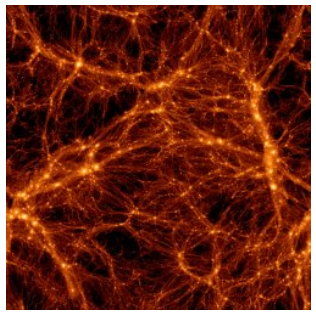
Parallel Computing with Adaptive Mesh Refinement

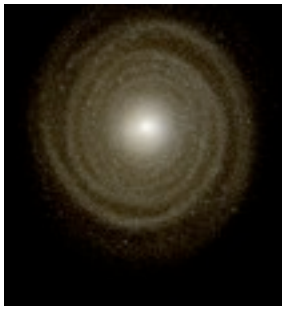
Cosmological simulations with RAMSES

Romain Teyssier, Markus Wetzstein

University of Zurich

HP2C High Performance and High Productivity Computing

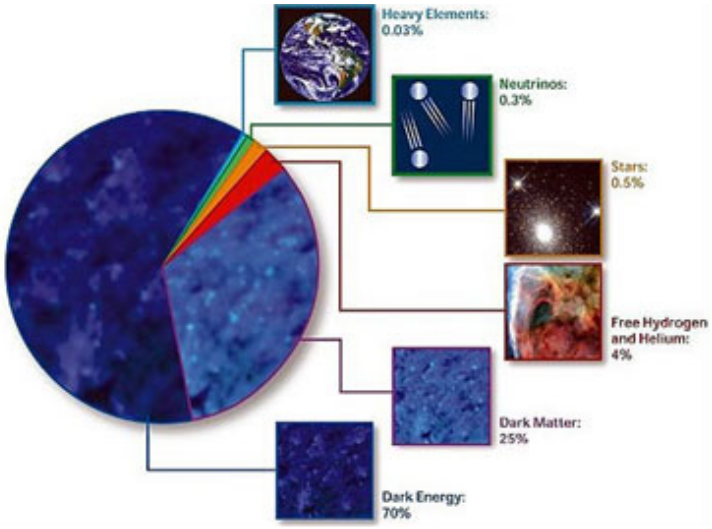


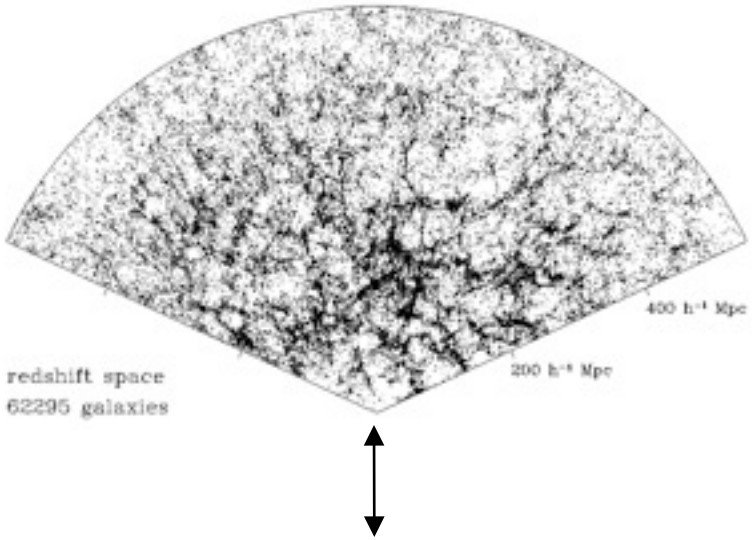


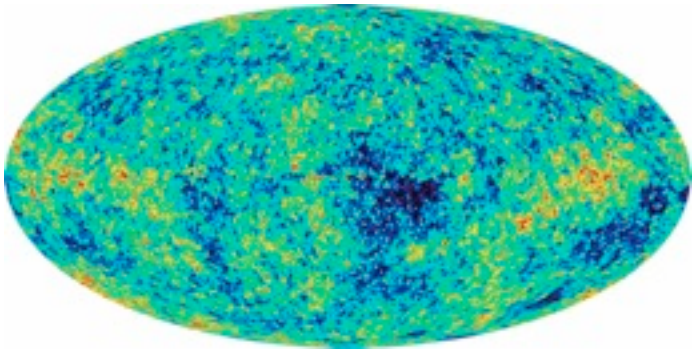
Outline

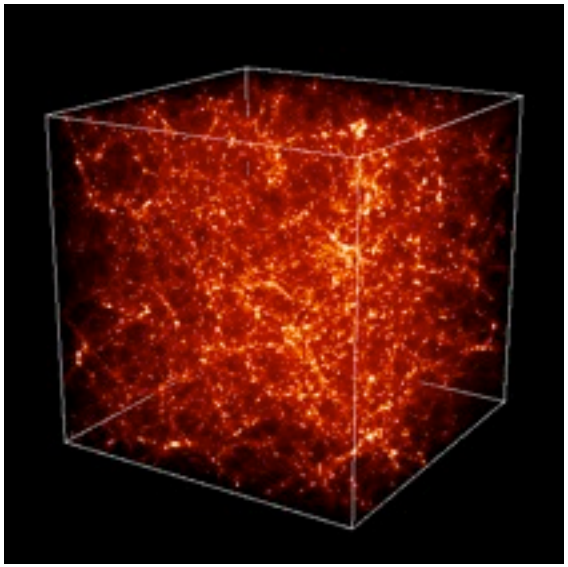
- Cosmological simulations
- Adaptive Mesh Refinement techniques
- Graded octree data structure
- Hydrodynamics with AMR
- Adaptive time stepping
- Parallel computing with RAMSES
- Cosmological simulations with RAMSES

Cosmological simulations



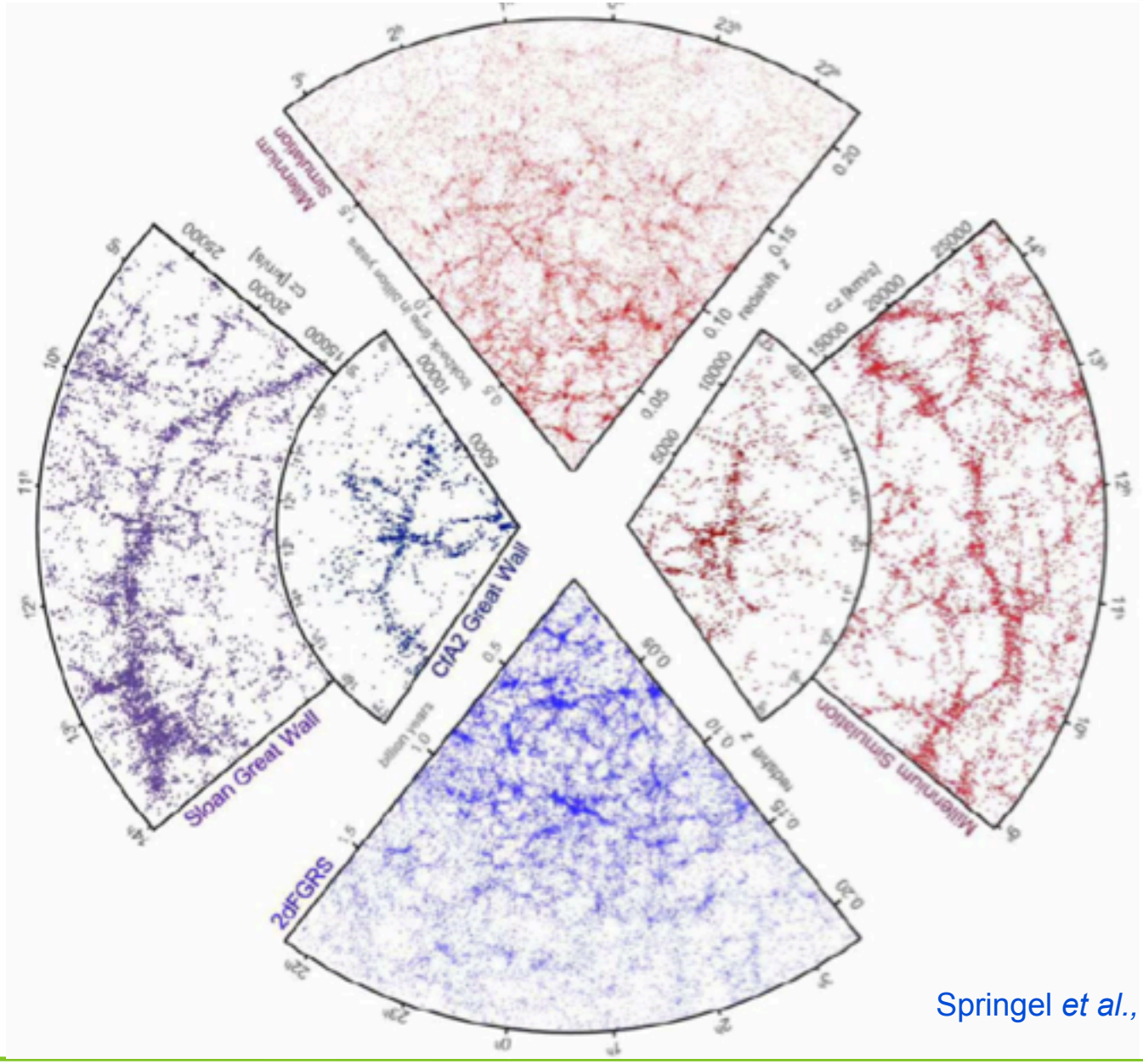






From Gaussian random fields to galaxies: non-linear dynamics of gravitational instability with N-body and hydrodynamics codes.

Cosmological simulations



Springel *et al.*, Nature, 2006

Godunov scheme for hyperbolic systems

The system of conservation laws

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} = 0$$

is discretized using the following integral form:

$$\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} + \frac{\mathbf{F}_{i+1/2}^{n+1/2} - \mathbf{F}_{i-1/2}^{n+1/2}}{\Delta x} = 0$$

The time average flux function is computed using the self-similar solution of the inter-cell Riemann problem:

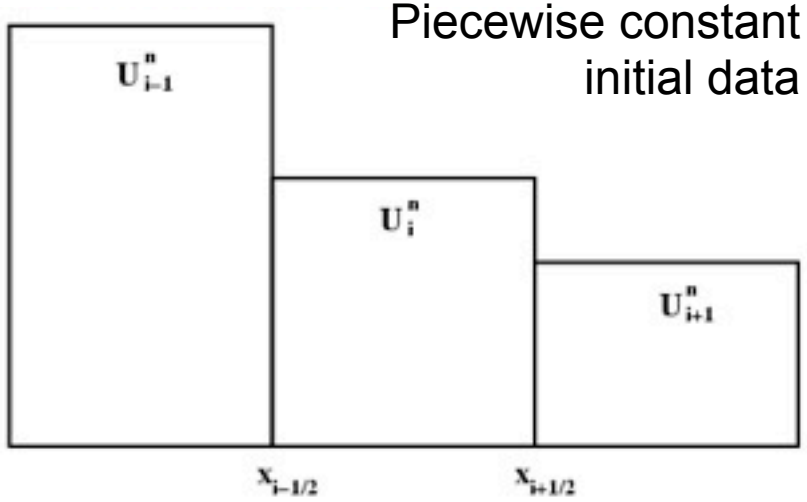
$$\mathbf{U}_{i+1/2}^*(x/t) = \mathcal{RP}\left[\mathbf{U}_i^n, \mathbf{U}_{i+1}^n\right]$$

$$\mathbf{F}_{i+1/2}^{n+1/2} = \mathbf{F}(\mathbf{U}_{i+1/2}^*(0))$$

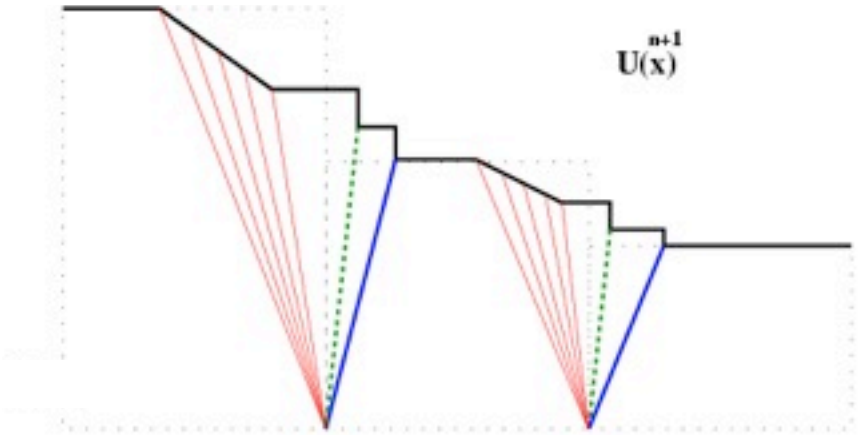
This defines the Godunov flux:

$$\mathbf{F}_{i+1/2}^{n+1/2} = \mathbf{F}^*(\mathbf{U}_i^n, \mathbf{U}_{i+1}^n)$$

Piecewise constant initial data



- **Godunov, S. K.** (1959), A Difference Scheme for Numerical Solution of Discontinuos Solution of Hydrodynamic Equations, *Math. Sbornik, 47, 271-306*, translated US Joint Publ. Res. Service, JPRS 7226, 1969.



Advection: 1 wave, Euler: 3 waves, MHD: 7 waves

Beyond second order Godunov schemes ?

Smooth regions of the flow

More efficient to go to higher order.

Spectral methods can show *exponential convergence*.

More flexible approaches: use *ultra-high-order* shock-capturing schemes: 4th order scheme, ENO, WENO, discontinuous Galerkin and discontinuous element methods

Discontinuity in the flow

More efficient to refine the mesh, since higher order schemes drop to first order.

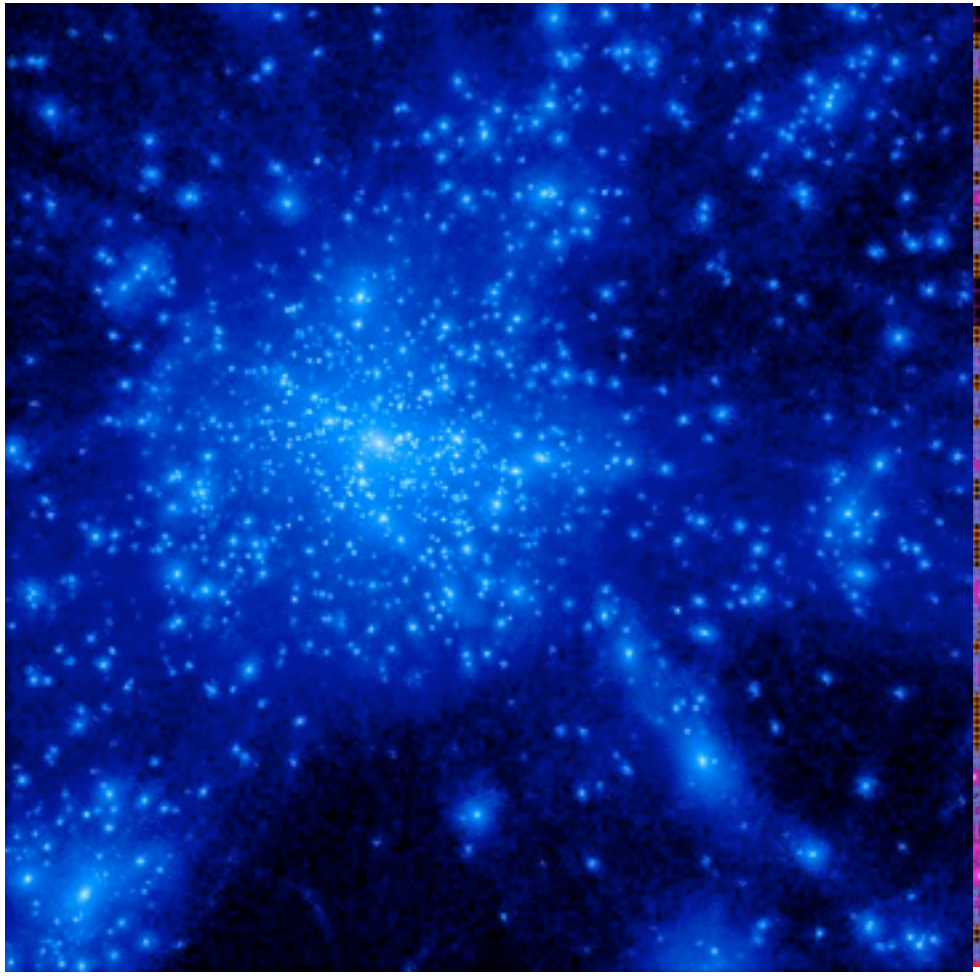
Adaptive Mesh Refinement is the most appealing approach.

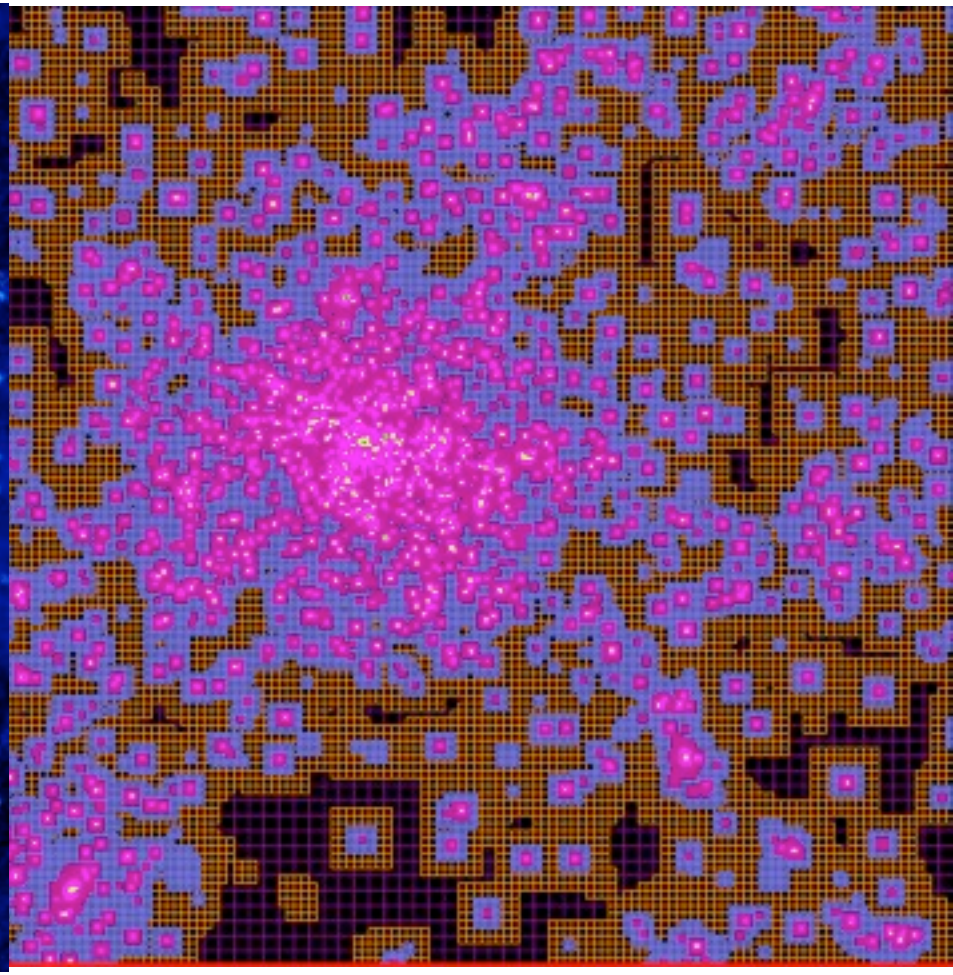
What about the future ?

Combine the 2 approaches.

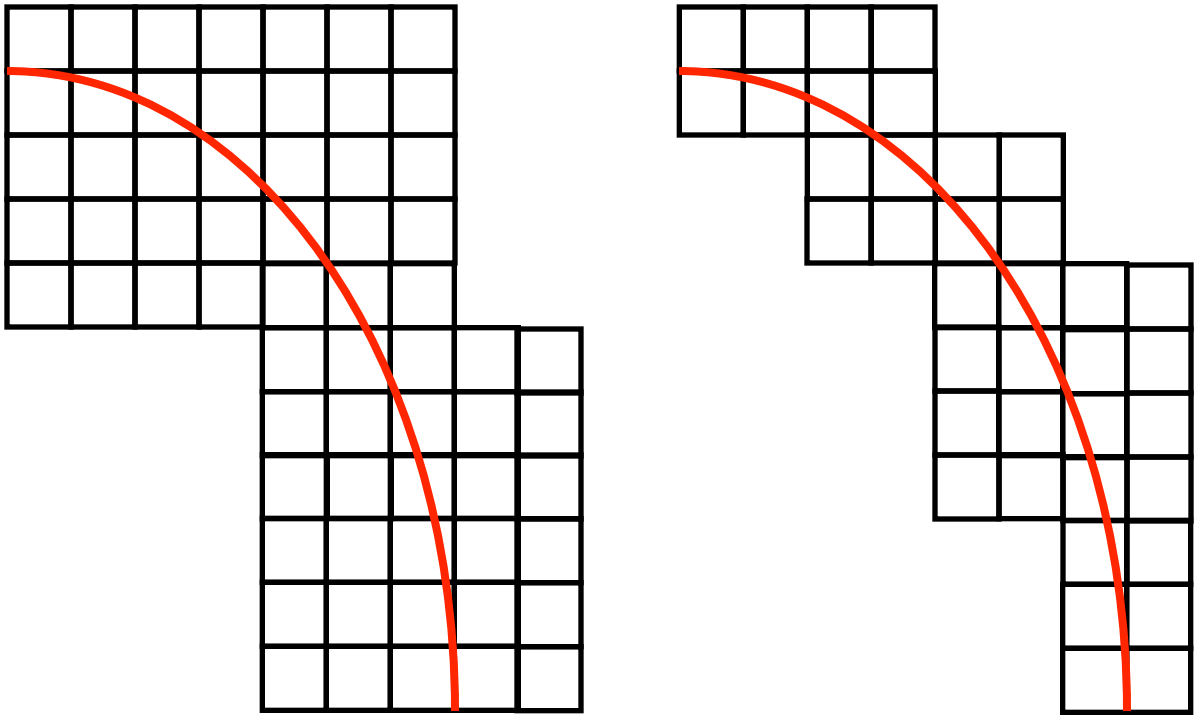
Usually referred to as "*h-p adaptivity*".

Adaptive Mesh Refinement





Patch-based versus tree-based



AMR codes in astrophysics

ENZO: Greg Bryan, Michael Norman, Tom Abel

ART: Andrey Kravtsov, Anatoly Klypin

RAMSES: Romain Teyssier

NIRVANA: Udo Ziegler

AMRVAC: Gabor Thot and Rony Keppens

FLASH: The Flash group (PARAMESH lib)

ORION: Richard Klein, Chris McKee, Phil Colella

PLUTO: Andrea Mignone (CHOMBO lib, Phil Colella)

CHARM: Francesco Miniati (CHOMBO lib, Phil Colella)

ASTROBear: Adam Frank

Graded Octree structure

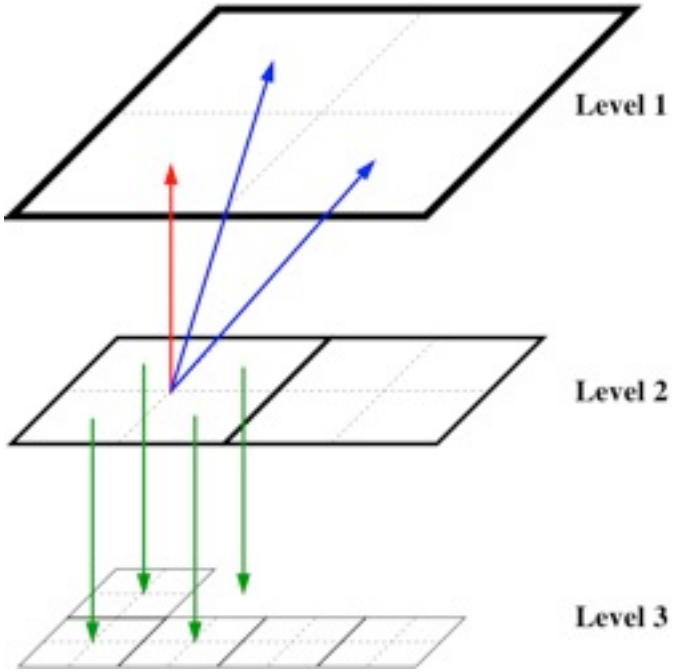
Fully Threaded Tree (Khokhlov 98).

Cartesian mesh refined on a *cell by cell basis.*

`octs`: small grid of 8 cells

Pointers (arrays of index)

- 1 parent cell
- 6 neighboring parent cells
- 8 children octs
- 2 linked list indices



Cell-centered variables are updated level by level using linked lists.

Cost = 2 integer per cell.

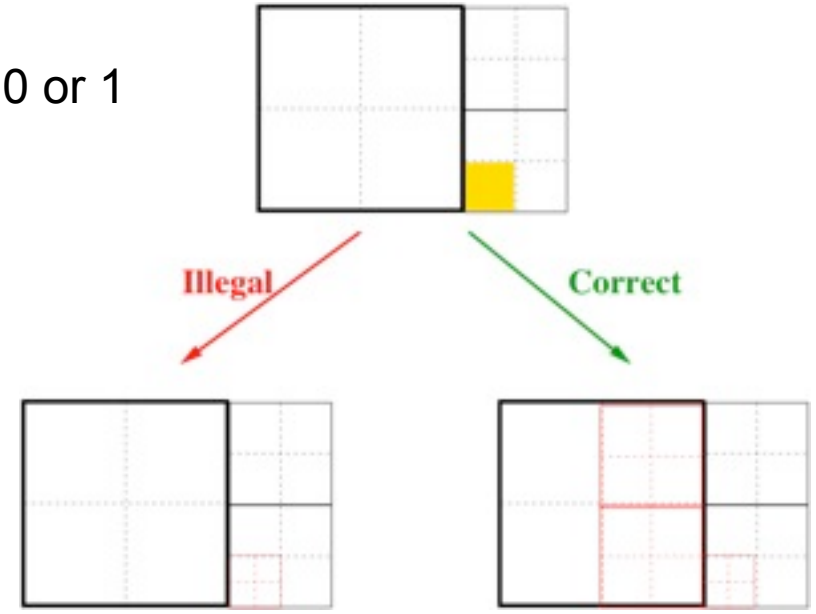
Optimize mesh adaptation to complex flow geometries, but CPU overhead compared to unigrid can be as large as 50%.

2 type of cell: - “leaf” or active cell

- “split” or inactive cell

Refinement rules for graded octree

Compute the refinement map: flag = 0 or 1



Step 1: mesh consistency

if a split cell contains at least one split or marked cell, then mark the cell with flag = 1 and mark its 26 neighbors

Step 2: physical criteria

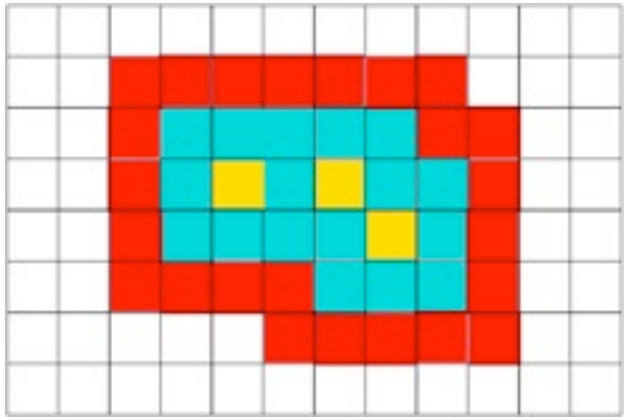
quasi-Lagrangian evolution, Jeans mass

geometrical constraints (zoom)

Truncation errors, density gradients…

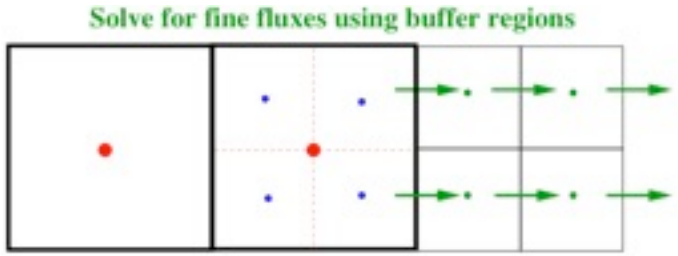
Step 3: mesh smoothing

apply a dilatation operator (mathematical morphology) to regions marked for refinement → convex hull



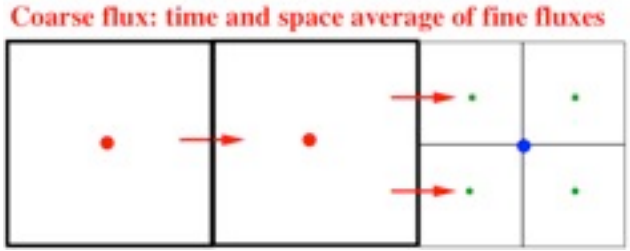
Godunov schemes and AMR

Berger & Oliger (84), Berger & Collela (89)



Prolongation (interpolation) to finer levels

- fill buffer cells (boundary conditions)
- create new cells (refinements)



Restriction (averaging) to coarser levels

- destroy old cells (de-refinements)

Flux correction at level boundary

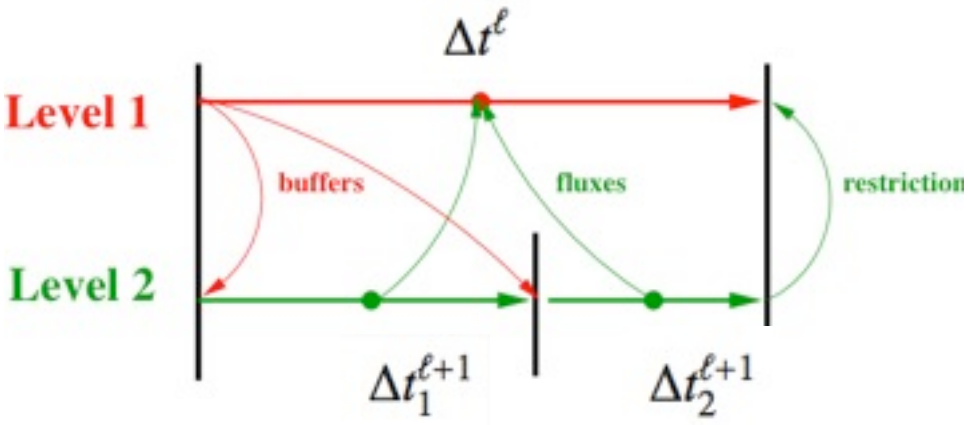
$$(\mathbf{F}^{n+1/2,\ell}_{i+1/2,j}) = \frac{(\mathbf{F}^{n+1/2,\ell+1}_{i+1/2,j-1/4}) + (\mathbf{F}^{n+1/2,\ell+1}_{i+1/2,j+1/4})}{2}$$

Careful choice of interpolation variables (conservative or not ?)

Several interpolation strategies (with $R^T P = I$) :

- straight injection
- tri-linear, tri-parabolic reconstruction

Adaptive Time Stepping

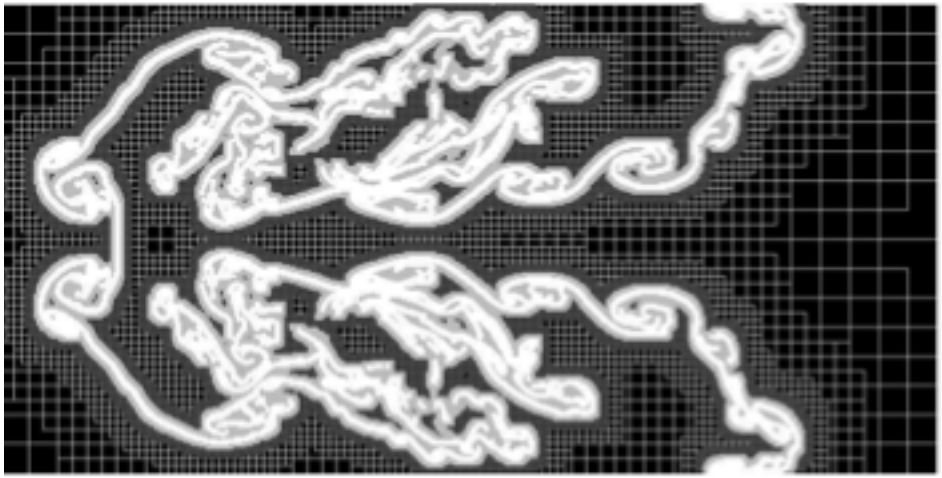


Time integration: single time step or recursive sub-cycling

- froze coarse level during fine level solves (one order of accuracy down !)
- average fluxes in time at coarse fine boundaries

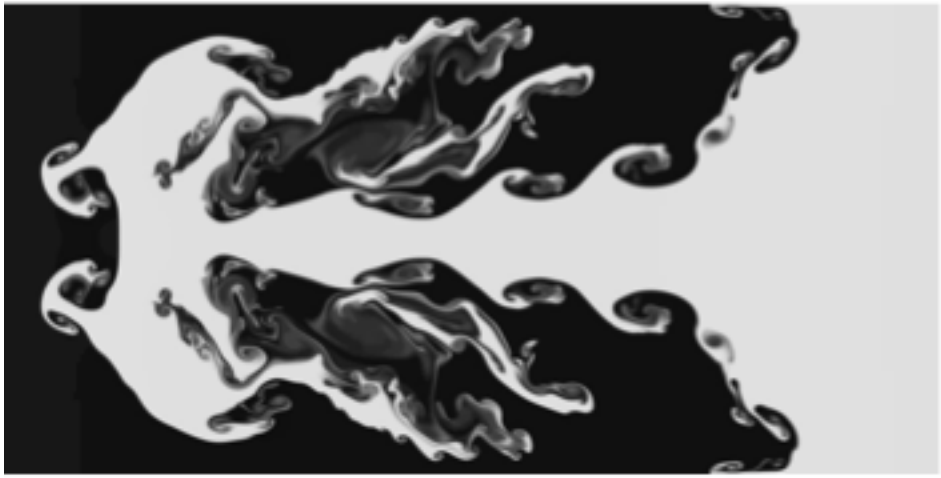
$$(\mathbf{F}_{i+1/2,j}^{n+1/2,\ell}) = \frac{1}{\Delta t_1^{\ell+1} + \Delta t_2^{\ell+1}} \left(\Delta t_1^{\ell+1} \frac{(\mathbf{F}_{i+1/2,j-1/4}^{n+1/4,\ell+1}) + (\mathbf{F}_{i+1/2,j+1/4}^{n+1/4,\ell+1})}{2} + \Delta t_2^{\ell+1} \frac{(\mathbf{F}_{i+1/2,j-1/4}^{n+3/4,\ell+1}) + (\mathbf{F}_{i+1/2,j+1/4}^{n+3/4,\ell+1})}{2} \right)$$

Complex geometry with AMR



Maximum numerical dissipation occurs at the 2 fluids interface.

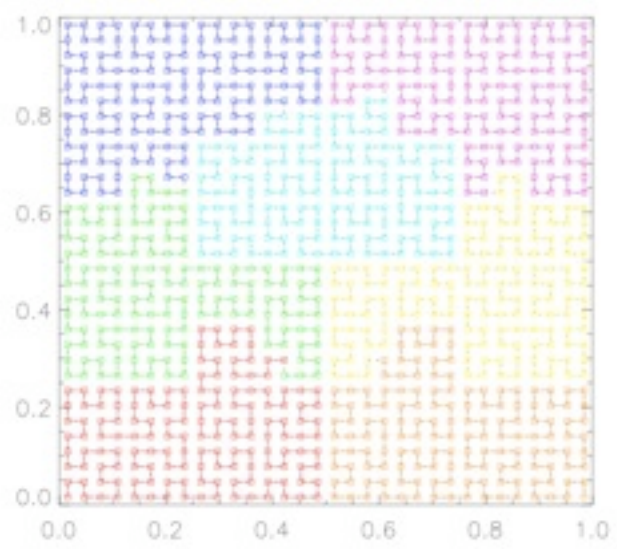
The optimal refinement strategy is based on density gradients.



The number of required cells is directly related to the *fractal exponent* n of the 2D surface.

$$N_{cell} \propto (\Delta x)^{-n}$$

Domain decomposition for parallel computing



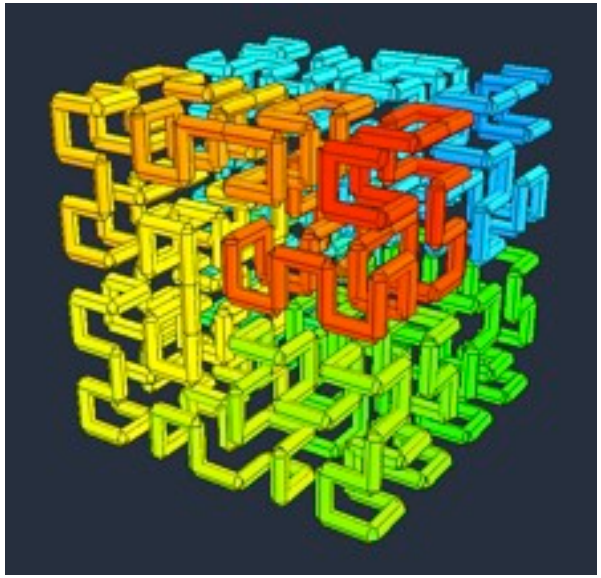
Parallel computing using the MPI library with a domain decomposition based on the *Peano-Hilbert curve*.

Algorithm inspired by gravity solvers (tree codes).
Use locally essential tree.

Tested and operational up to 76'000 core.
Scaling depends on problem size and complexity.

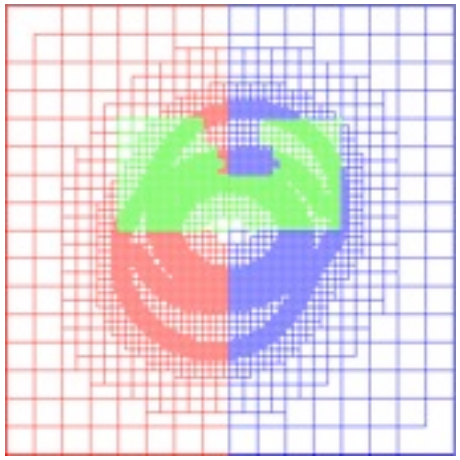
Salmon, J.K. and Warren, M.S., "Parallel out-of-core methods for N-body simulation", In Eighth SIAM Conference on Parallel Processing for Scientific Computing, SIAM,1997.

Peter MacNeice, Kevin M. Olsonb, Clark Mobarryc, Rosalinda de Fainchteind and Charles Packer, « PARAMESH: A parallel adaptive mesh refinement community toolkit, », 2000, Computer Physics Communications, 126, 3.



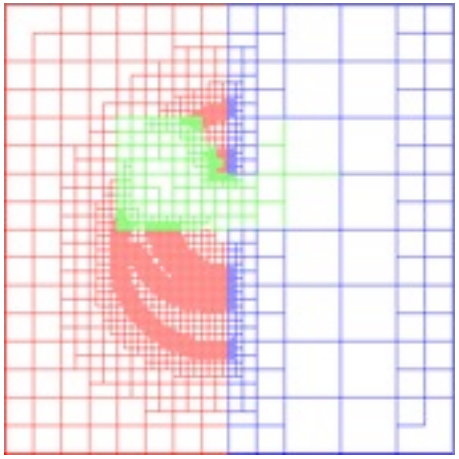
Locally essential trees

Salmon 90
Warren 92
Dubinski 96

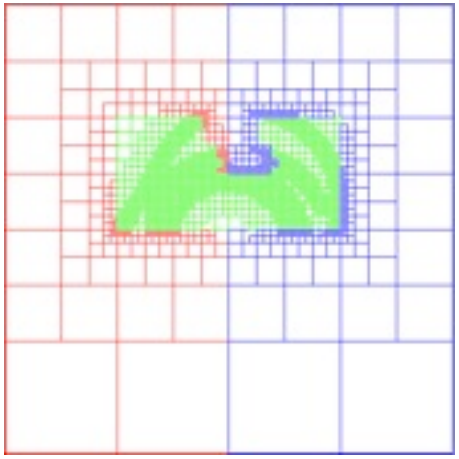


Domain decomposition over 3 processors

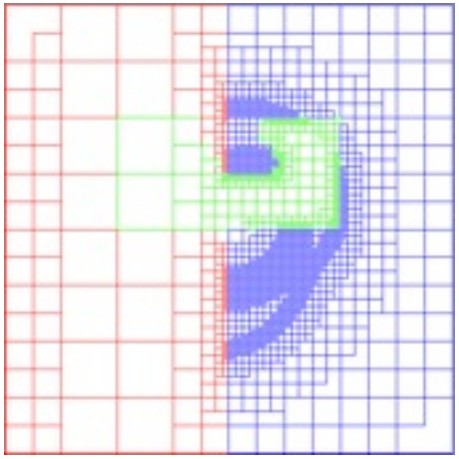
Each processor octree is surrounded by ghost cells (local copy of distant processor octrees) so that the resulting local octree contains all the necessary information.



Locally essential tree in processor #1



Locally essential tree in processor #2



Locally essential tree in processor #3

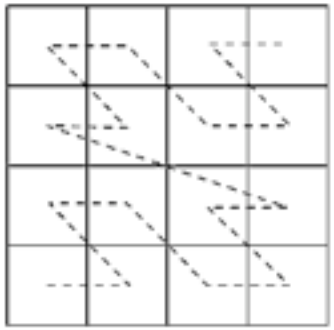
Dynamic partitioning in RAMSES

Several cell ordering methods:

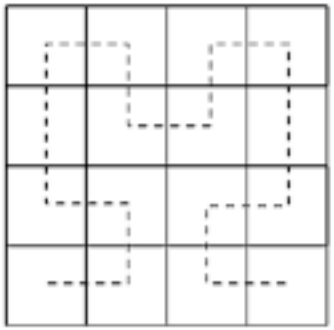
1- column, row or plane major

2- Hilbert or Morton

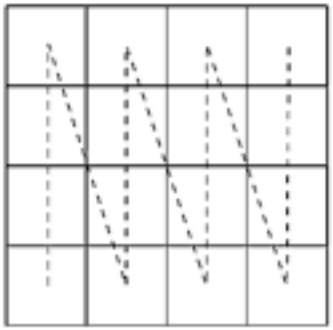
3- User defined (angular, column+Hilbert…)



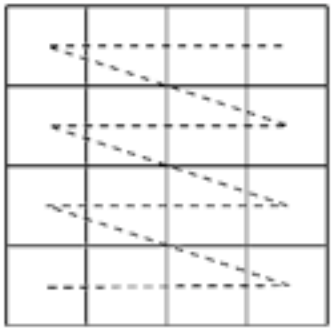
Morton



Hilbert



Column major

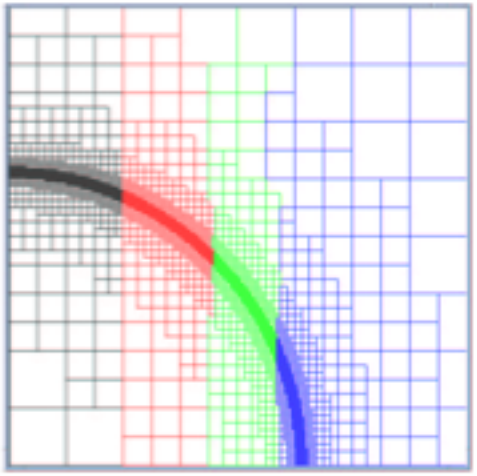


Row major

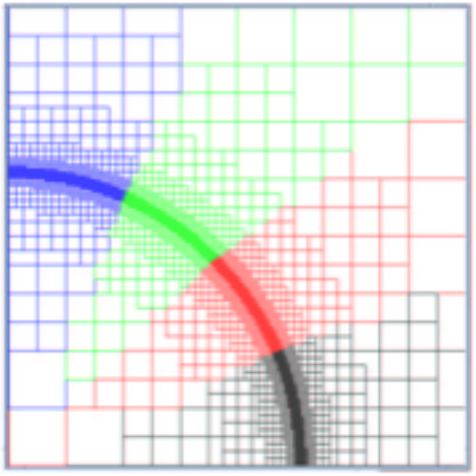
`nremap=10`

Dynamic partioning is performed every N steps by sorting each cell along chosen ordering and redistributing the mesh across processors. Usually, a good choice is N=10 (10% overhead).

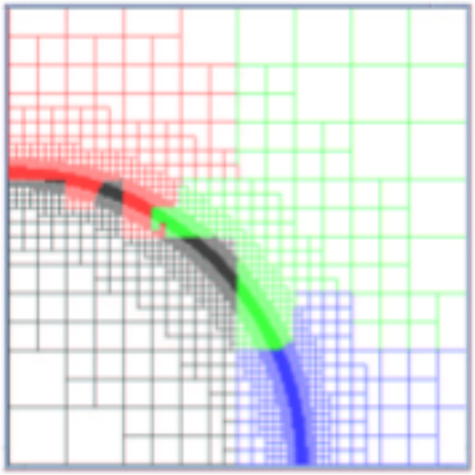
Is there an optimal load balancing strategy ?



Column major

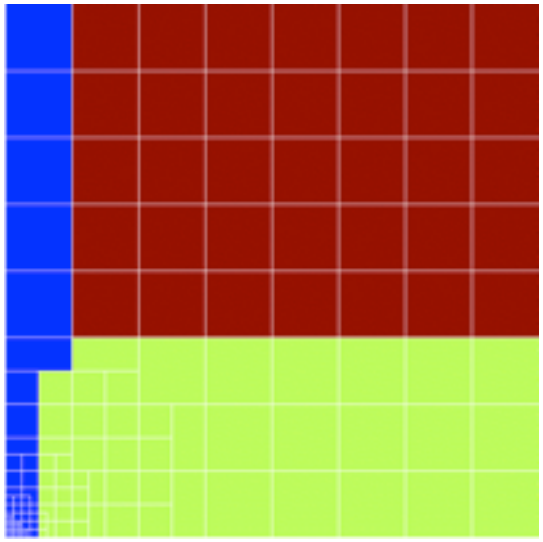


Angular

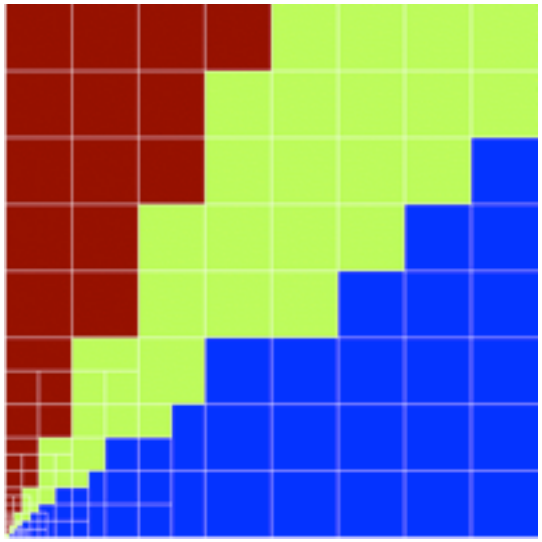


Hilbert

Is there an optimal load balancing strategy ?

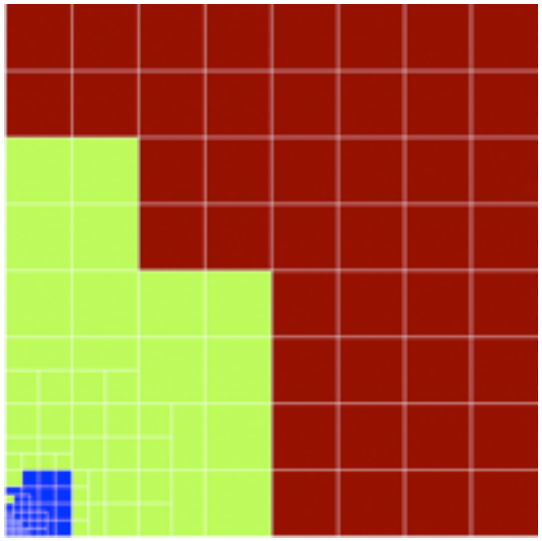


Recursive bisection



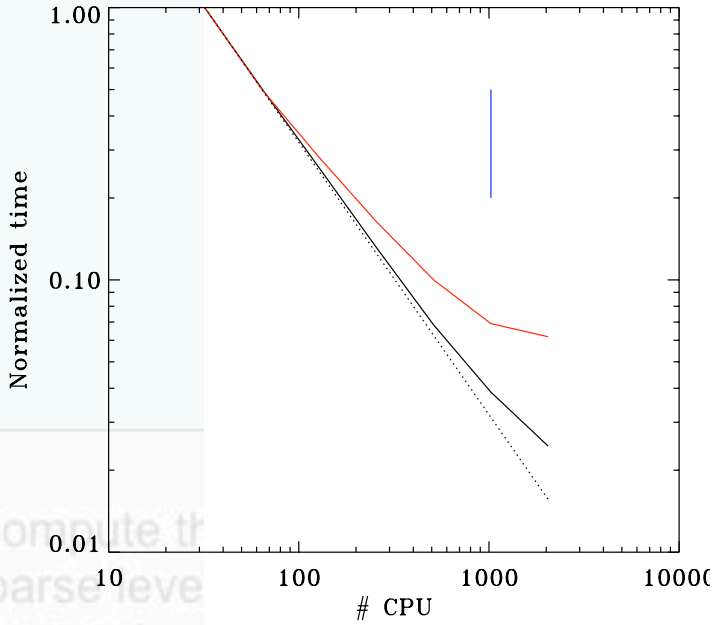
Angular ordering

Hilbert ordering



Load balancing issues in RAMSES

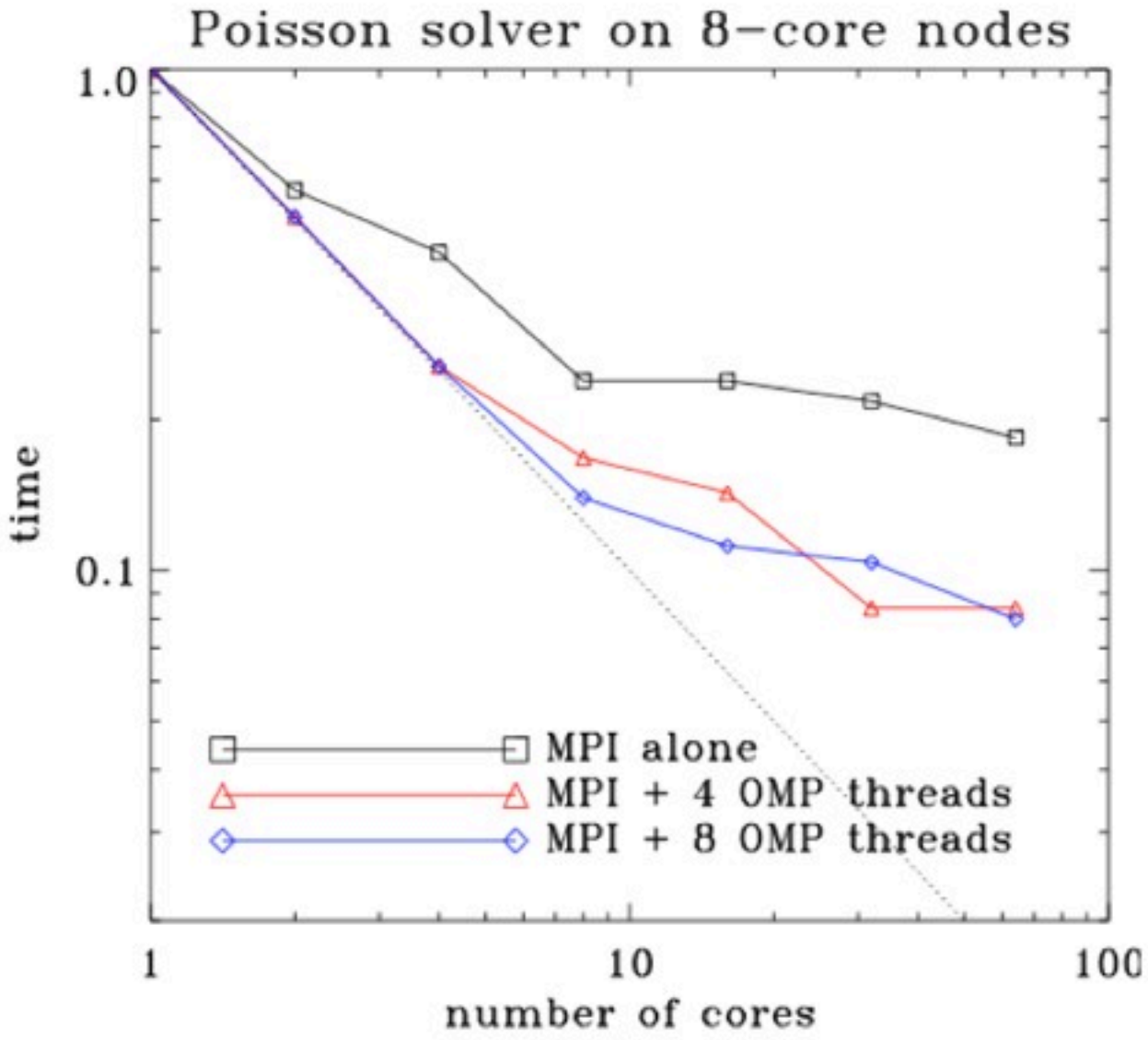
```
Mesh structure
Level  1 has          1 grids (       0,       1,       0,)
Level  2 has          8 grids (       0,       2,       0,)
Level  3 has         64 grids (       0,       7,       0,)
Level  4 has        512 grids (       0,      49,       0,)
Level  5 has       4096 grids (       0,     384,       4,)
Level  6 has      32768 grids (       0,    3075,      32,)
Level  7 has     262144 grids (       0,   24607,     256,)
Level  8 has      21147 grids (       0,    2309,      20,)
Level  9 has      27690 grids (       0,    2985,      27,)
Level 10 has      42860 grids (       0,    3948,      41,)
Level 11 has      85768 grids (       0,    7676,      83,)
Level 12 has     213596 grids (       0,   11561,     208,)
Level 13 has     672299 grids (       0,   12783,     656,)
Level 14 has    2388883 grids (       0,   21258,    2332,)
Level 15 has    2795115 grids (       0,   12304,    2729,)
Level 16 has    2850931 grids (       0,    8315,    2784,)
Level 17 has    1662701 grids (       0,    5140,    1623,)
Level 18 has     674656 grids (       0,    2859,     658,)
Level 19 has     253539 grids (       0,    1524,     247,)
Main step=  6735 econs=-1.86E-03 epot=-3.41E-05 ekin= 2.27E-05 eint= 1.57E-08
Fine step= 72805 t=-2.65682E+00 dt= 3.196E-05 a= 2.955E-01 mem=13.8% 17.0%
Level=    19 New star=    66 Tot=   1604520 Mass= 6.322E-08 Lost= 0.0%
  ==> Level=   19 Step=    4 Error= 5.302E-05
Fine step= 72806 t=-2.65679E+00 dt= 3.463E-05 a= 2.955E-01 mem=13.8% 17.0%
Level=    19 New star=    71 Tot=   1604591 Mass= 6.322E-08 Lost= 0.0%
  ==> Level=   18 Step=    4 Error= 3.163E-05
  ==> Level=   19 Step=    4 Error= 2.101E-05
Fine step= 72807 t=-2.65675E+00 dt= 3.571E-05 a= 2.955E-01 mem=13.8% 17.0%
Level=    19 New star=    88 Tot=   1604679 Mass= 6.323E-08 Lost= 0.0%
  ==> Level=   19 Step=    4 Error= 2.267E-05
Fine step= 72808 t=-2.65672E+00 dt= 3.845E-05 a= 2.955E-01 mem=13.8% 17.0%
Level=    19 New star=    79 Tot=   1604758 Mass= 6.323E-08 Lost= 0.0%
```



Adaptive time steps and global domain decomp. lead to poor load balance.

For each problem, there is a plateau in the parallel efficiency.

Hybrid OpenMP and MPI approach



Cosmology with RAMSES

Particle-Mesh on AMR grids:

Cloud size equal to the local mesh spacing

Poisson solver on the AMR grid

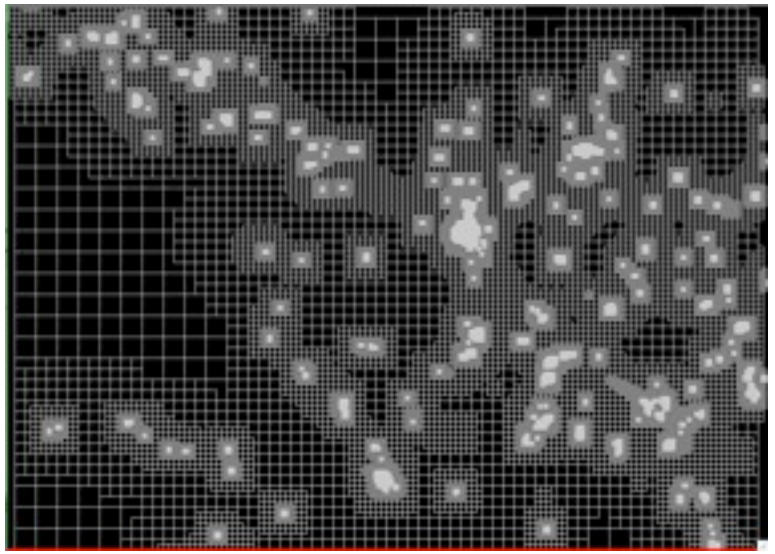
Multigrid or Conjugate Gradient
Interpolation to get Dirichlet boundary conditions (one way interface)

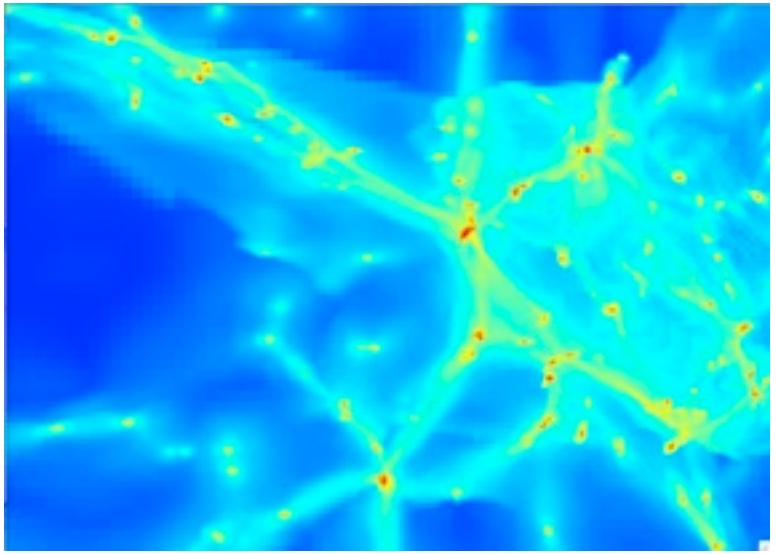
Quasi-Lagrangian mesh evolution:

roughly constant number of particles per cell

$$n = \frac{\rho_{DM}}{m_{DM}} + \frac{\rho_{gas}}{m_{gas}} + \frac{\rho_*}{m_*}$$

Trigger new refinement when n > 10-40 particles. The fractal dimension is close to 1.5 at large scale (filaments) and is less than 1 at small scales (clumps).



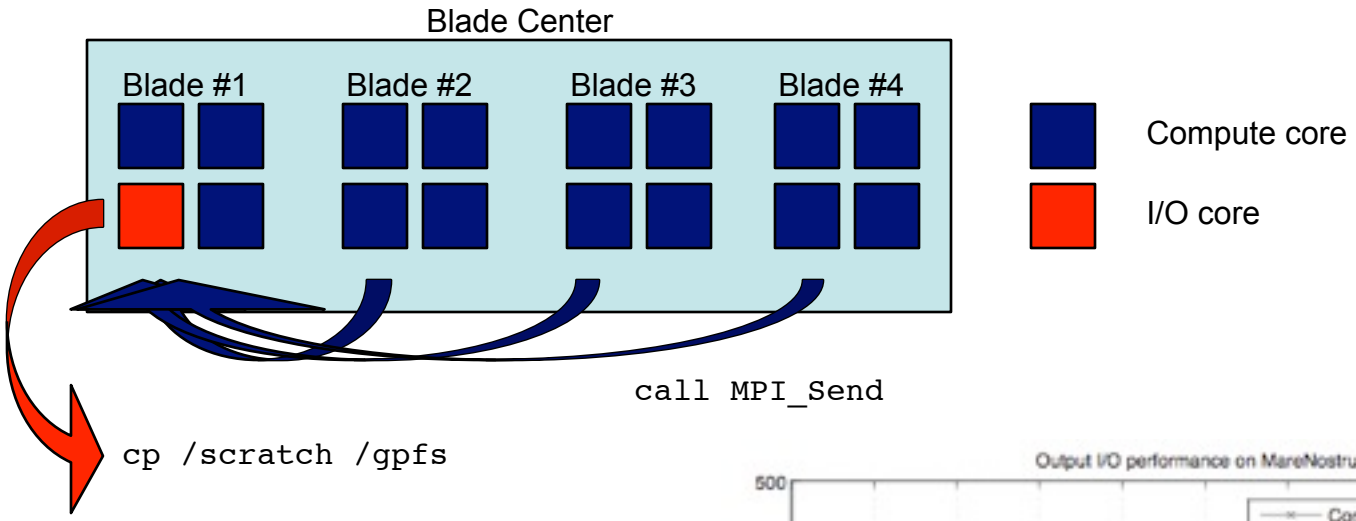


Cosmological simulation with RAMSES

```
&RUN_PARAMS
cosmo=.true.
pic=.true.
poisson=.true.
nrestart=0
nremap=10
nsubcycle=1,2
ncontrol=1
/

&OUTPUT_PARAMS
noutput=10
aout=0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0
/

&INIT_PARAMS
filetype='grafic'
initfile(1)='/scratchdir/grafic_files'
/

&AMR_PARAMS
levelmin=7
levelmax=14
ngridtot=2000000
nparttot=3000000
nexpand=1
/

&REFINE_PARAMS
m_refine=7*8.
/
```

- Pure N-body case.
- `nrestart` controls check-point restarts.
- `nremap` controls load-balancing frequency.
- Choose your output frequency.
- Initial conditions and cosmological model read in from a set of `grafic` files.
- Choose base (coare) grid resolution with `levelmin`.
- Choose ultimate refinement resolution with `levelmax`.
- Critical value `levelmax=21`!
- `ngridmax` (`npartmax`) controls the maximum memory available for AMR cells (particles).
- Choose refinement strategy (here 8 particles per cell triggers a refinement).

Parallel I/O strategy



Output files several times per run (one every 2 hours).

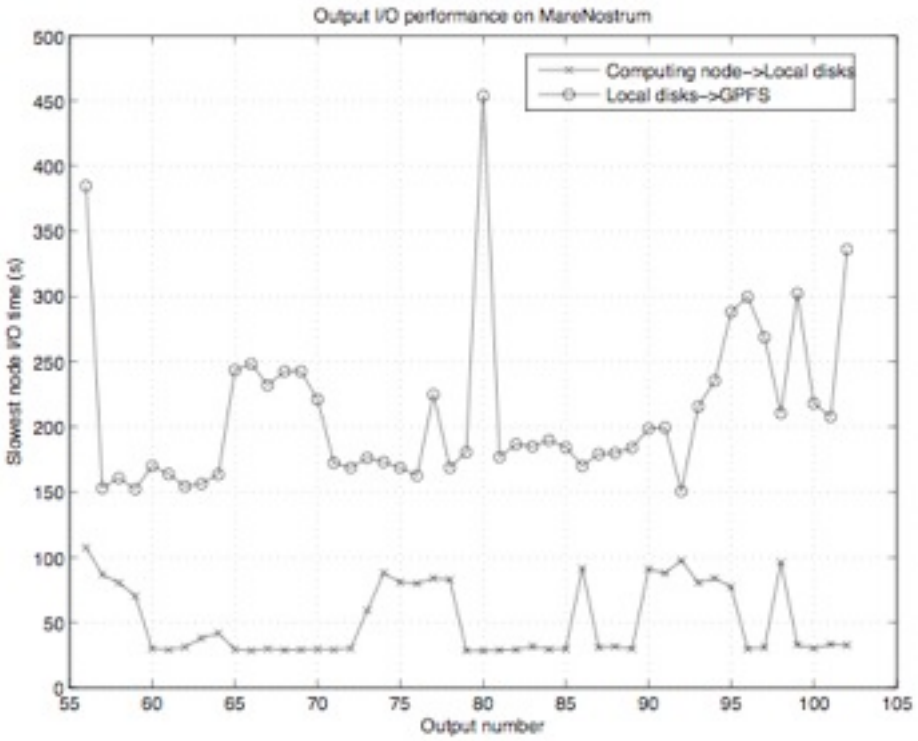
Need to minimize computational downtime due to parallel file system contention (competitive disc access).

Compute group: 2048 compute cores :
`MPI_COMM_WORLD_COMP`

I/O group: 1 I/O core per 32 compute cores :
`MPI_COMM_WORLD_IOGRP` (64 dedicated I/O cores)

Compute core: -Write replaced by `MPI_Send`

I/O core:
- -Wait for compute core to send data
- -Dump data to local scratch disc
- -Resume computation when all data received
- -Transfer data from local scratch to GPFS



Very large N body simulations with RAMSES

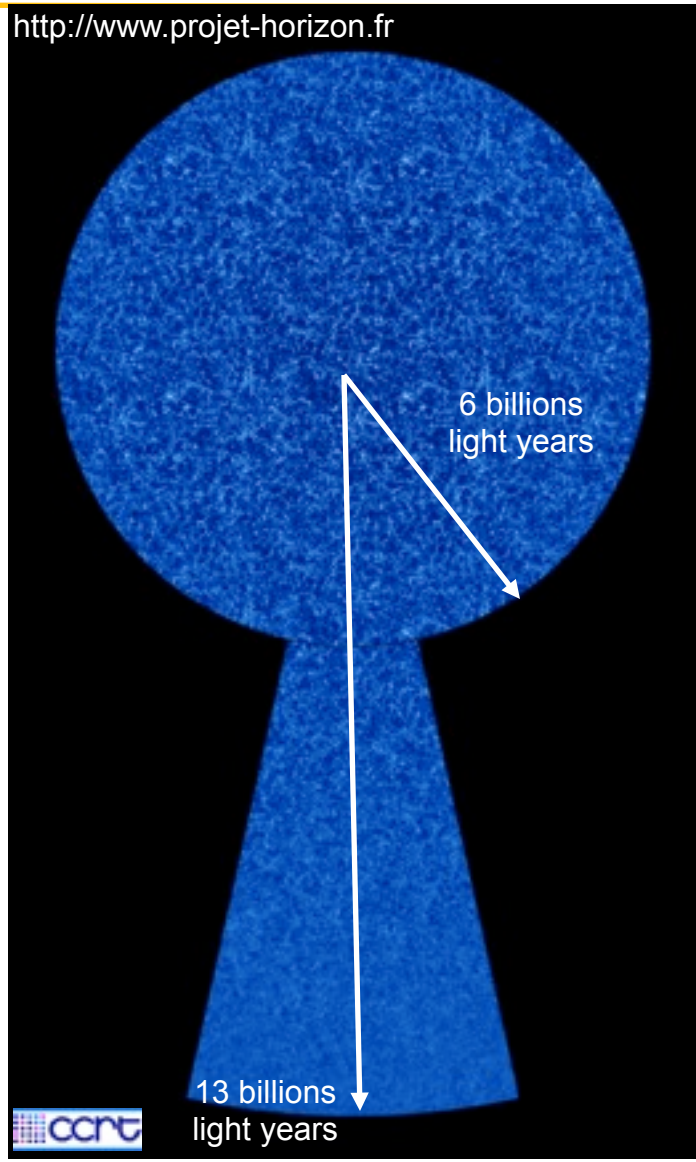
Billion dollar experiments need support from HPC

The Horizon simulation (2 Gpc/h)

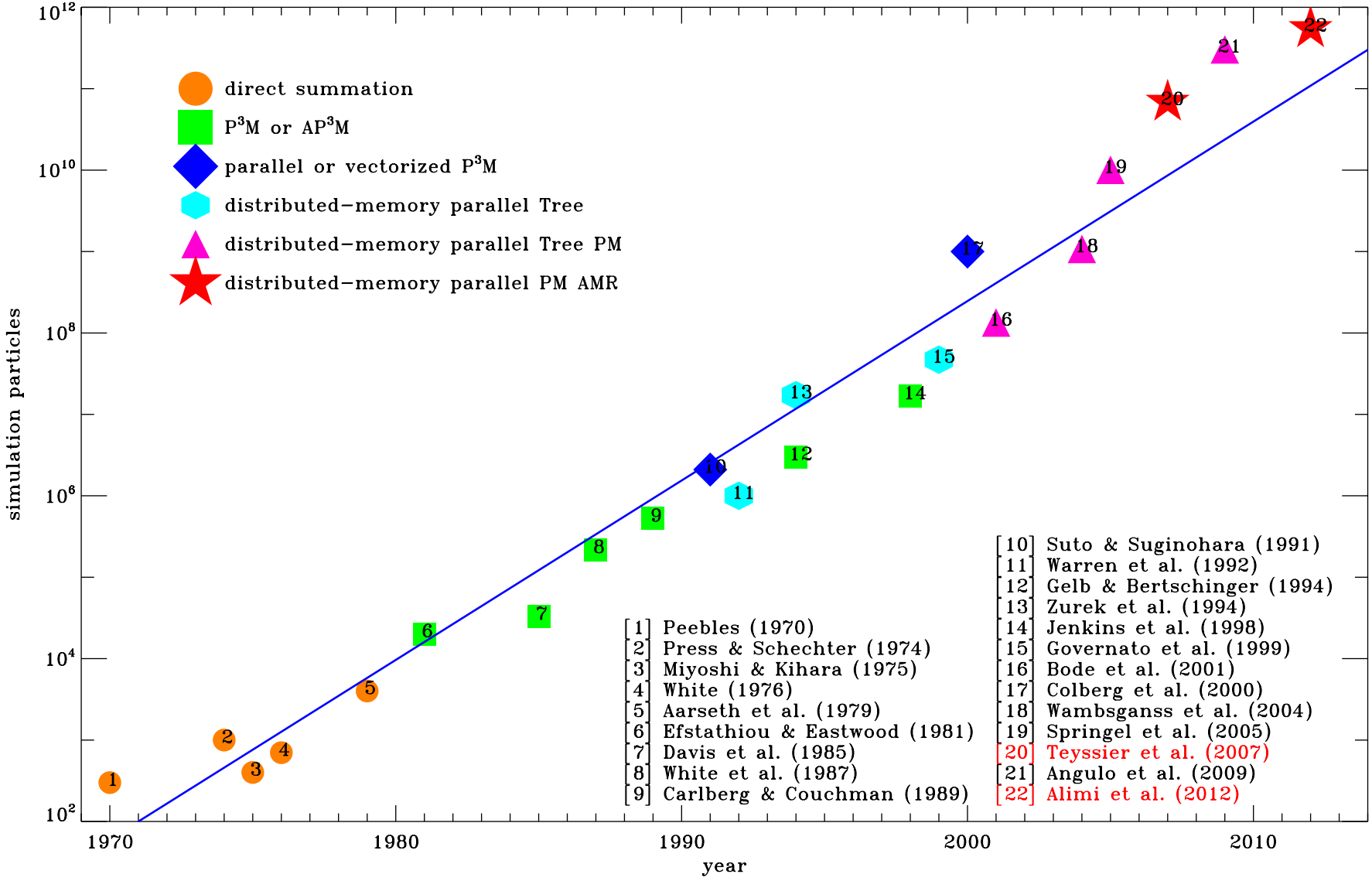
70 billion dark matter particles and 140 billion AMR cells
6144 core
2.4 GB per core
Wall-clock time 2 month
performed in 2007 on the CCRT BULL Novascale 3045

The DEUS simulation (21 Gpc/h)

550 billion dark matter particles and 2 trillion AMR cells
76032 core
4 GB per core
Wall-clock time 1 week
performed in 2012 on the CCRT BULL Bullx S6010



Cosmological N body simulations



Conclusions

- RAMSES is a graded octree AMR
- parallel computing using global domain decomposition
- scaling limited by load balancing
- massive N body run up to 76 000 core