Parallel Fast N-Body

HPCSE II - Spring 2015

Last time

- N-body solvers for long range interactions
  - Naive: O(N^2)
  - Idea: treat cluster of far away particles as one
    - Barnes-Hut: O(N logN)
    - Fast Multipole Method: O(N)

The Barnes-Hut algorithm

- Asymptotic complexity: O(N logN)
- BUT much higher implementation complexity

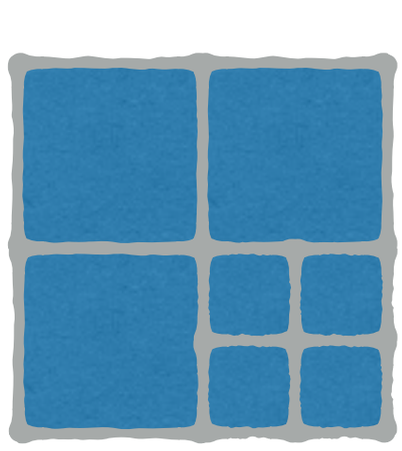
```
tree building

multipole expansions

tree traversal

force summation
```

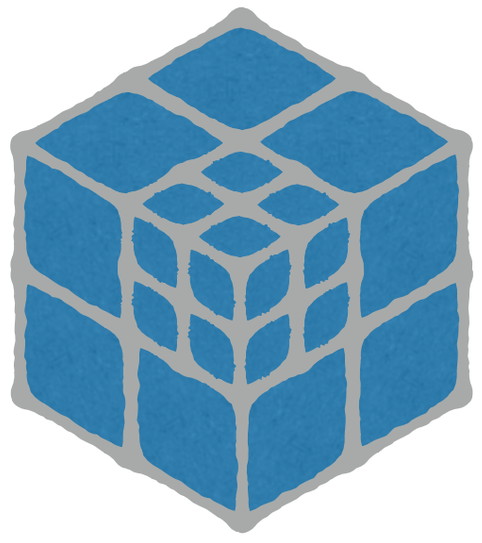
VS

```
force summation
```

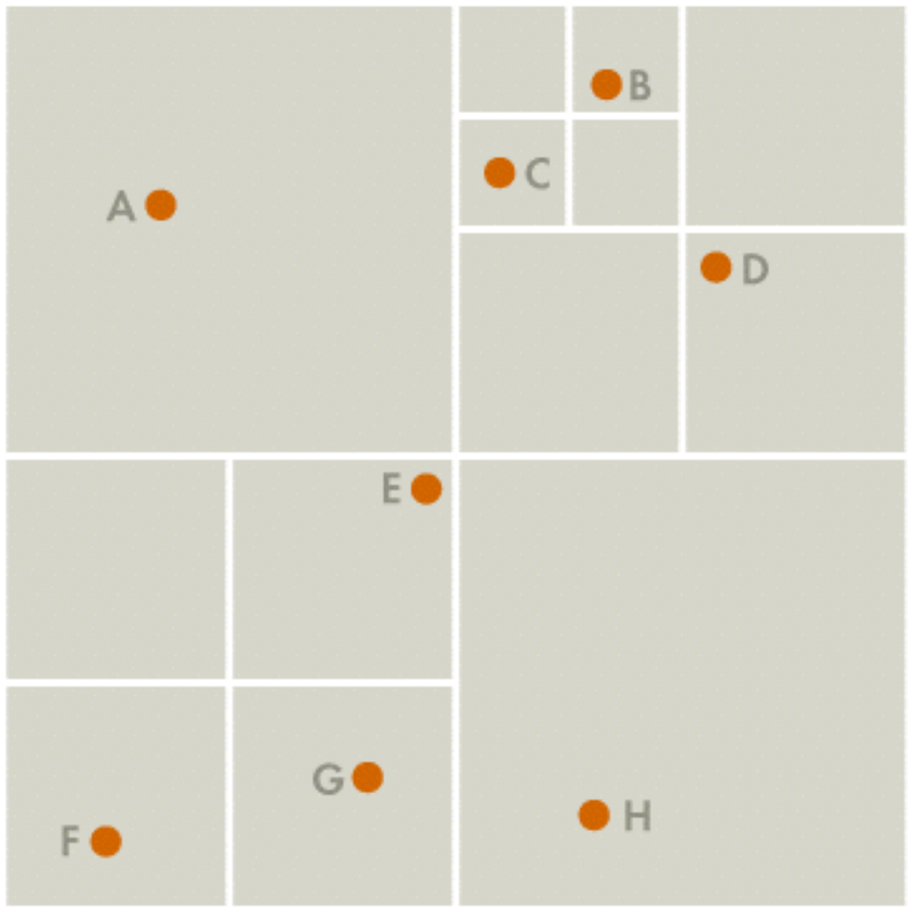
Data structure

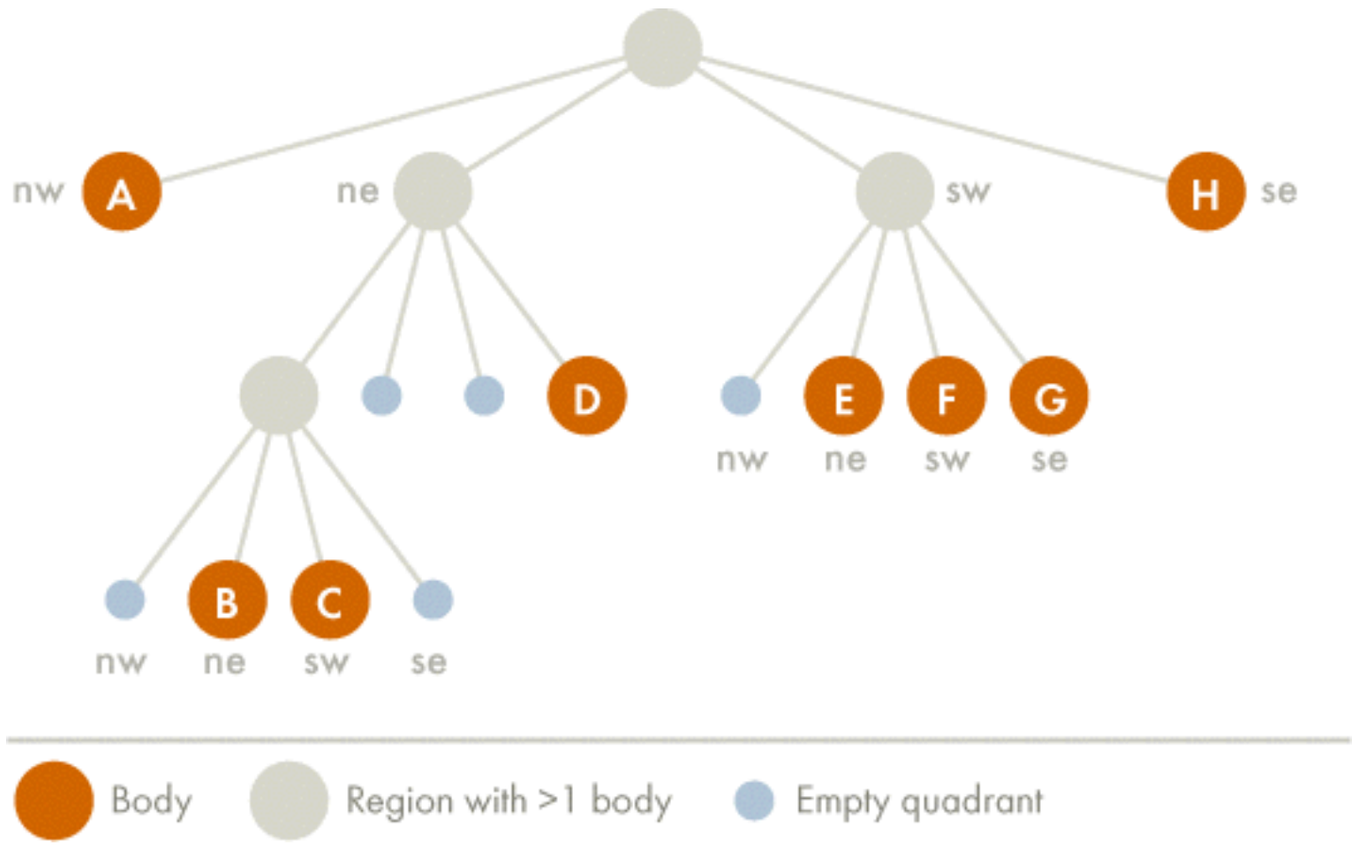
- Requirements:
  - Hierarchical spatial decomposition
  - Simple to build and traverse
- Extension of the binary tree in multiple dimensions:
  - Quadtree (2D)
  - Octree (3D)





Quadtree





pictures from http://arborjs.org/docs/barnes-hut

Building the quadtree

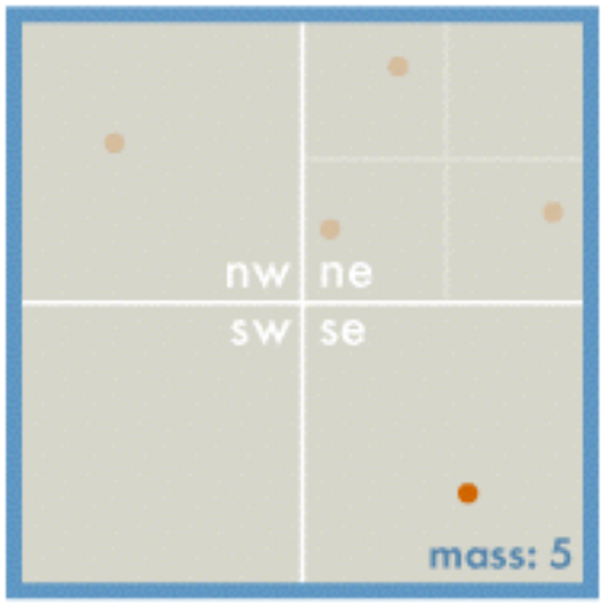
- Top-down approach
- Particles are inserted in an initially empty tree
- 3 types of nodes:
  - internal - contain multiple leaves
  - leaf - contains a single particle
  - empty

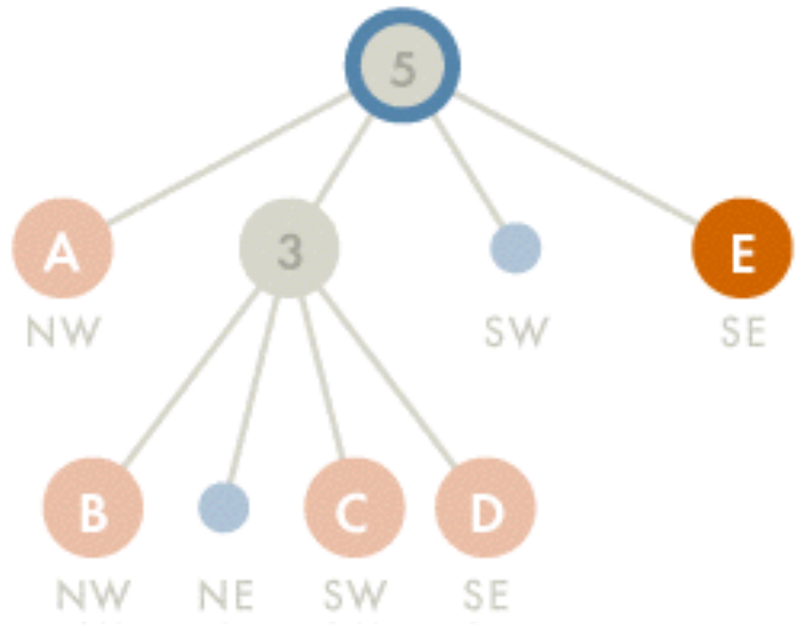
Building the quadtree

```
for each particle

    current_node = root_node

    while (true)

        if (current node != empty && !internal)

            insert particle, exit loop

    else if (current_node == internal)

        put into corresponding quadrant of current_node

    else

        split current_node into its children (one for each quadrant)

        put particles into their quadrants nodes
```

Building the quadtree

Add body E
fits in se corner of root node





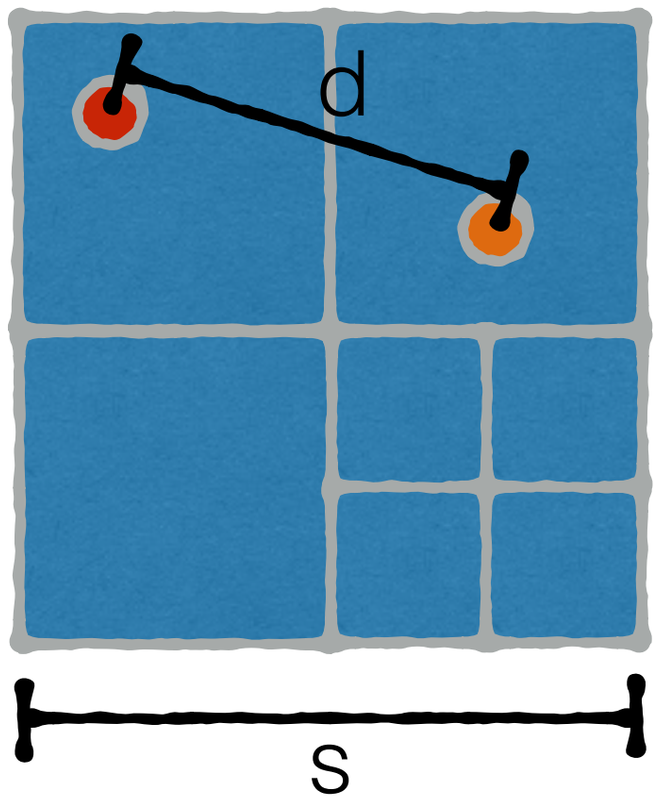
pictures from http://arborjs.org/docs/barnes-hut

Multipole expansions

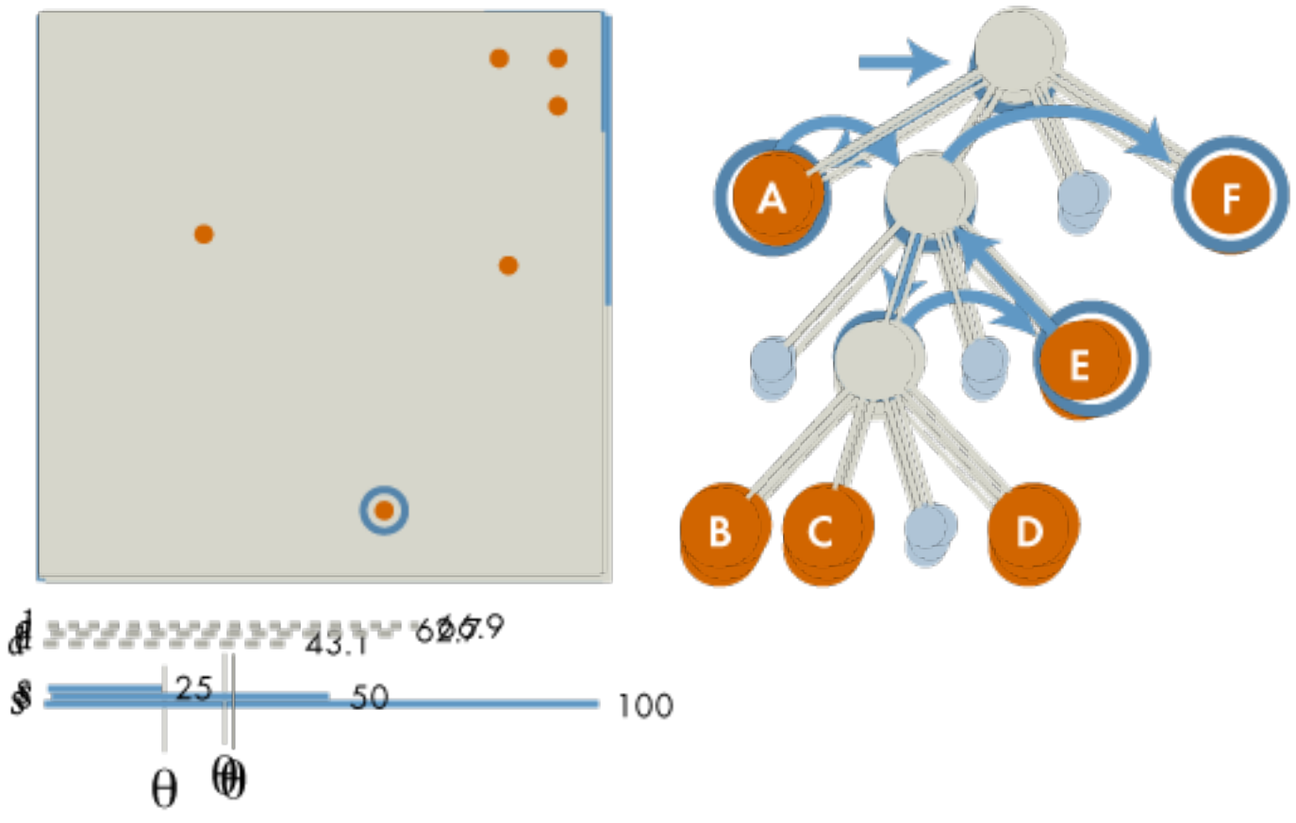
- Multipole expansions computations
  - Option 1: while building the tree
    (more difficult to parallelize)
  - Option 2: after building the tree
    (bottom-up, “reduction” operation)

Traversing the tree

```
for each particle

   current_node = root_node

   if (current_node==leaf && particle contained not self)

      compute force, add to self

else if (s/d<theta)

   compute force with internal node

else

   follow this procedure on each child
```



Traversing the tree



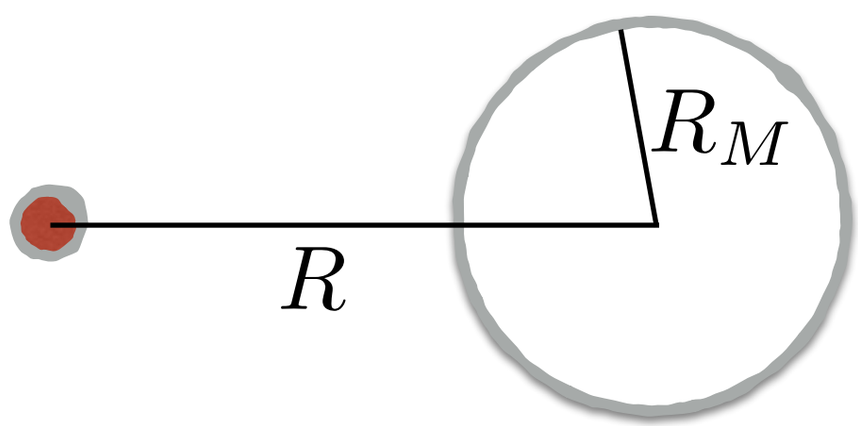
pictures from http://arborjs.org/docs/barnes-hut

Fast multipole method

- Particle-box interactions
- Box-box interactions
- Convergence rate proportional to

$$\left(\frac{R_M}{R}\right)^{P+1}$$

P multipole moments expansions

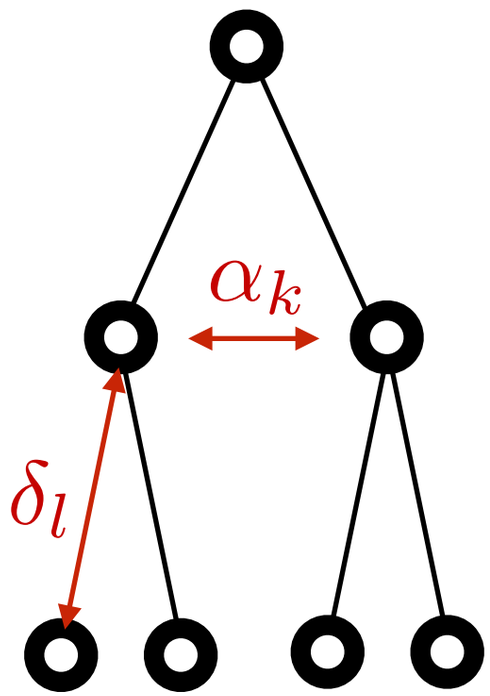


Box-box interactions

- Interactions computed at the coarsest level

$$\delta_l = \frac{(-1)^{l+1}}{(Z_G - Z_M)^l} \sum_{k=0}^{P} \binom{l+k-1}{k} \frac{\alpha_k}{(Z_G - Z_M)^k)}$$

$$l = 1, \ldots, P$$



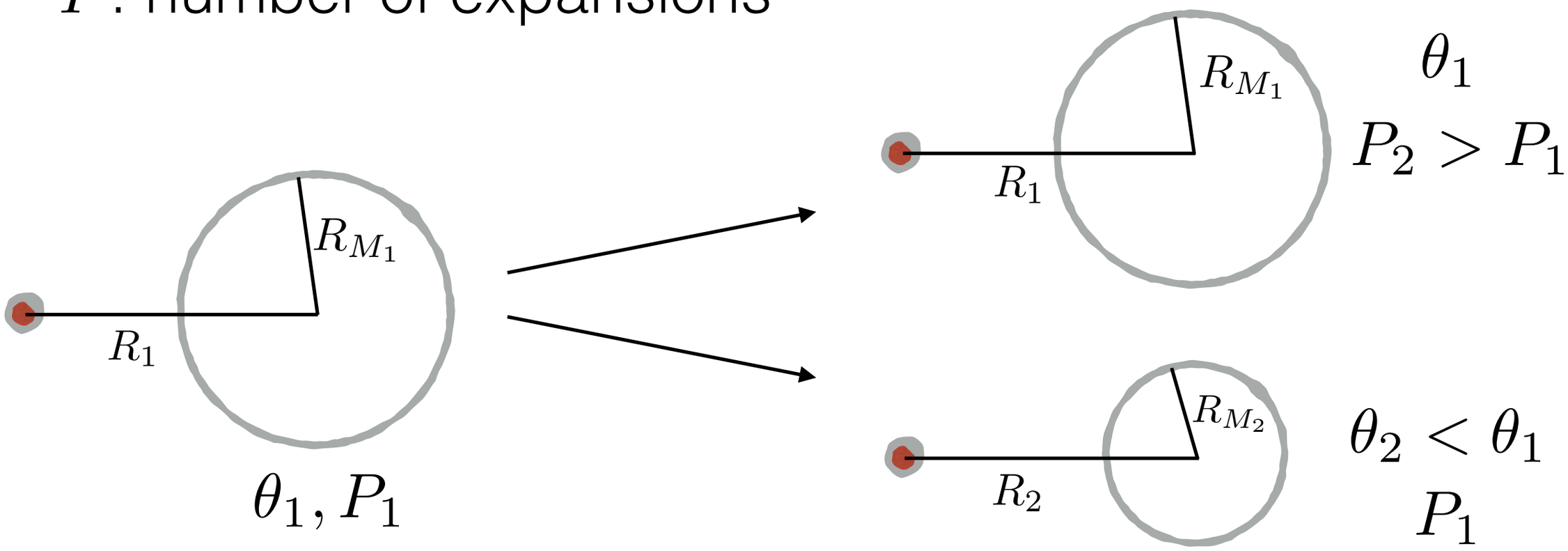
- Velocities computed at finest level

$$\delta_l^{\text{children}} = \sum_{k=l}^{P} \binom{k-1}{k-l} \delta_k^{\text{parent}} \left(Z_M^{\text{children}} - Z_M^{\text{parent}}\right)^{k-l}$$

$$V(Z) = \frac{i}{2\pi} \sum_{l=1}^{P} \delta_l (Z - Z_G)^{l-1}$$

Accuracy considerations

- Multiple parameters affect accuracy
  - $\theta = \frac{R_M}{R}$: ratio of box size to distance
  - P: number of expansions



1 Gpc/h

Millennium Simulation

10.077.696.000 particles

(z = 0)

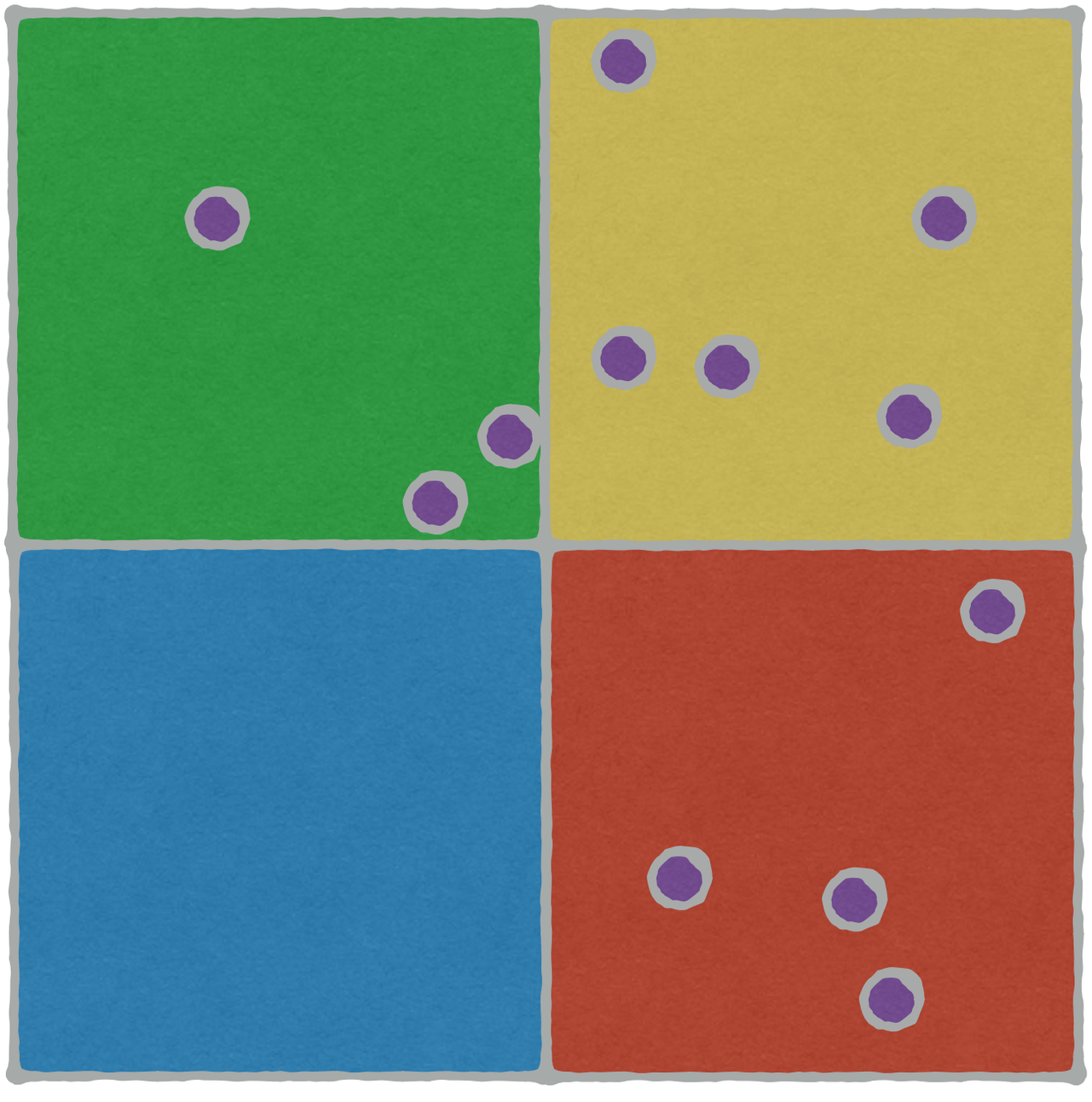
Parallelization of BH

- Material based on Gibbon et al, Parallel Tree Codes (http://juser.fz-juelich.de/record/16155/files/IAS_Series_06.pdf)
- Irregular code
  - How to distribute and define the work?
  - How to minimize communication in MPI?

Parallel algorithm

```
for t in time

  (i)   perform domain decomposition

  (ii)  build tree

  (iii) traverse tree and build
        interaction list

  (iv)  perform force summation
```

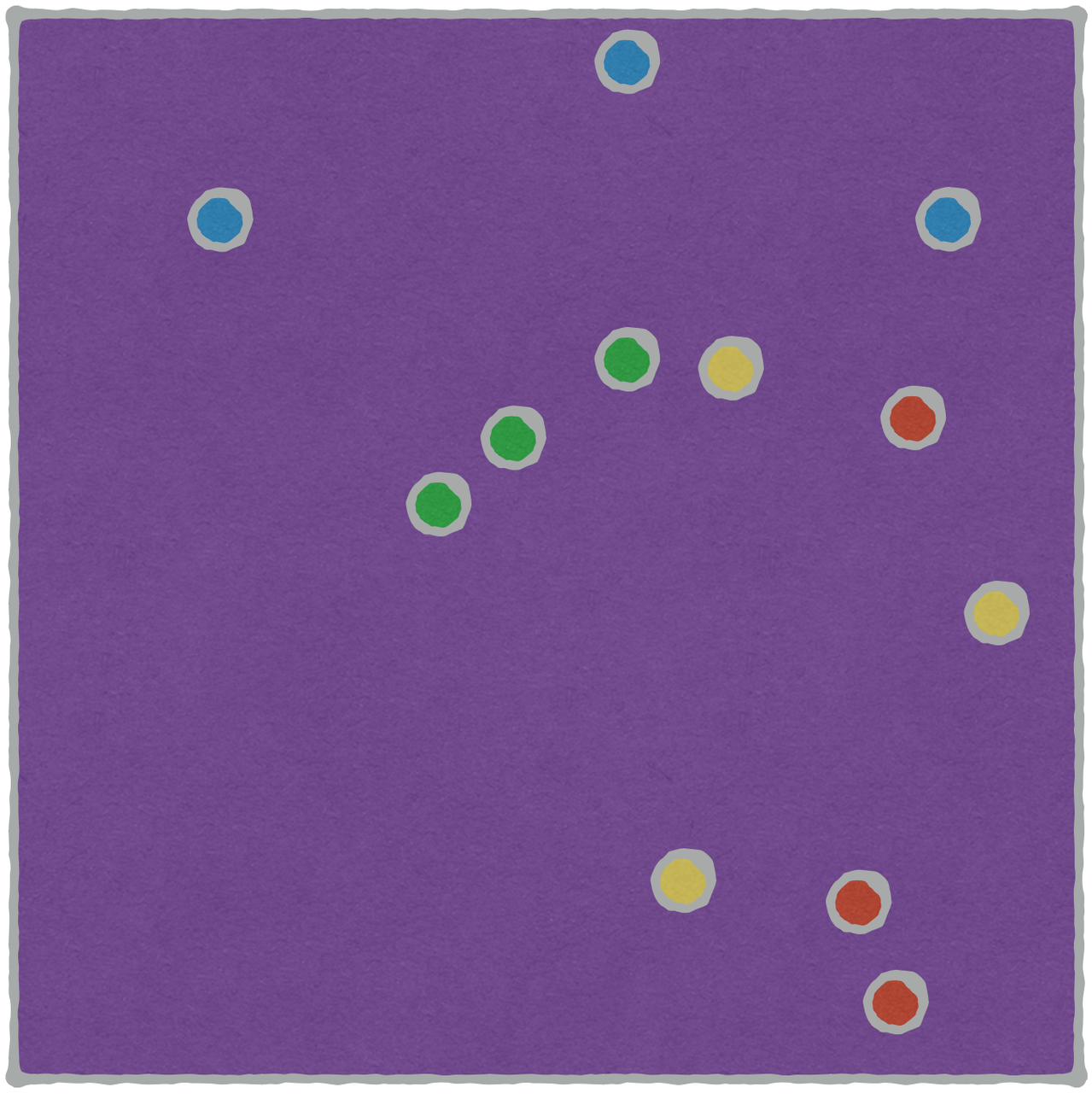
Domain decomposition



✓ Simple

x Highly unbalanced

Domain decomposition



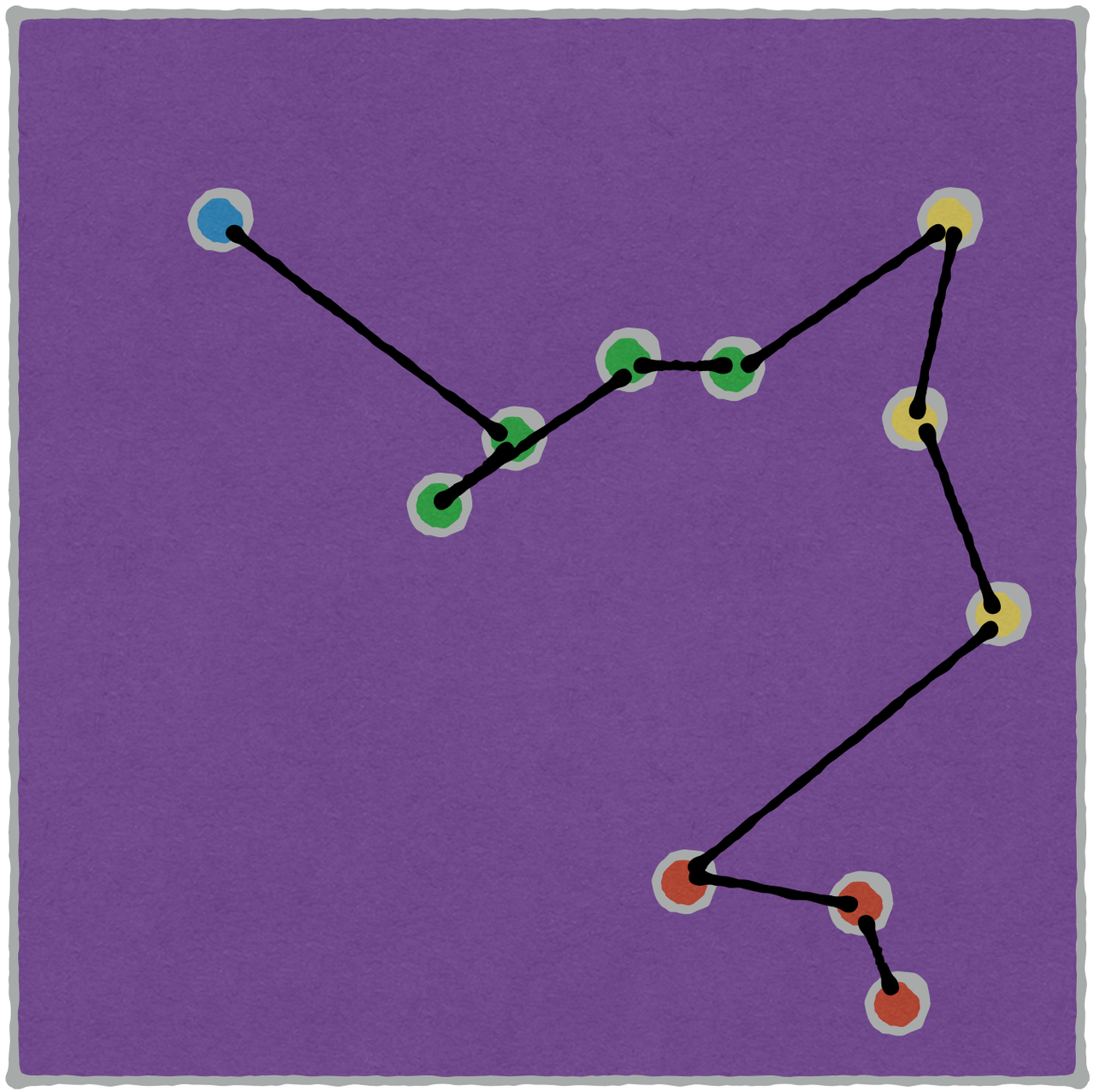
✓ Simple

x Unbalanced (different number of force summations)

x Bad locality

vision - what to treat as multipole expansion and what to treat as direct summation

Domain decomposition



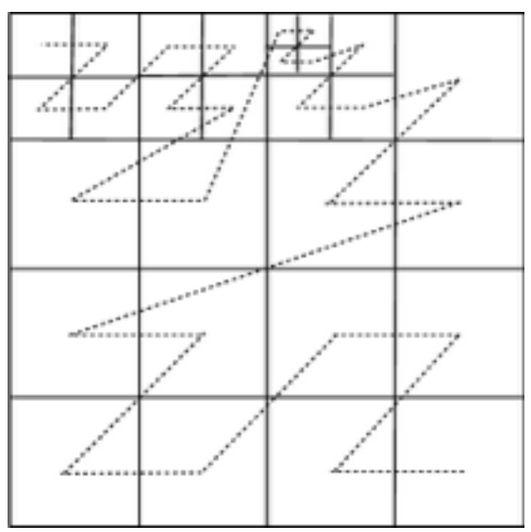
- ✓ Good locality with space-filling curves
- ✓ Load balancing
- x Requires sorting

Space-filling curves

- SFC show better locality
- Split particles along a 1D curve
- Morton indexing (Z-ordering):
  computation by interleaving bits of coordinates

  (i) Compute particle index

  (ii) Sort particles
- Other options: Hilbert, Peano



Z-ordering SFC

picture from http://flash.uchicago.edu/~jbgallag/2012/flash4_ug/node14.html

Load balancing

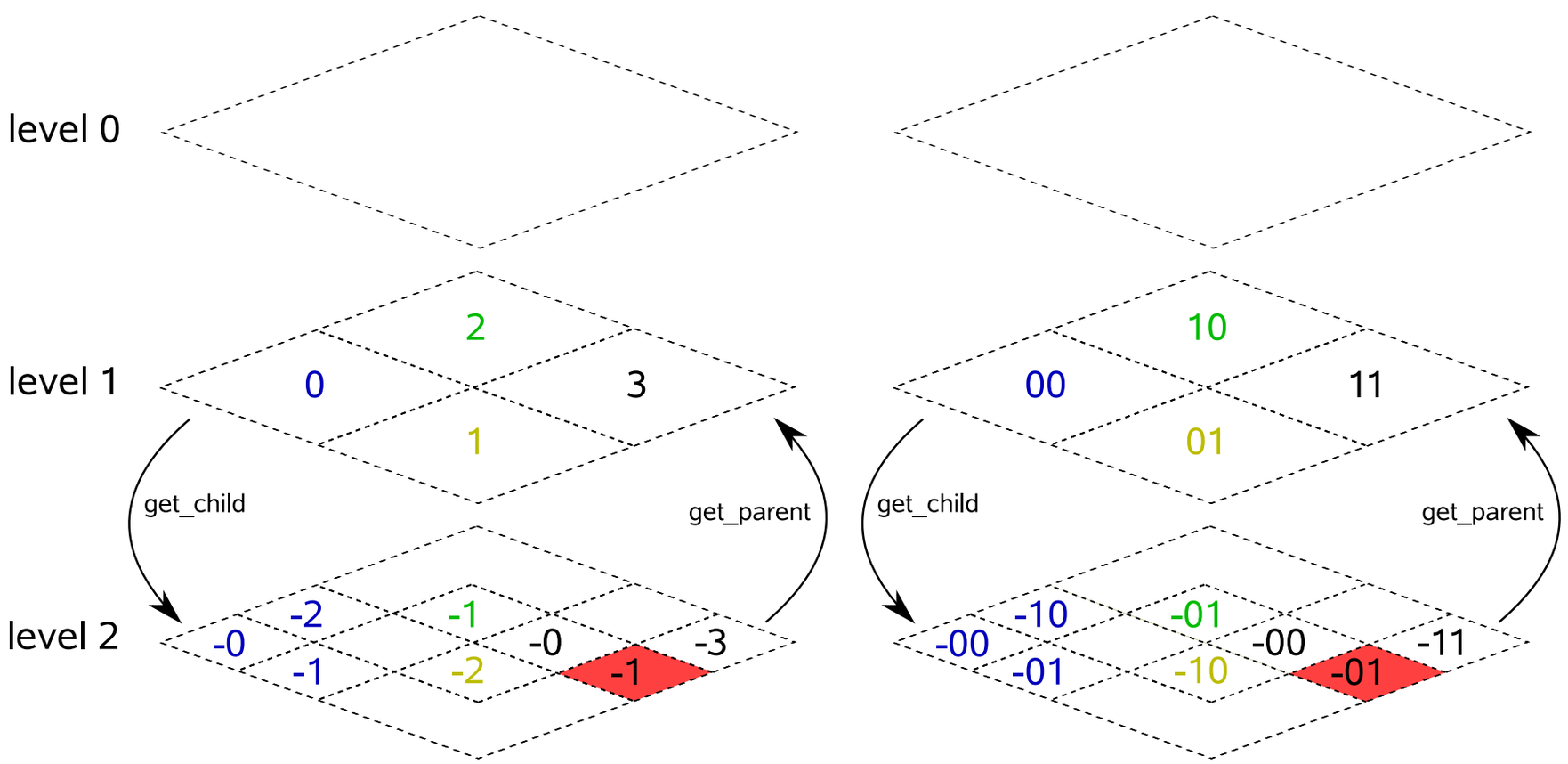
- Use information of previous time steps
  - #interactions computed for a given particle?
  - Rationale: the number doesn’t change much

Building the quadtree

- Each parallel unit builds its part of the tree
- Needed: “ghosts” at extremes of 1D curve
- Z-ordering encodes the quadtree
  - position in the tree must be extracted
  - access is O(1) instead of O(logN)

Building the quadtree

- Parents can be obtained by right bit shifts
- Children are obtained by left bit shifts



picture from Gibbon et al, Parallel Tree Codes

Tree traversal

- Each parallel element creates an interaction list

- On distributed memory architectures:
  interaction list reduces communication messages

Force summation

- Given interaction lists, update particles
- Separate step:
  - keeps physics separated
  - some load balancing can be performed