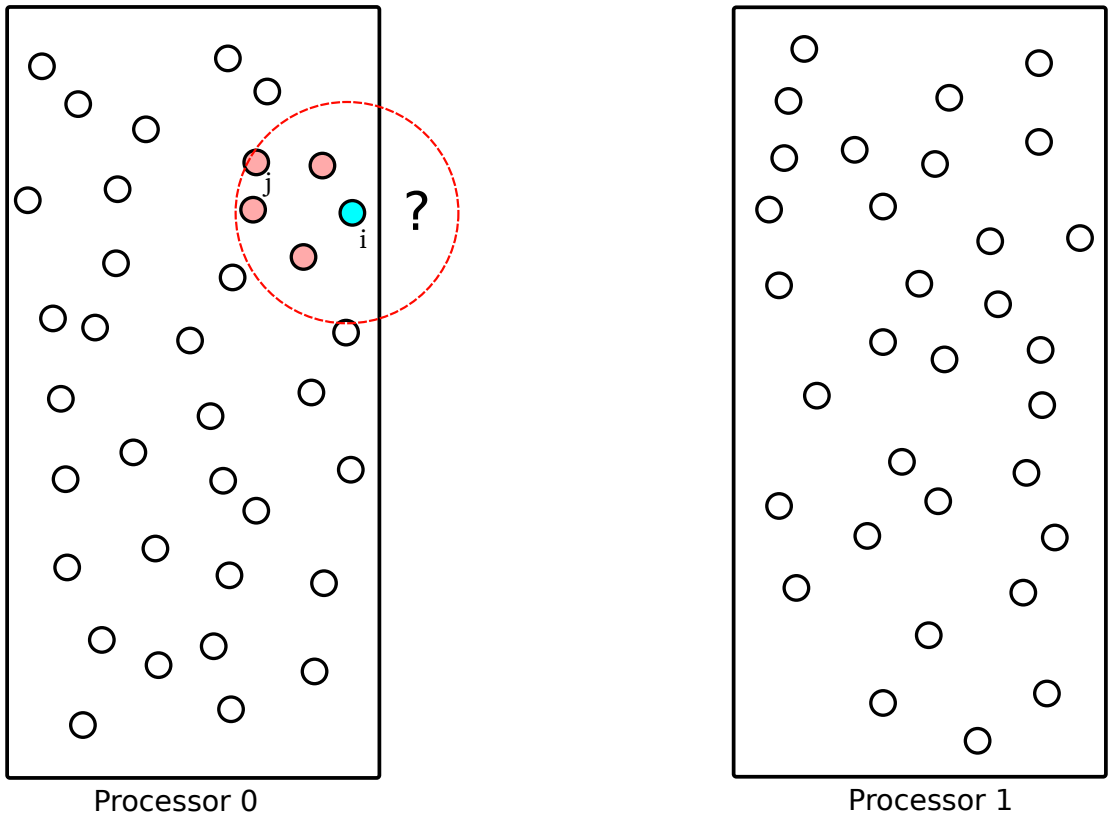
Processor 0



Processor 1

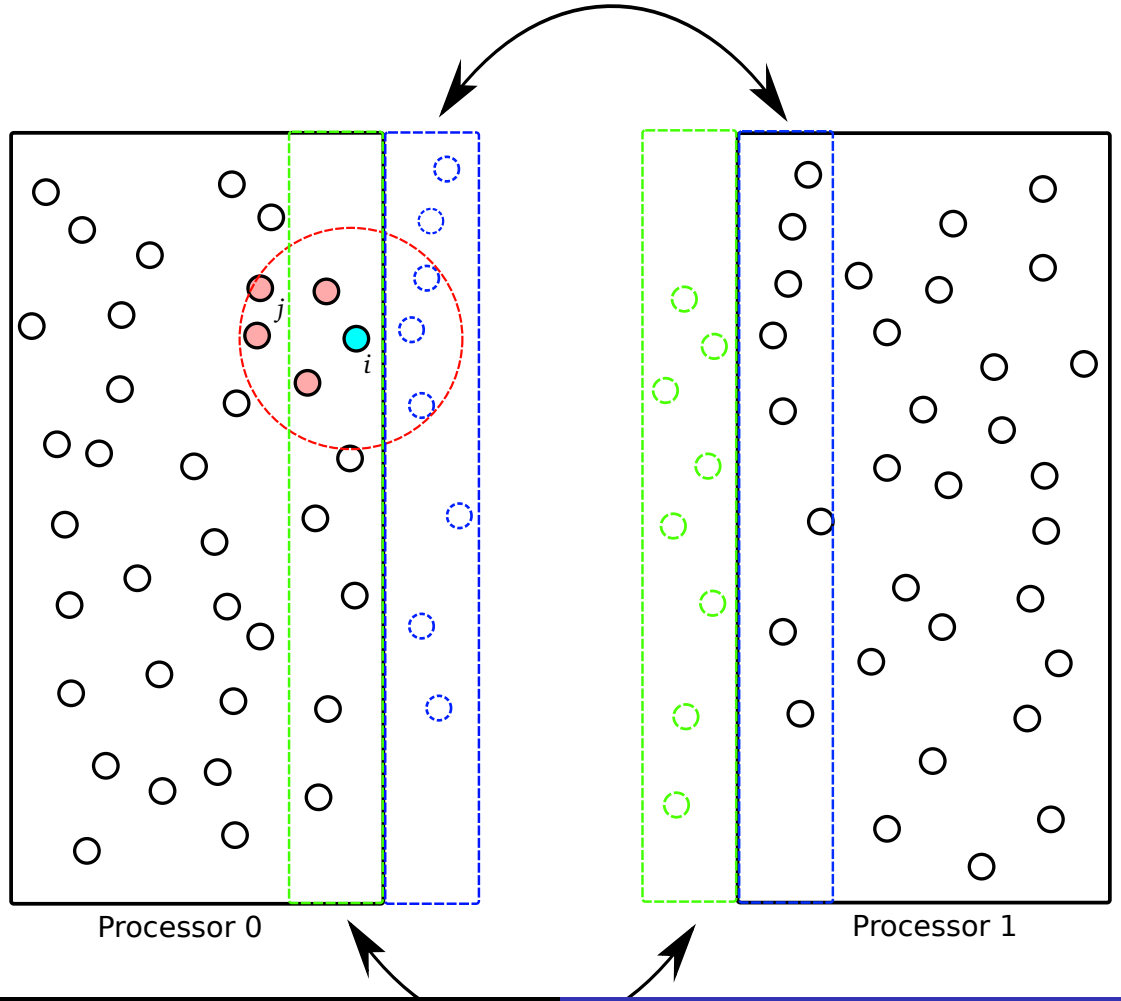
Doing it in Parallel

For every particle, find neighbors. *Oops!*



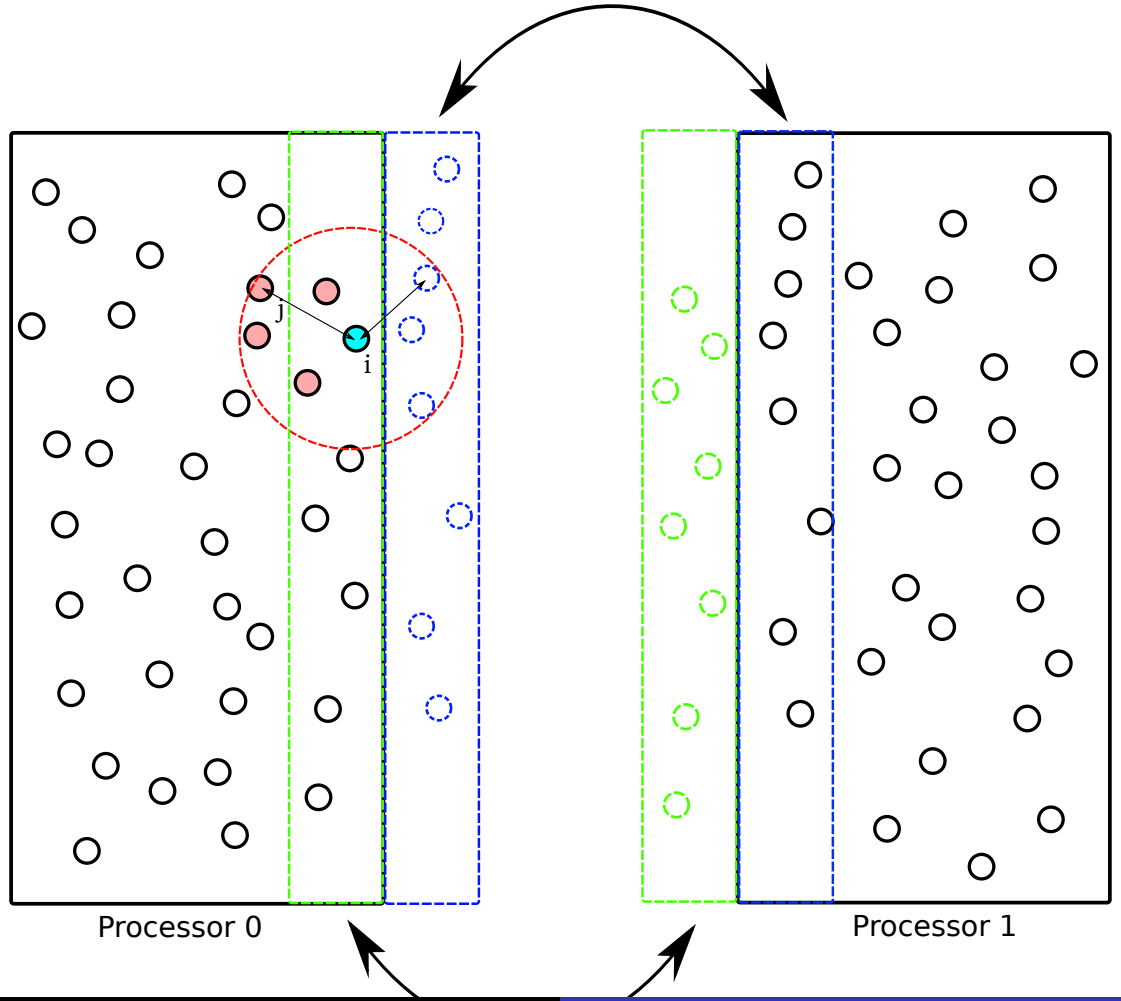
Doing it in Parallel

Exchange ghost data.



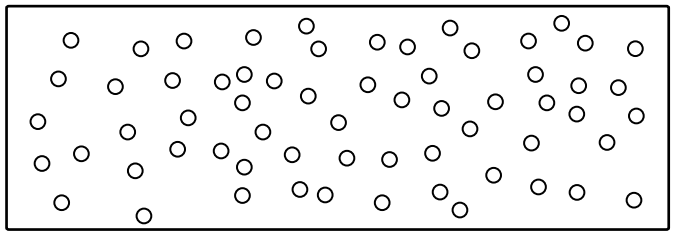
Doing it in Parallel

Exchange ghost data. And compute interactions..

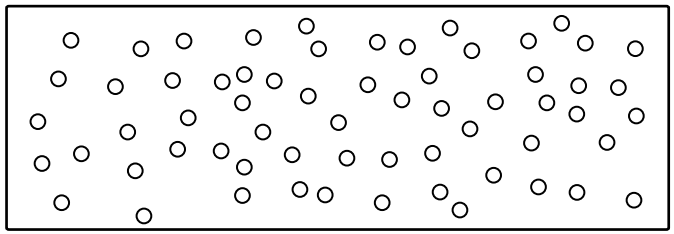


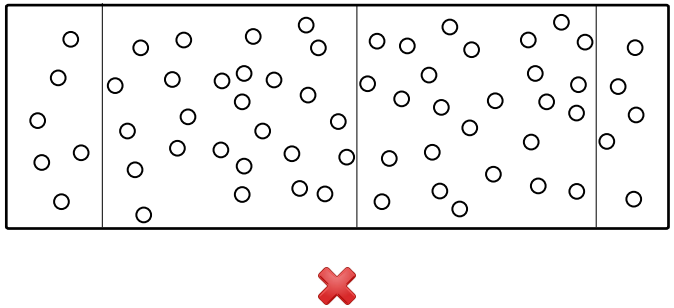
Requirements of a good partition

Requirements of a good partition

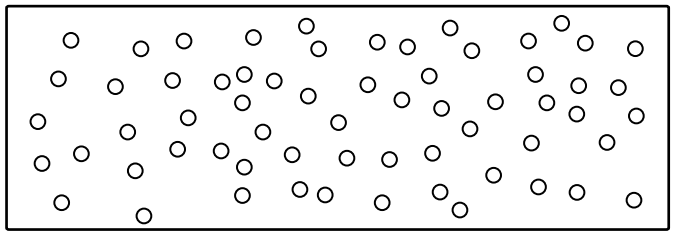


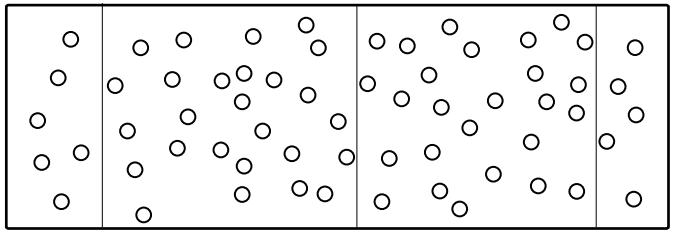
Requirements of a good partition

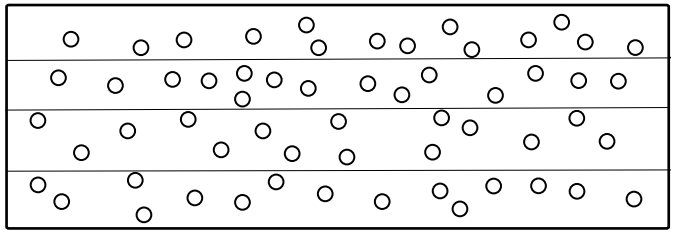




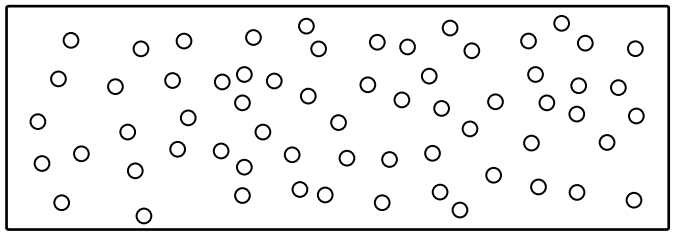
Requirements of a good partition

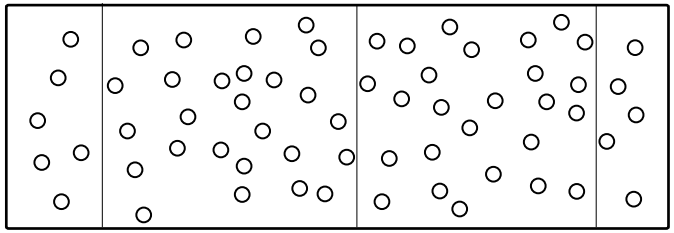


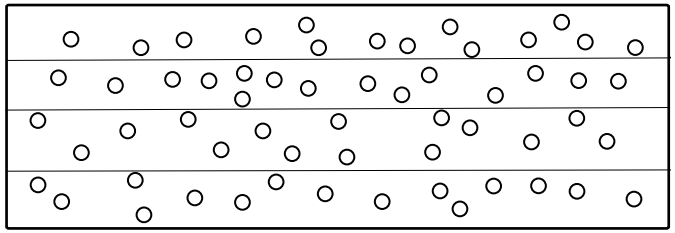


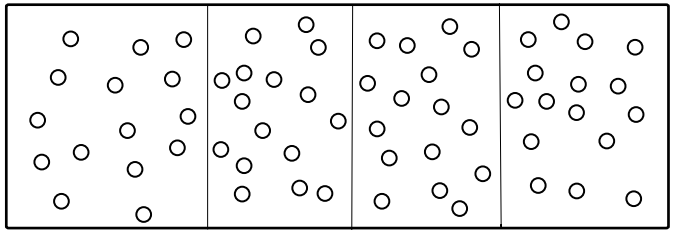


Requirements of a good partition









Outline

Scope of this work

Algorithms

- Geometric Partitioning Algorithms
- Recursive Coordinate Bisection (RCB)
- Recursive Inertial Bisection (RIB)
- Hilbert Space Filling Curves (HSFC)

Scope of this work

Algorithms

- Geometric Partitioning Algorithms
- Recursive Coordinate Bisection (RCB)
- Recursive Inertial Bisection (RIB)
- Hilbert Space Filling Curves (HSFC)

Applications

- Free surface flows
- Fixed domain incompressible NS
- Elastic solid mechanics

Scope of this work

Algorithms

- Geometric Partitioning Algorithms
- Recursive Coordinate Bisection (RCB)
- Recursive Inertial Bisection (RIB)
- Hilbert Space Filling Curves (HSFC)

Applications

- Free surface flows
- Fixed domain incompressible NS
- Elastic solid mechanics

- Partitioning quality
- Execution times
- Scale-up

Algorithm classification

Geometric partitioners

- Physical coordinates as input
- Most general for numerical work
- Natural for particle methods

Algorithm classification

Geometric partitioners

- Physical coordinates as input
- Most general for numerical work
- Natural for particle methods

Examples

- Recursive Coordinate Bisection (RCB)
- Recursive Inertial Bisection (RIB)
- Space Filling Curves (SFC)

Algorithm classification

Graph partitioners

- Data represented as a graph
- Inherently suitable for mesh-based methods
- Cell based graph-partitioning may be used for SPH

Algorithm classification

Graph partitioners

- Data represented as a graph
- Inherently suitable for mesh-based methods
- Cell based graph-partitioning may be used for SPH

Examples

- METIS/ParMETIS
- PTScotch
- Hypergraph partitioning

Recursive Coordinate Bisection (RCB)

Algorithm and Advantages

- Recursive
- Orthogonal cuts
- Fast

Recursive Coordinate Bisection (RCB)

Algorithm and Advantages

- Recursive
- Orthogonal cuts
- Fast

Disadvantages

- Not rotationally invariant
- Stretched halo regions

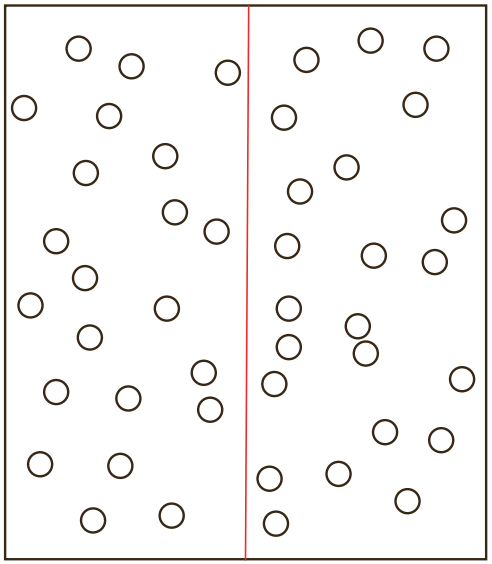
Recursive Coordinate Bisection (RCB)

Algorithm and Advantages

- Recursive
- Orthogonal cuts
- Fast

Disadvantages

- Not rotationally invariant
- Stretched halo regions



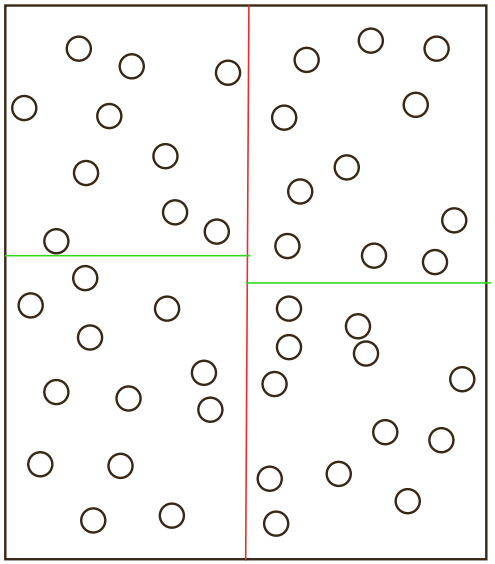
Recursive Coordinate Bisection (RCB)

Algorithm and Advantages

- Recursive
- Orthogonal cuts
- Fast

Disadvantages

- Not rotationally invariant
- Stretched halo regions



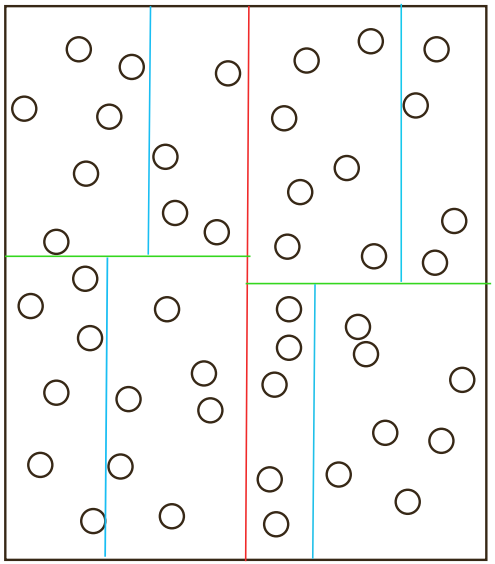
Recursive Coordinate Bisection (RCB)

Algorithm and Advantages

- Recursive
- Orthogonal cuts
- Fast

Disadvantages

- Not rotationally invariant
- Stretched halo regions



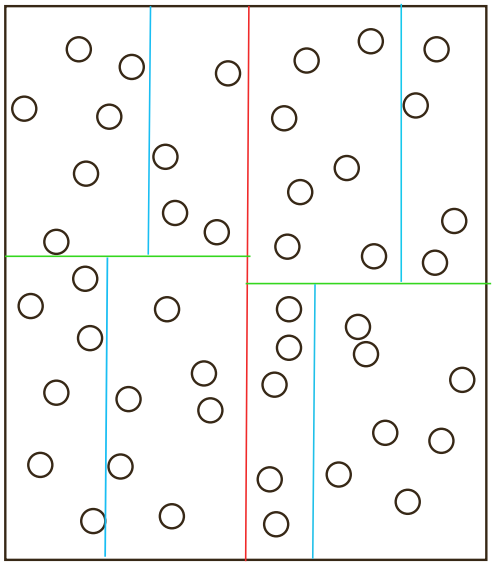
Recursive Coordinate Bisection (RCB)

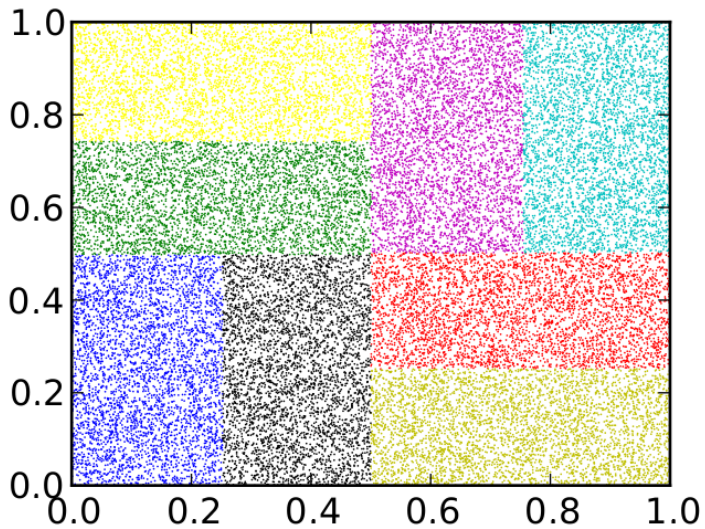
Algorithm and Advantages

- Recursive
- Orthogonal cuts
- Fast

Disadvantages

- Not rotationally invariant
- Leads to stretched halo-regions





Recursive Inertial Bisection (RIB)

Algorithm and Advantages

- Variant of RCB
- Orthogonal cuts to principal inertial axes
- Adaptive to rotations
- Lesser communication overhead

Recursive Inertial Bisection (RIB)

Algorithm and Advantages

- Variant of RCB
- Orthogonal cuts to principal inertial axes
- Adaptive to rotations
- Lesser communication overhead

Disadvantages

- Eigenvector computations

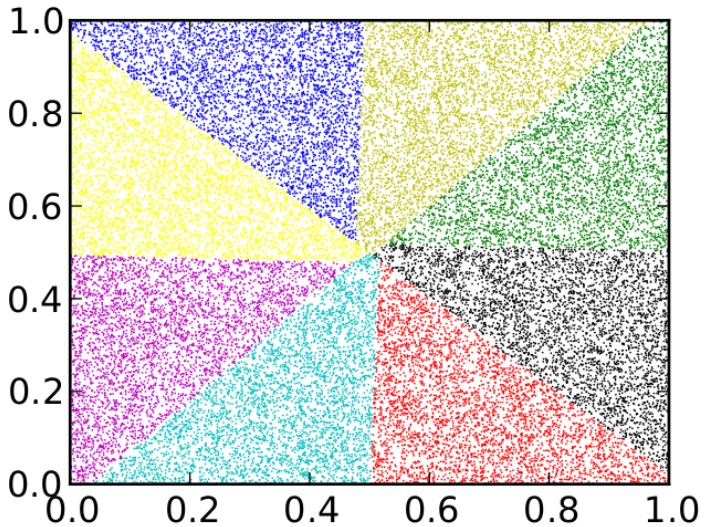
Recursive Inertial Bisection (RIB)

Algorithm and Advantages

- Variant of RCB
- Orthogonal cuts to principal inertial axes
- Adaptive to rotations
- Lesser communication overhead

Disadvantages

- Eigenvector computations



Hilbert Space Filling Curves (HSFC)

Algorithm and Advantages

- Use a SFC $f : R^3 \longrightarrow R$
- Order objects linearly
- Geometric locality

Hilbert Space Filling Curves (HSFC)

Algorithm and Advantages

- Use a SFC $f : R^3 \longrightarrow R$
- Order objects linearly
- Geometric locality

Disadvantages

- Particle distribution has *projections*
- Disconnected regions for complex geometries

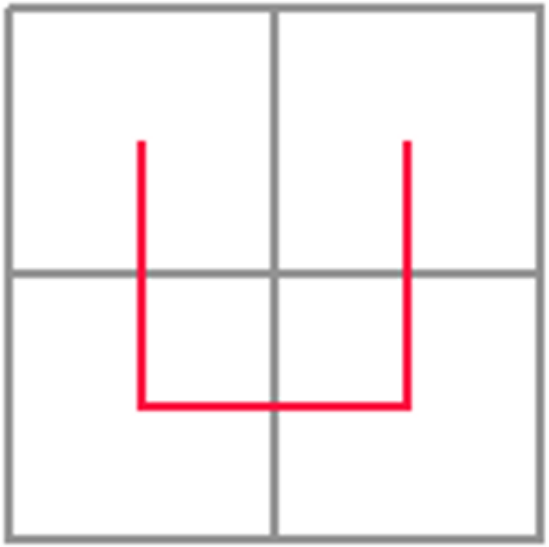
Hilbert Space Filling Curves (HSFC)

Algorithm and Advantages

- Use a SFC $f : R^3 \longrightarrow R$
- Order objects linearly
- Geometric locality

Disadvantages

- Particle distribution has *projections*
- Disconnected regions for complex geometries



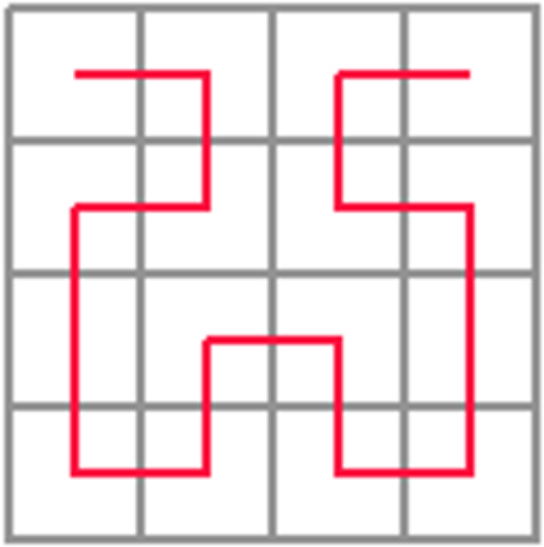
Hilbert Space Filling Curves (HSFC)

Algorithm and Advantages

- Use a SFC $f : R^3 \longrightarrow R$
- Order objects linearly
- Geometric locality

Disadvantages

- Particle distribution has *projections*
- Disconnected regions for complex geometries



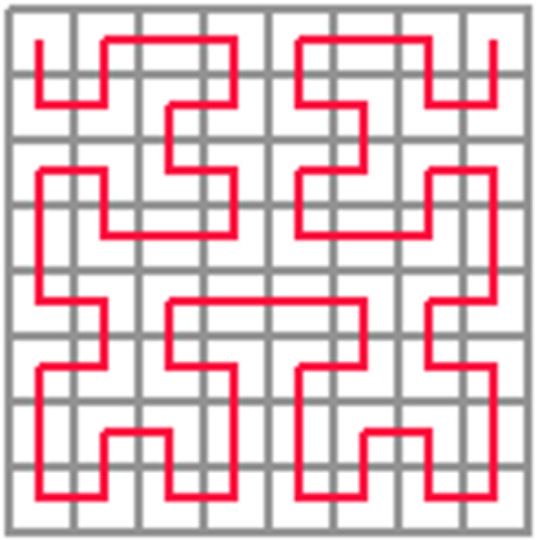
Hilbert Space Filling Curves (HSFC)

Algorithm and Advantages

- Use a SFC $f : R^3 \longrightarrow R$
- Order objects linearly
- Geometric locality

Disadvantages

- Particle distribution has *projections*
- Disconnected regions for complex geometries



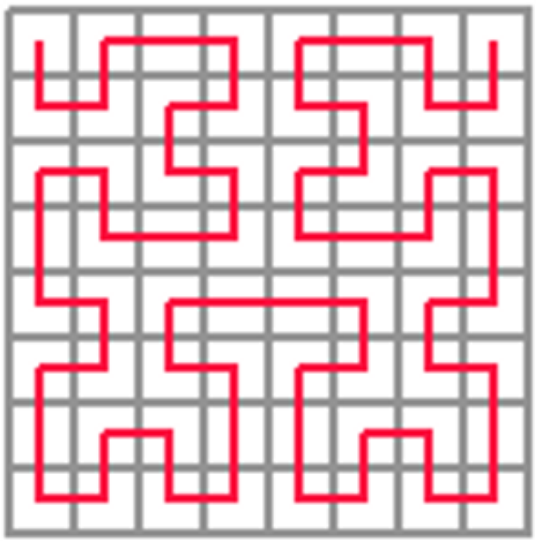
Hilbert Space Filling Curves (HSFC)

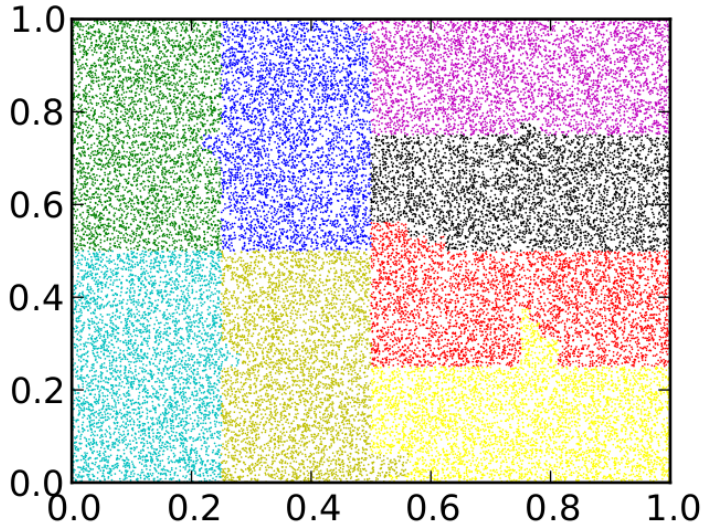
Algorithm and Advantages

- Use a SFC $f : R^3 \longrightarrow R$
- Order objects linearly
- Geometric locality

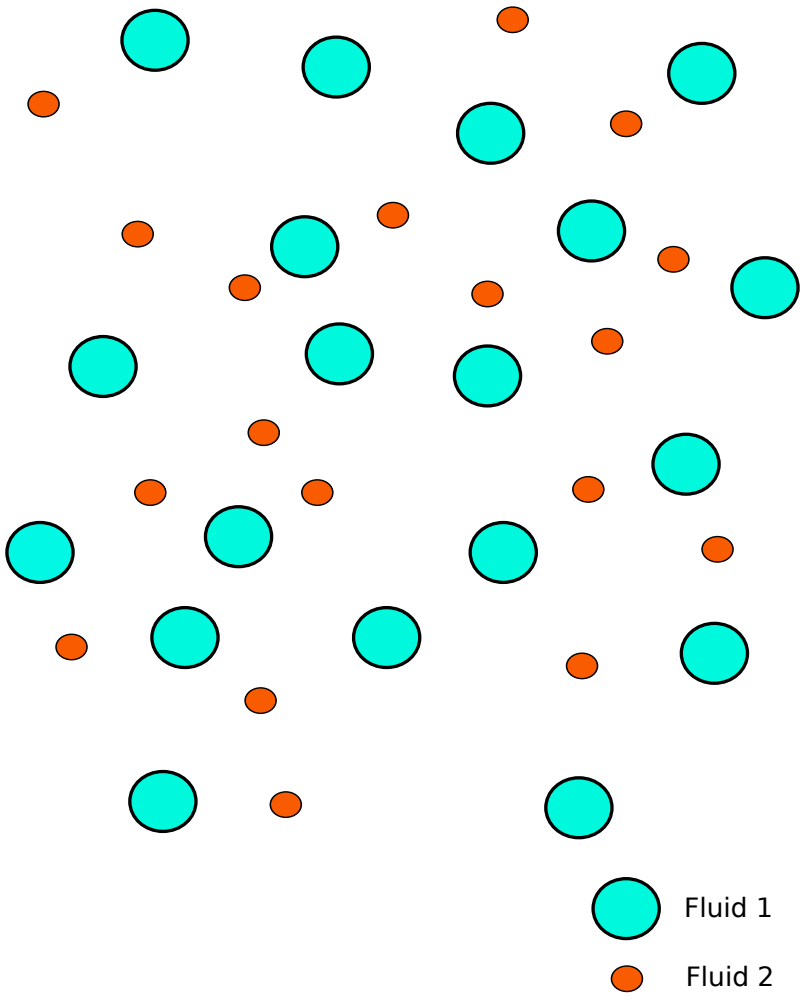
Disadvantages

- Particle distribution has *projections*
- Disconnected regions for complex geometries

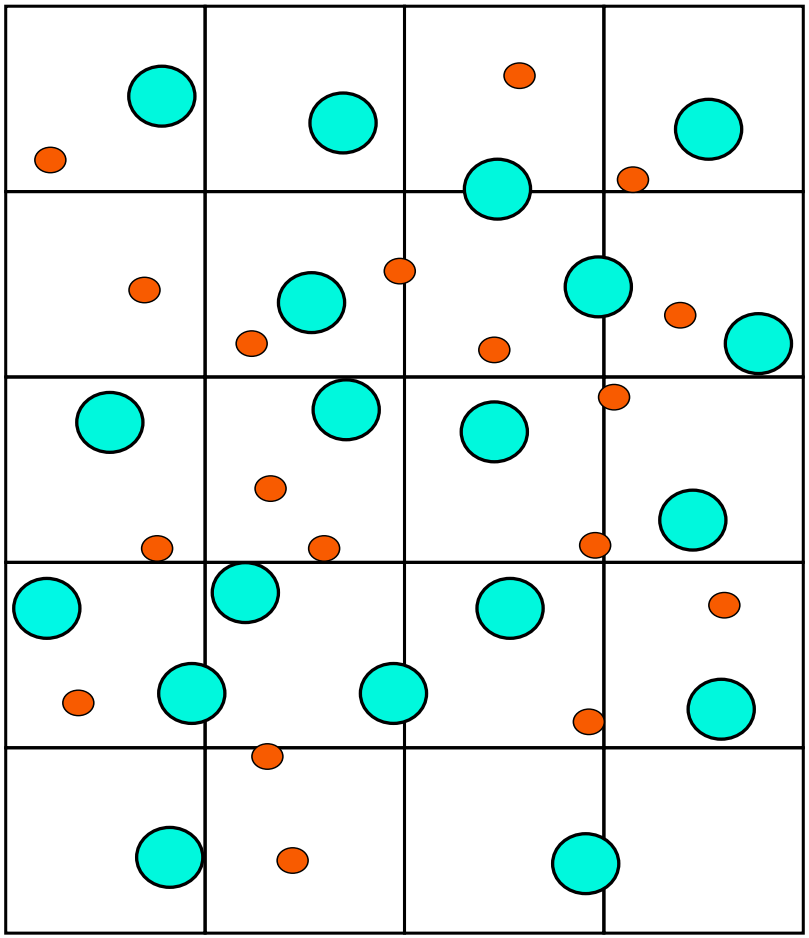




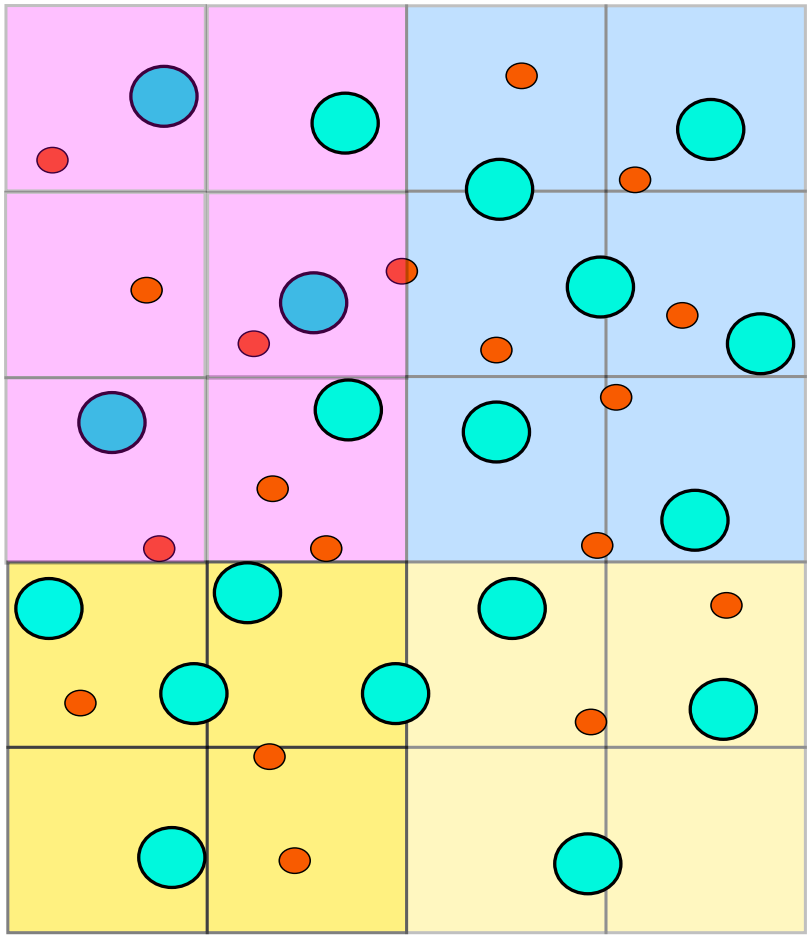
How do we do it?



How do we do it?



How do we do it?



Zoltan Data Management Library

What is it?

- Developed by Sandia National Laboratories
- Trilinos Project (9.0 September 2008)
- Zoltan v3.6 released in September 2011

What can it do?

- Dynamic Load Balancing
- Graph Coloring
- Dynamic memory management

Outline

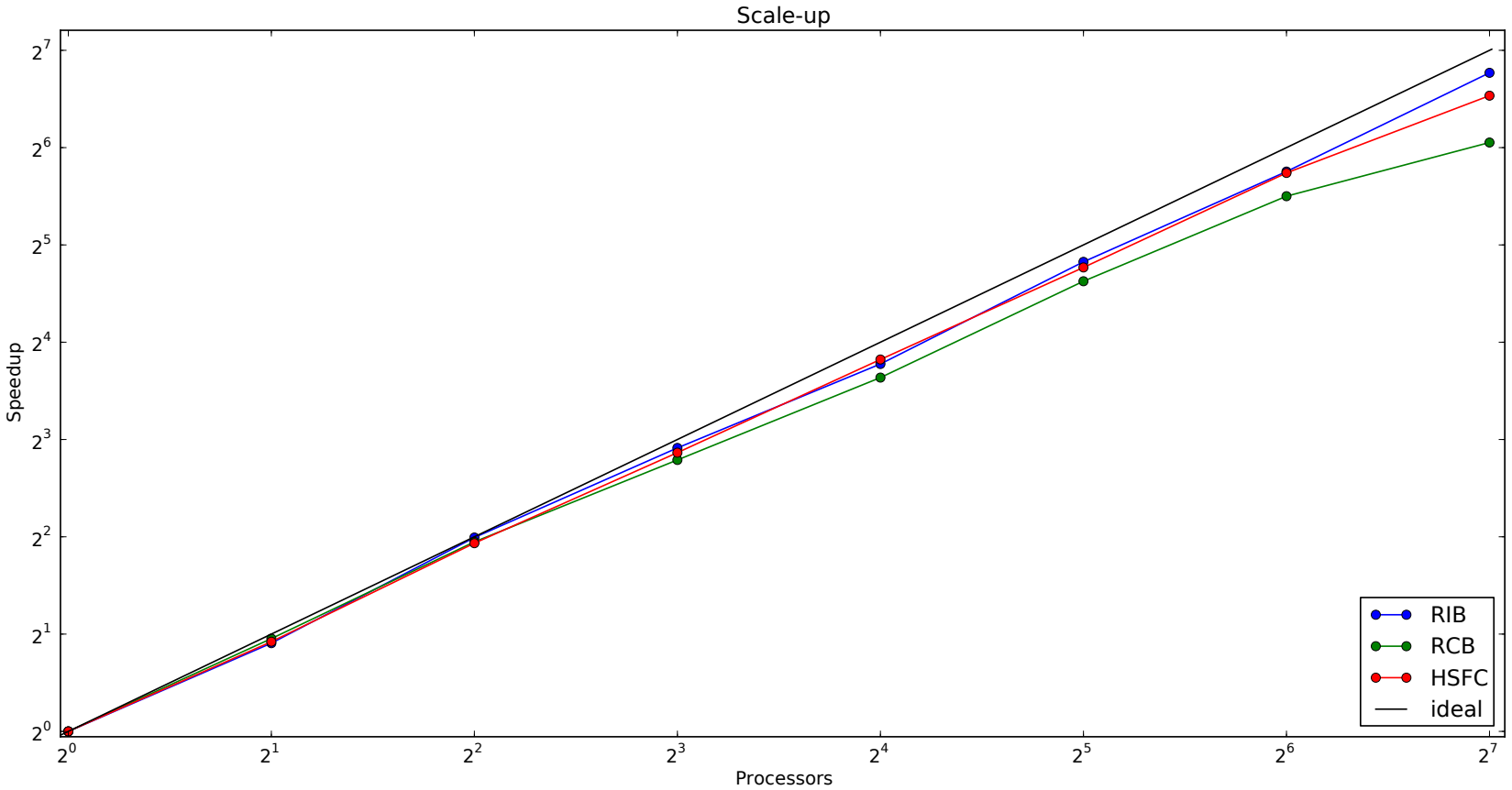
2D Dam Break

- $N_p \approx O(0.1M)$

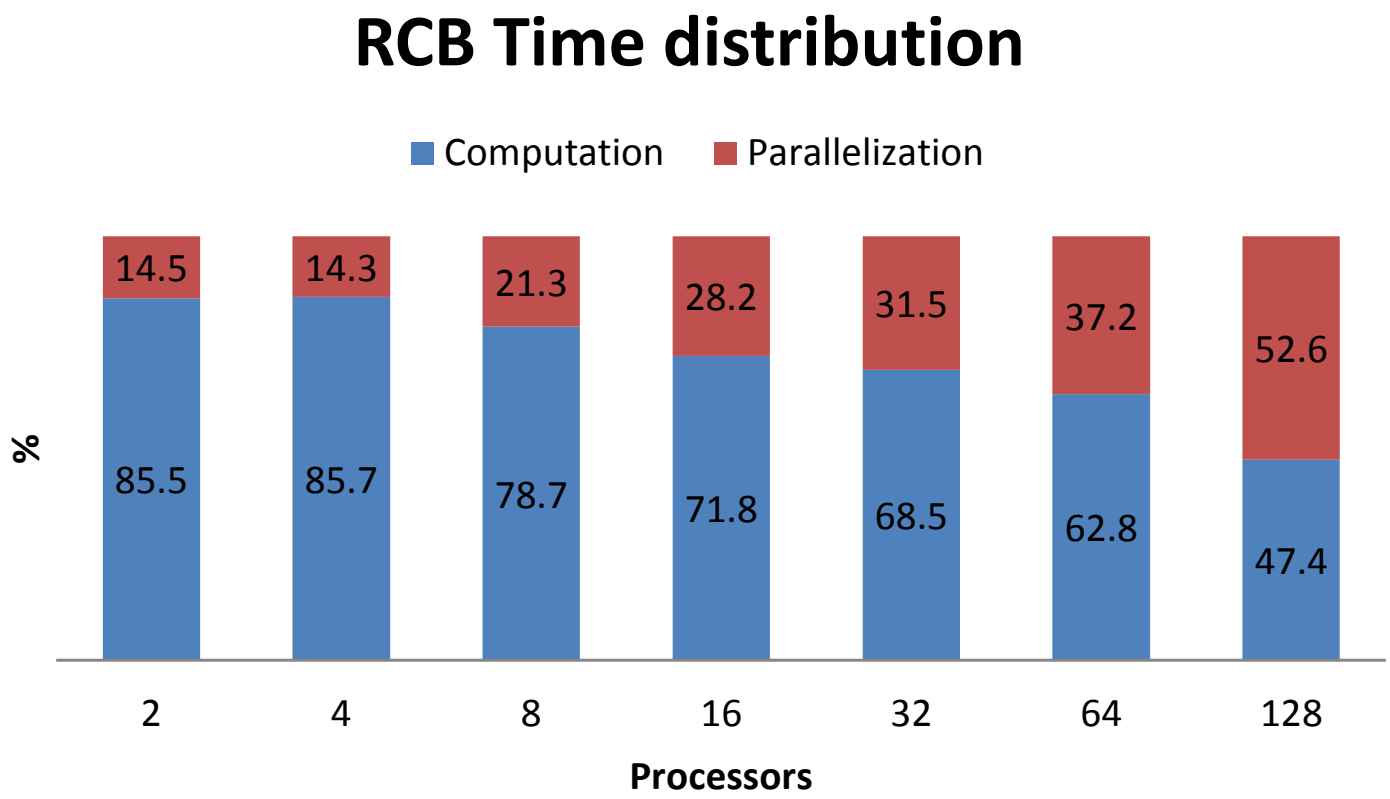
Machine Architecture

- Linux (CentOS) cluster
- Six-core AMD Opteron
- 1 Gigabit Ethernet interconnect
- 12 GB RAM per node

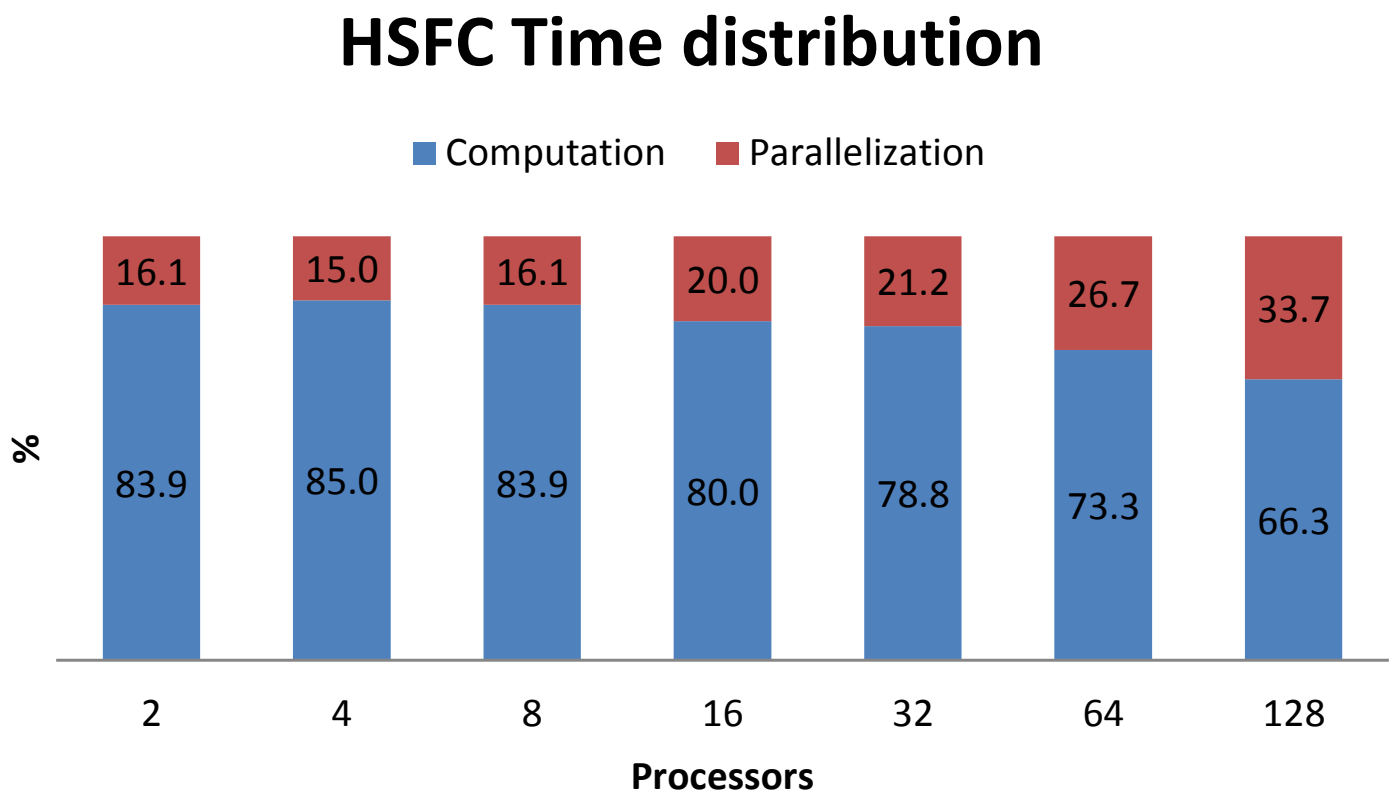
Scale-up : $N_p \approx 10$M, per-iteration



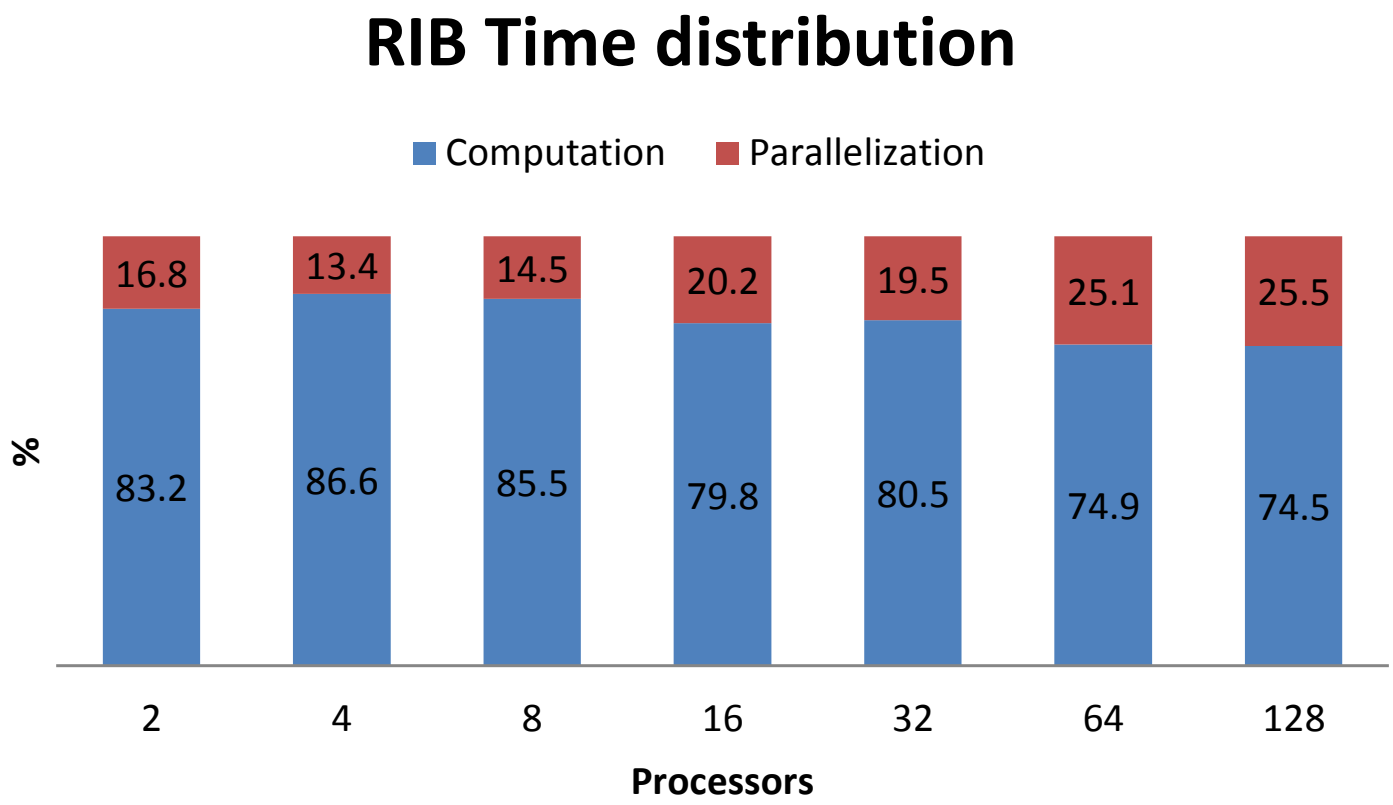
Time distribution : RCB



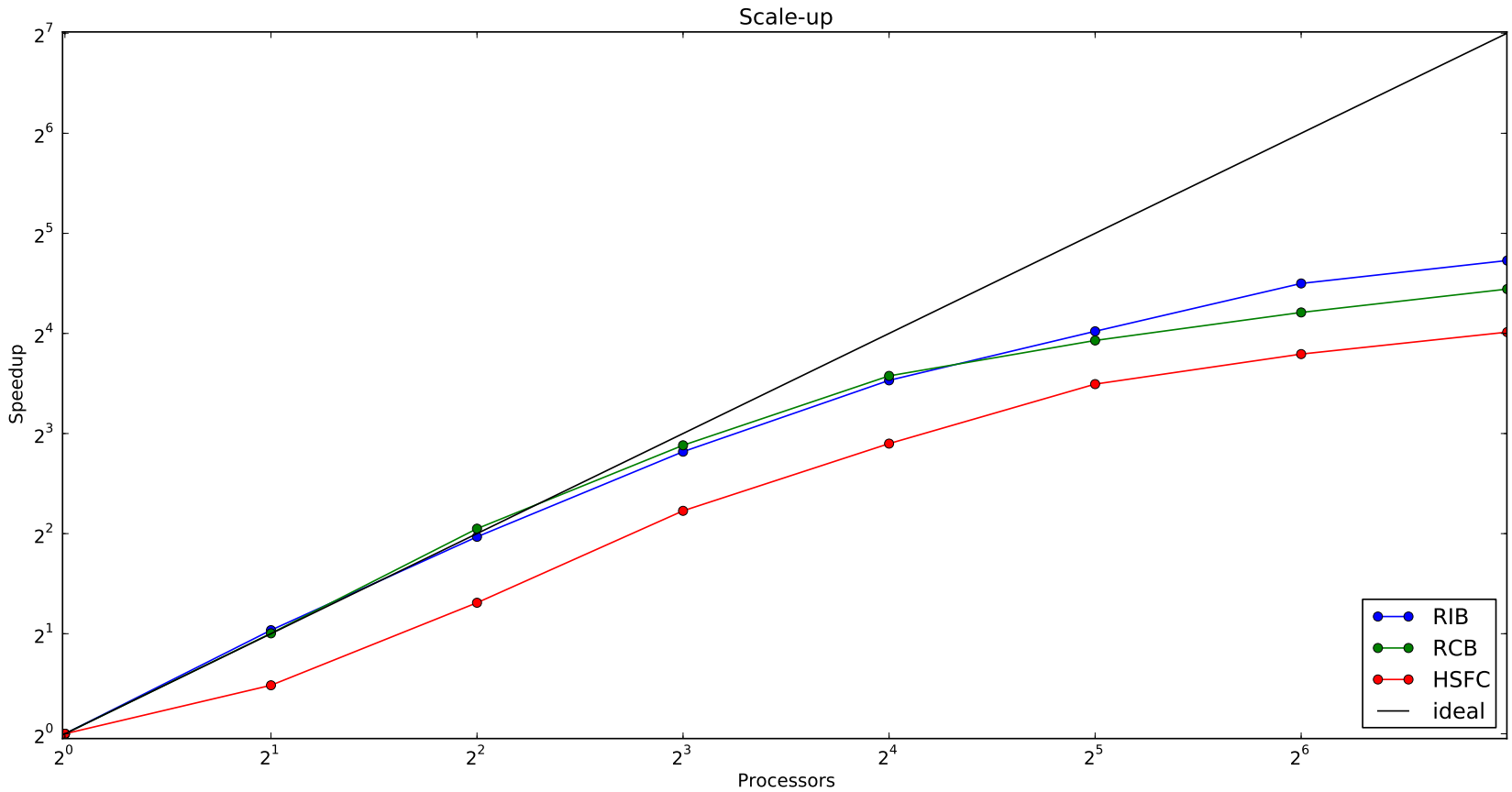
Time distribution : HSFC



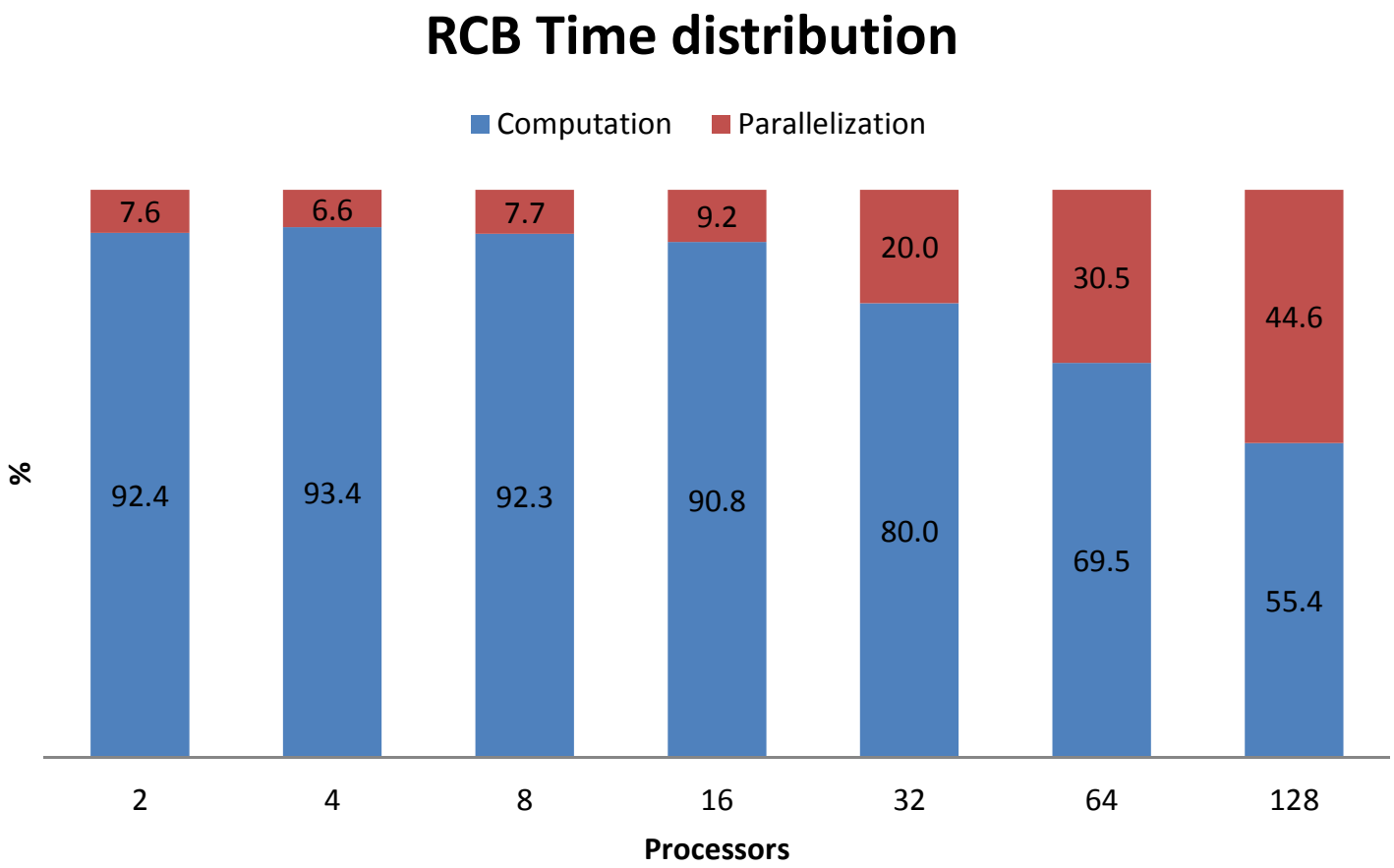
Time distribution : RIB



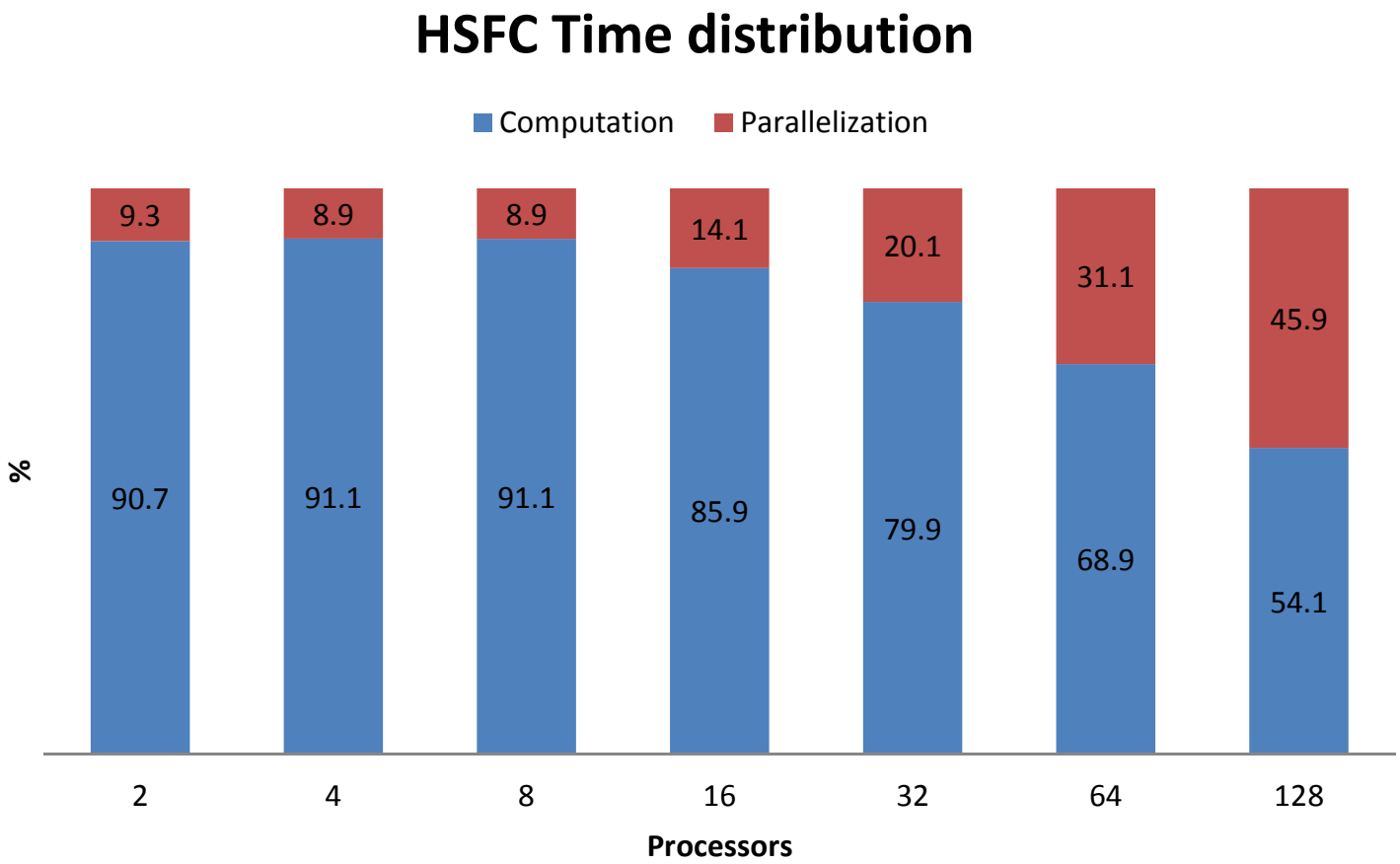
3D Dam break : Scale-up : $N_p \approx 10$M, per-iteration



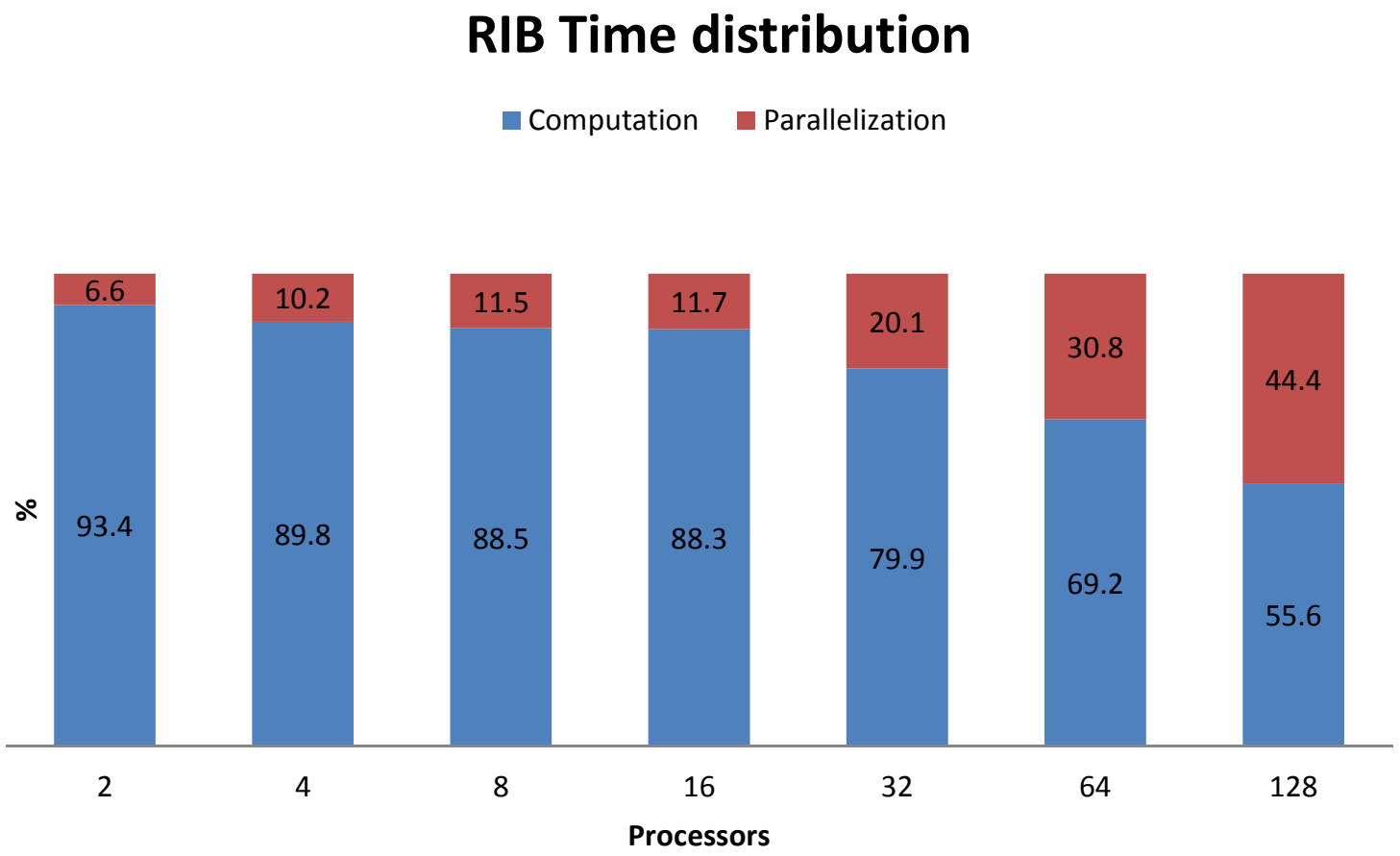
3D Dam break : RCB Time distribution



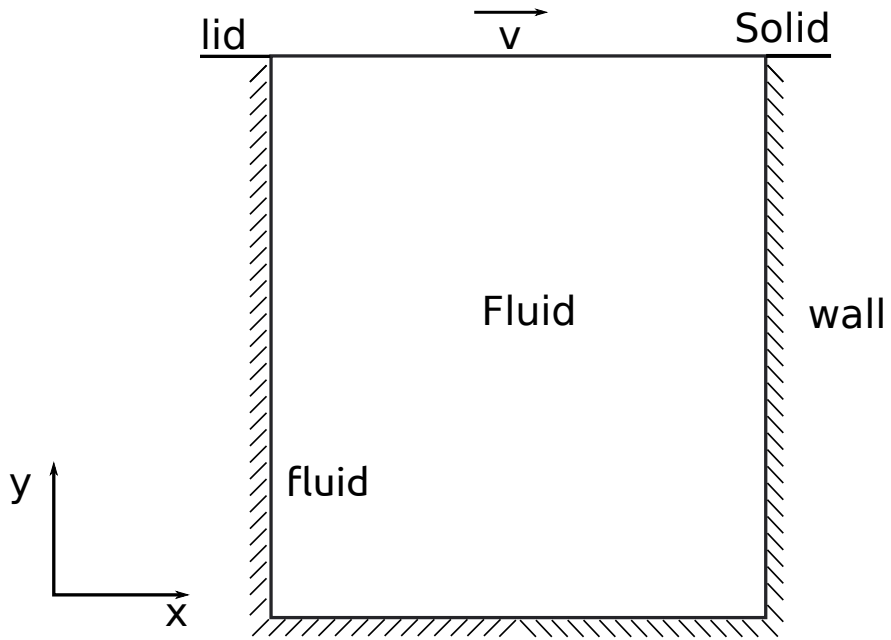
3D Dam break : HSFC Time distribution



3D Dam break : RIB Time distribution

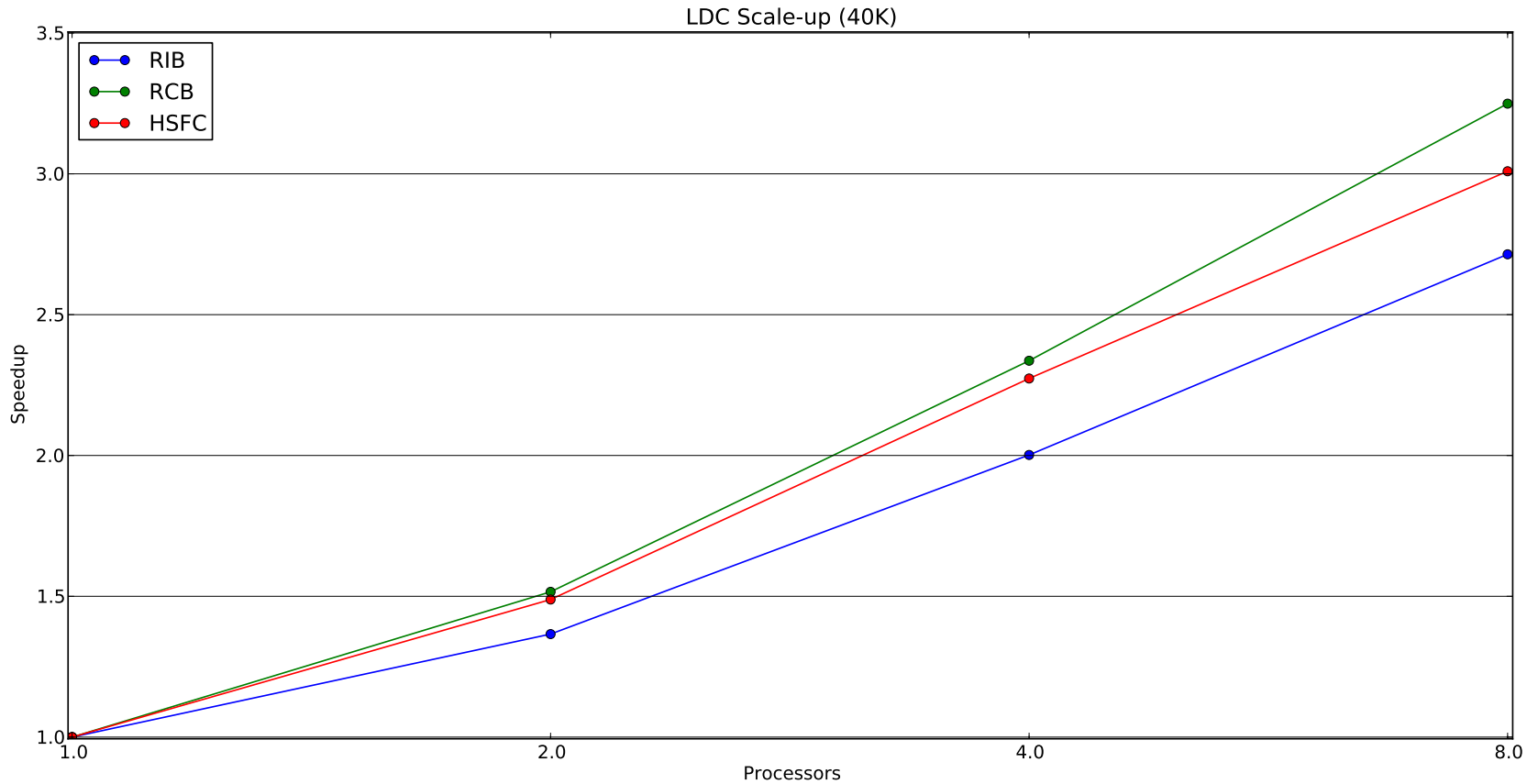


Lid Driven Cavity

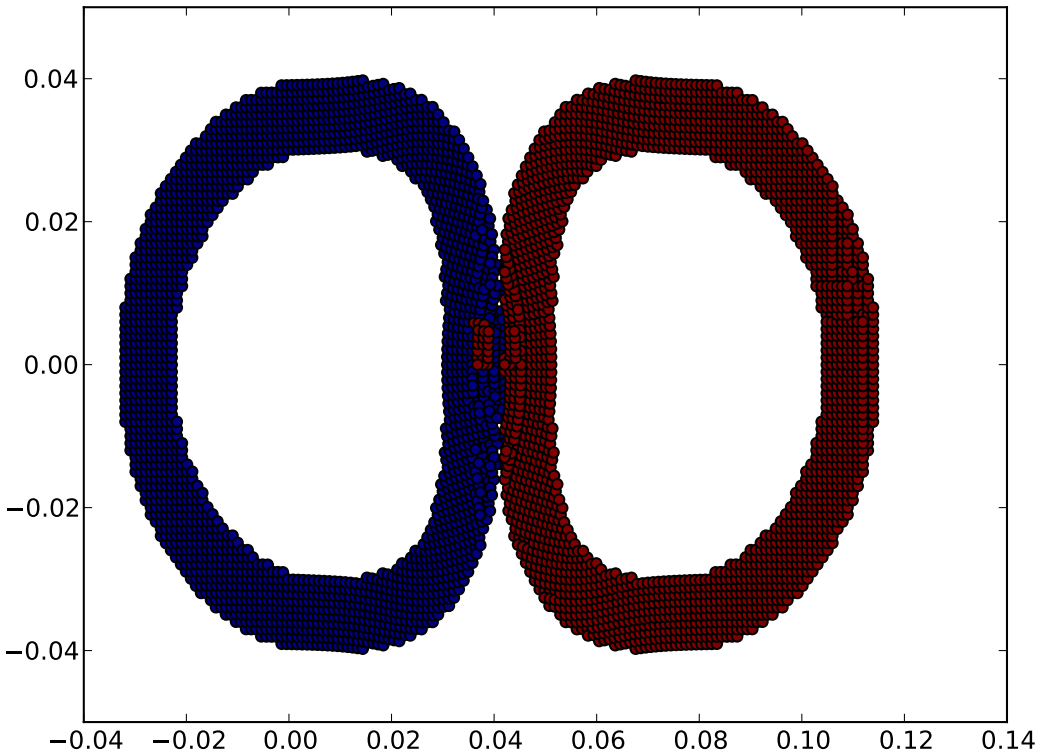


- Fixed $2D$ domain
- $N_p \leq 10^5$

Scale up

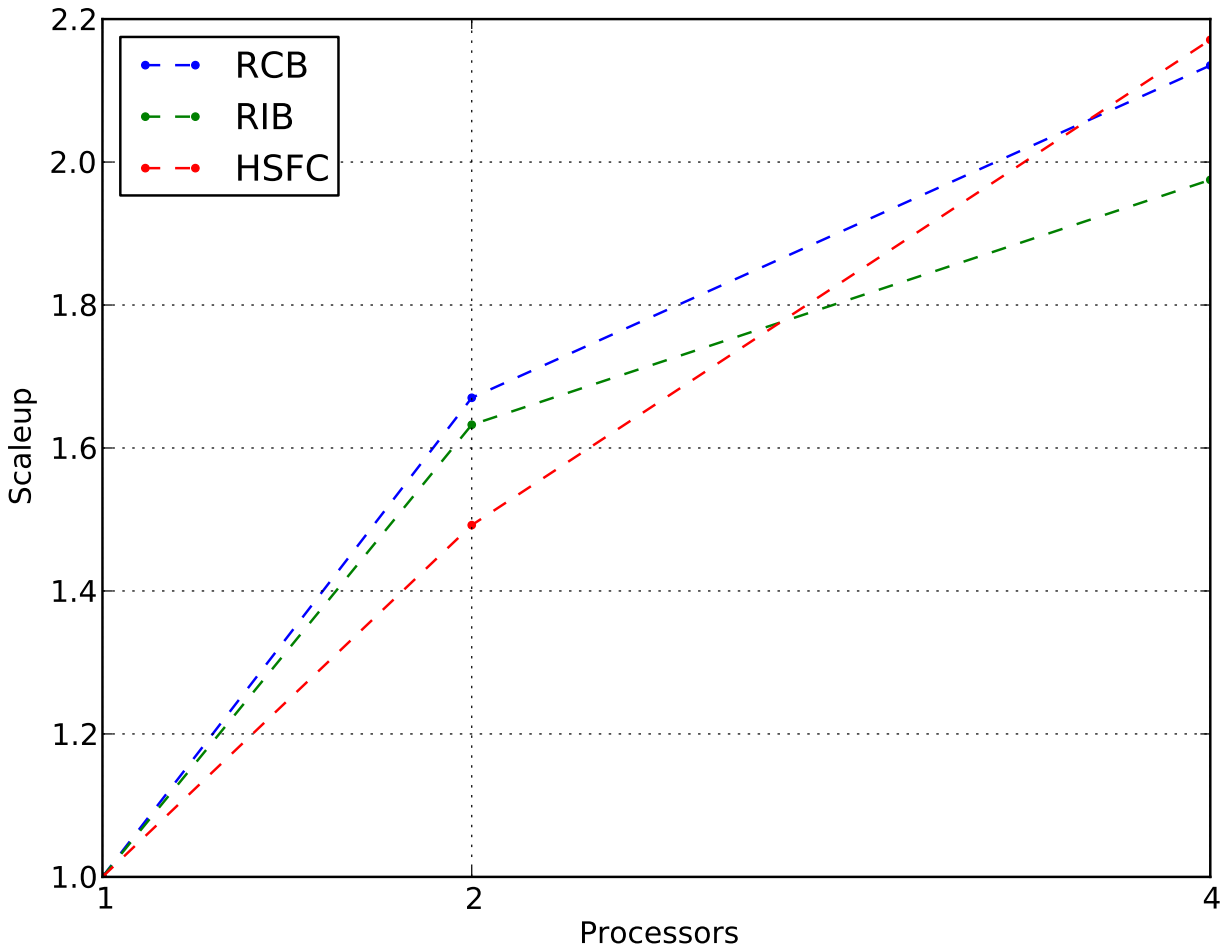

LDC Scale-up (40K)
RIB
RCB
HSFC
Speedup
Processors
3.5
3.0
2.5
2.0
1.5
1.0
1.0
2.0
4.0
8.0

Elastic collision



- Collision
- $N_p \approx 20,000$

Scale up



Conclusions

- Evaluation of geometric load balancing algorithms
- RIB for dynamic free surfaces
- RCB/HSFC for contact problems
- RCB for fixed domain problems

Future work

- HVI and 3D contact problems
- Scale up for massively parallel computers
- Coupled methods

Thank you!