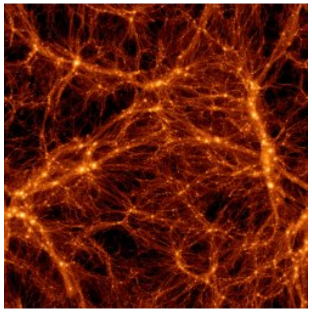
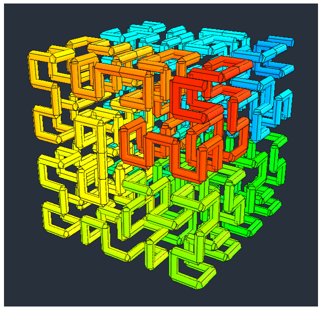
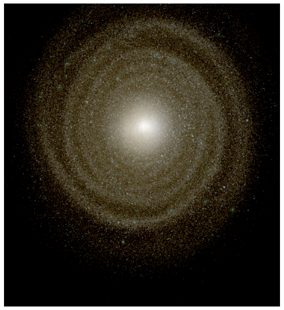
Cosmology on the Petascale

Romain Teyssier
George Lake (PI), Ben Moore, Joachim Stadel

University of Zurich

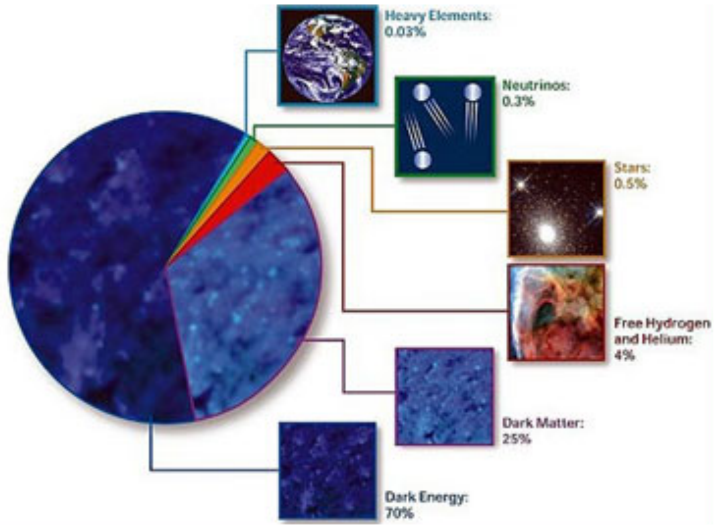
HP2C High Performance and High Productivity Computing

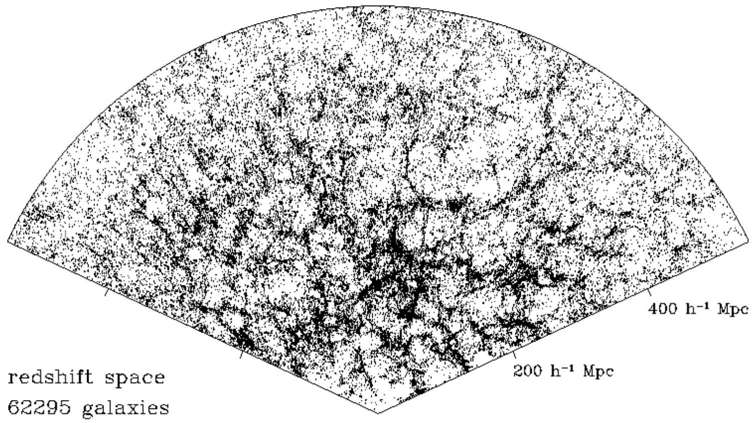




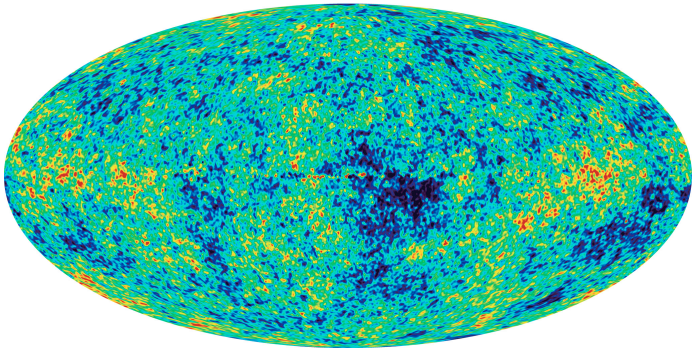
Outline

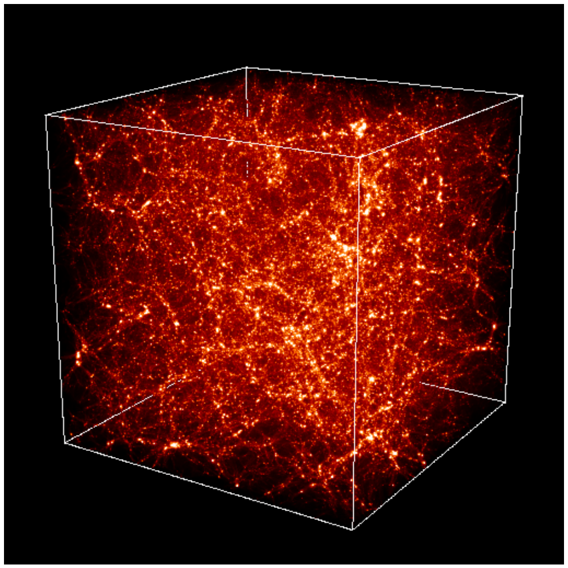
- General context
- Science objectives
- Code development
- Project organisation

Cosmological simulations



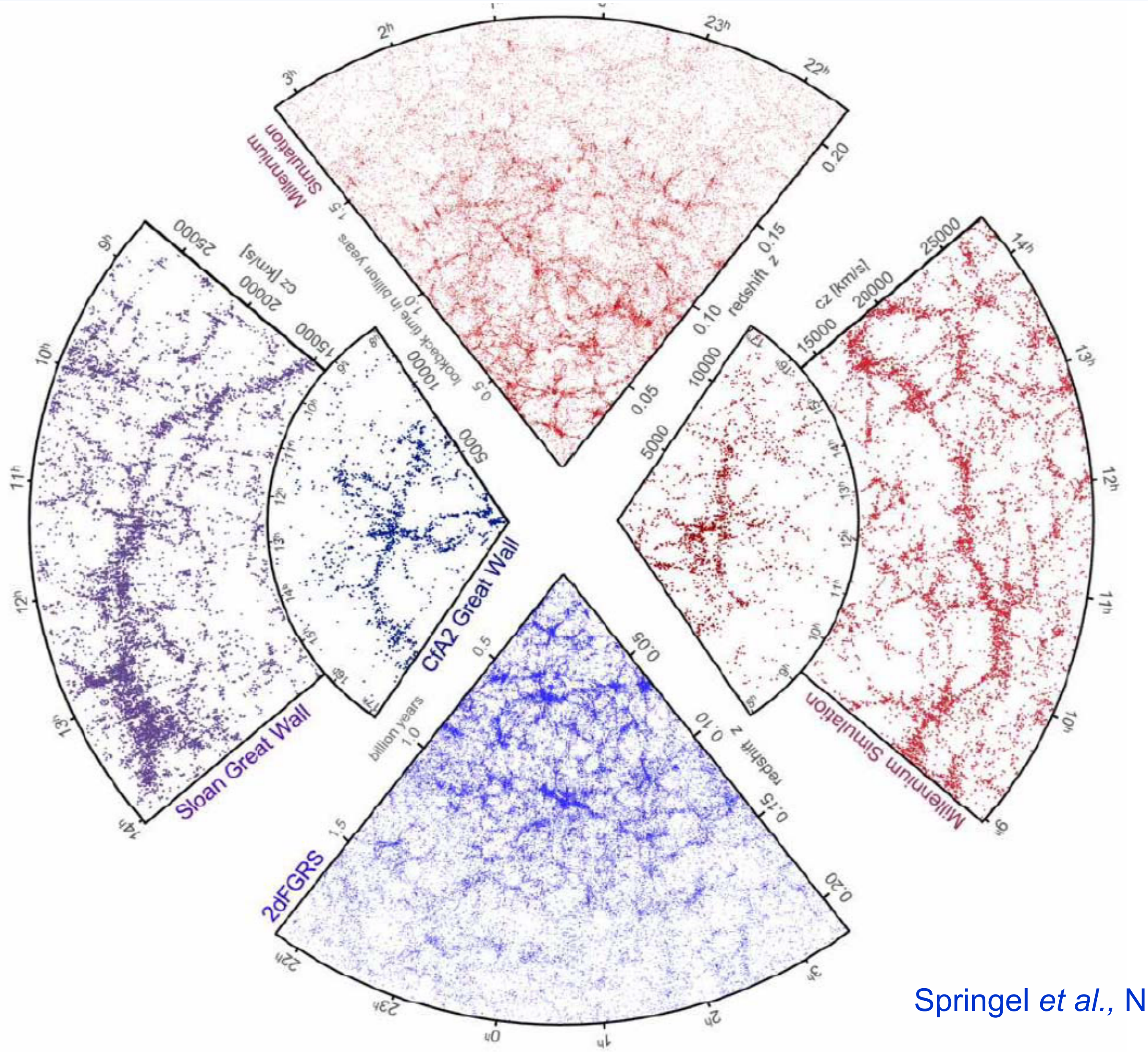




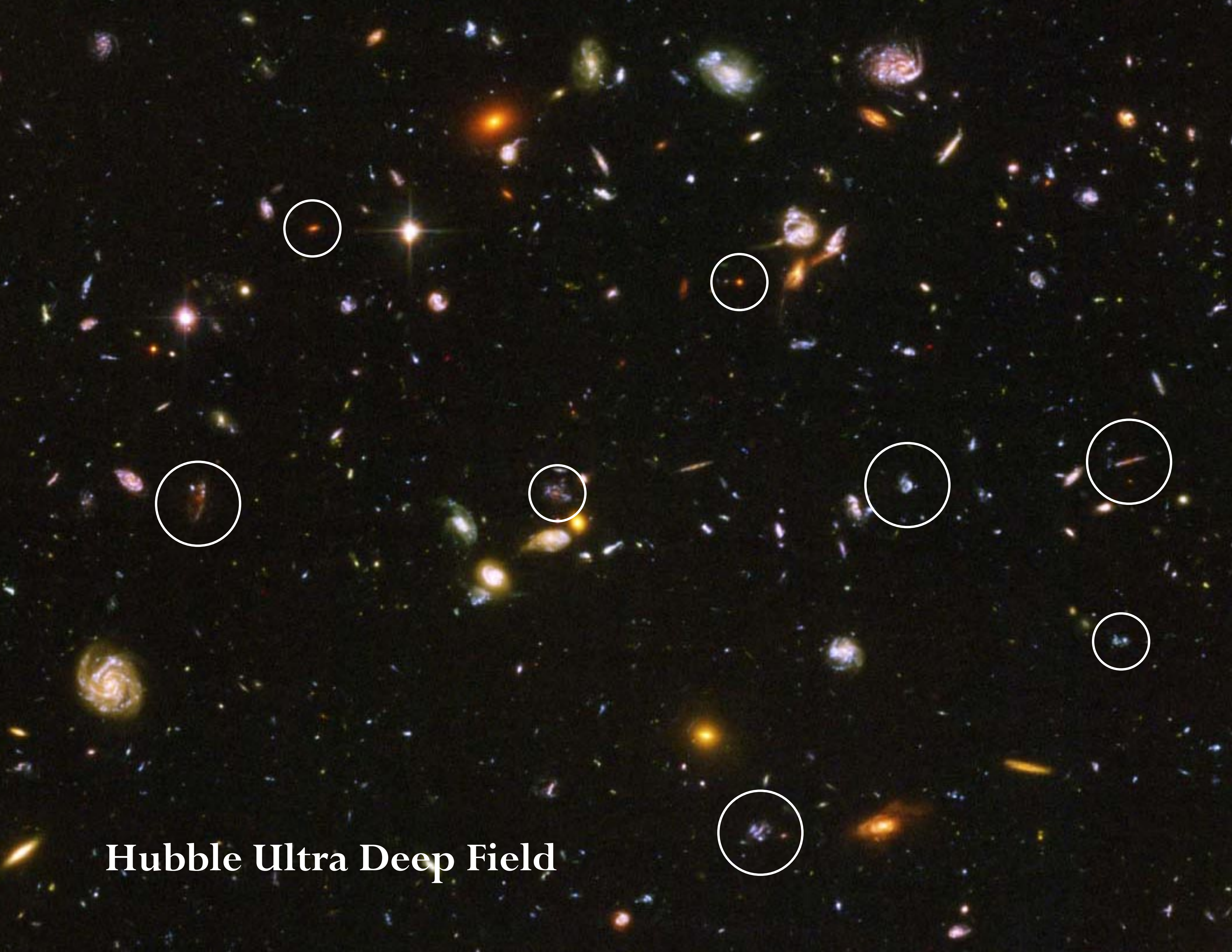


From Gaussian random fields to galaxies: non-linear dynamics of gravitational instability with N-body and hydrodynamics codes.

Cosmological simulations



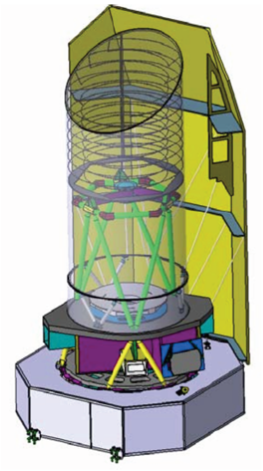
Springel *et al.*, Nature, 2006



Hubble Ultra Deep Field

Cosmological volumes

Billion dollar experiments need support from HPC

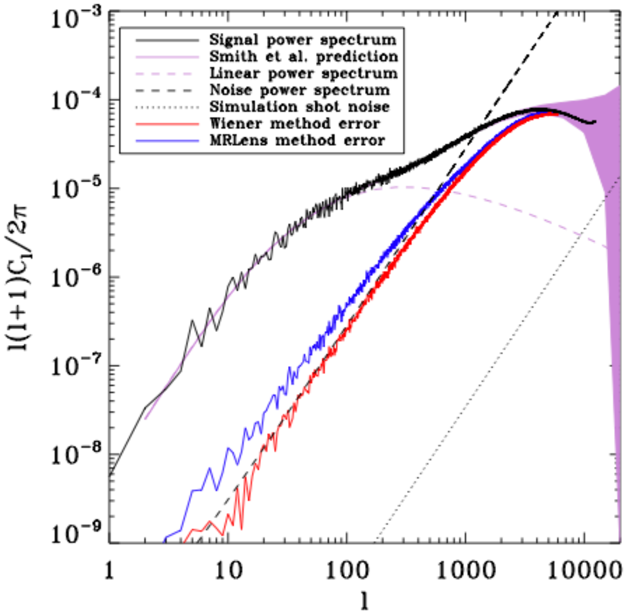


esa

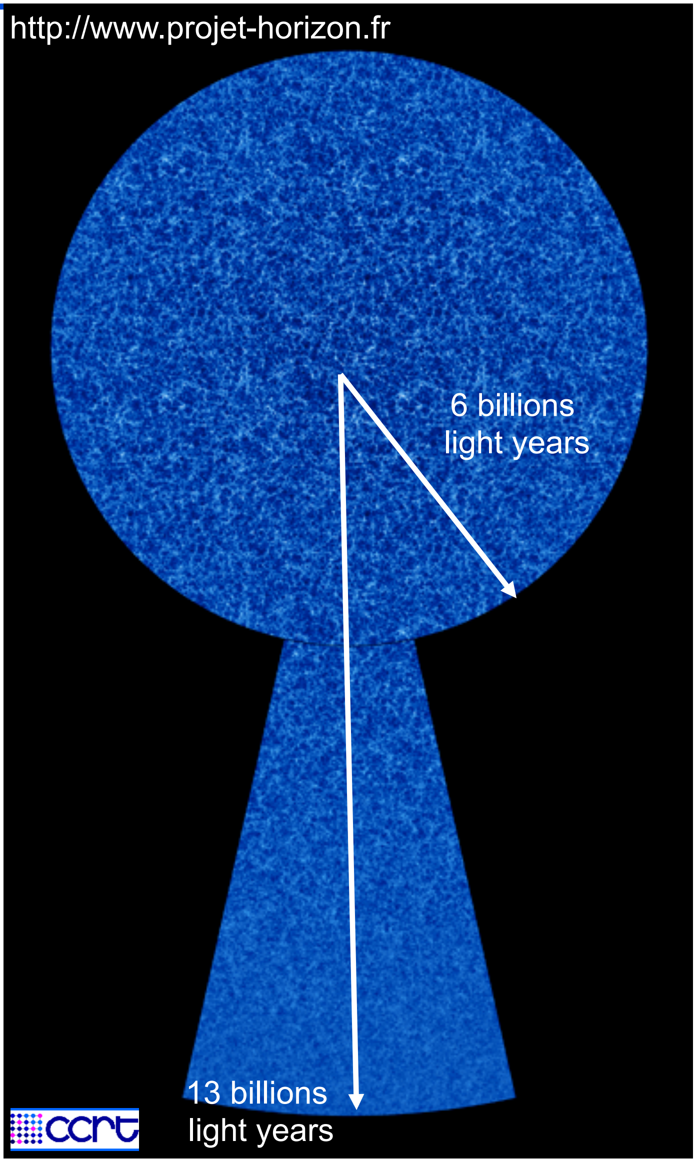
ESTEC, Noordwijk, The Netherlands

Observing the dark Universe with Euclid

17-18 November 2009



Precision cosmology

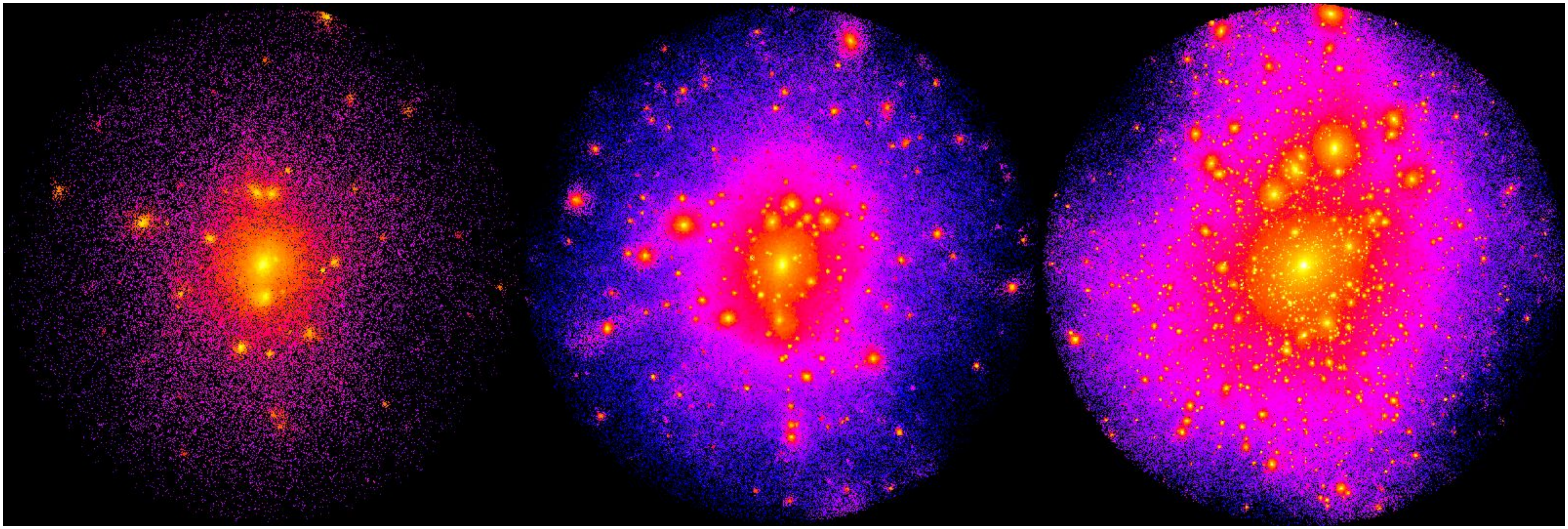


N=70,000,000,000 Teyssier *et al.* 2007

Zoom-in Simulations

Zoom-in strategy: focus computational resources on a particular Region-Of-Interest and degrade the rest of the box.

Much more demanding than full periodic box simulations.



N=100,000 1,000,000 10,000,000

From the "overmerging" problem to the "missing satellites" problem…

Moore *et al.* 1999

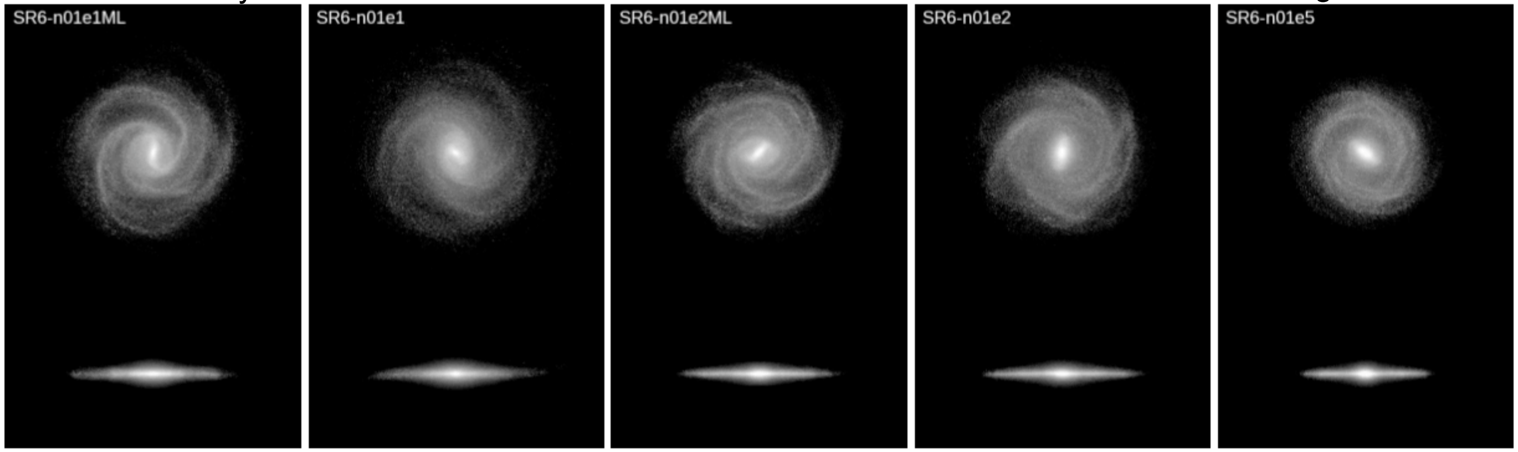
The GHALO project

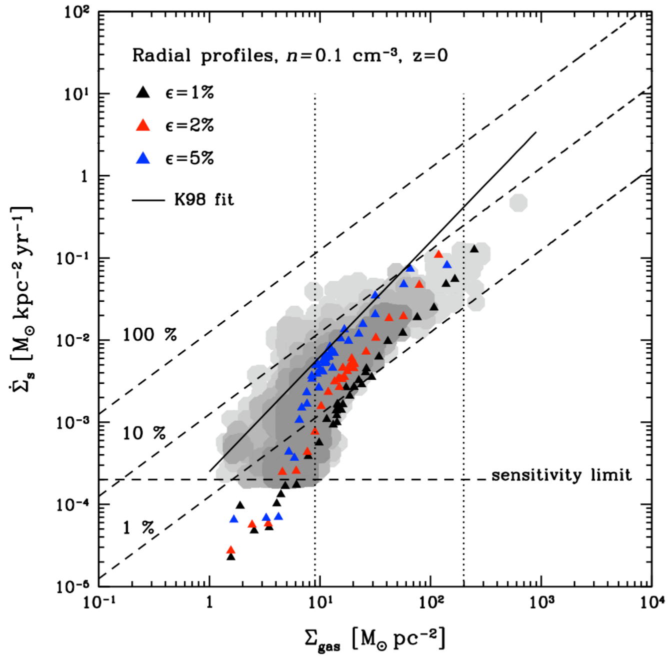
`PKDGRAV` code N=1,000,000,000 Stadel *et al.* 2009

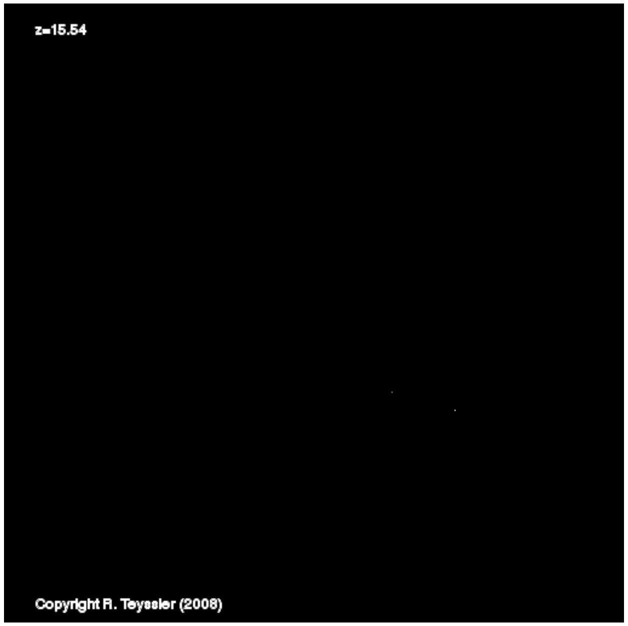
Galaxy formation: the impact of subgrid physics

Low SF efficiency

High SF efficiency







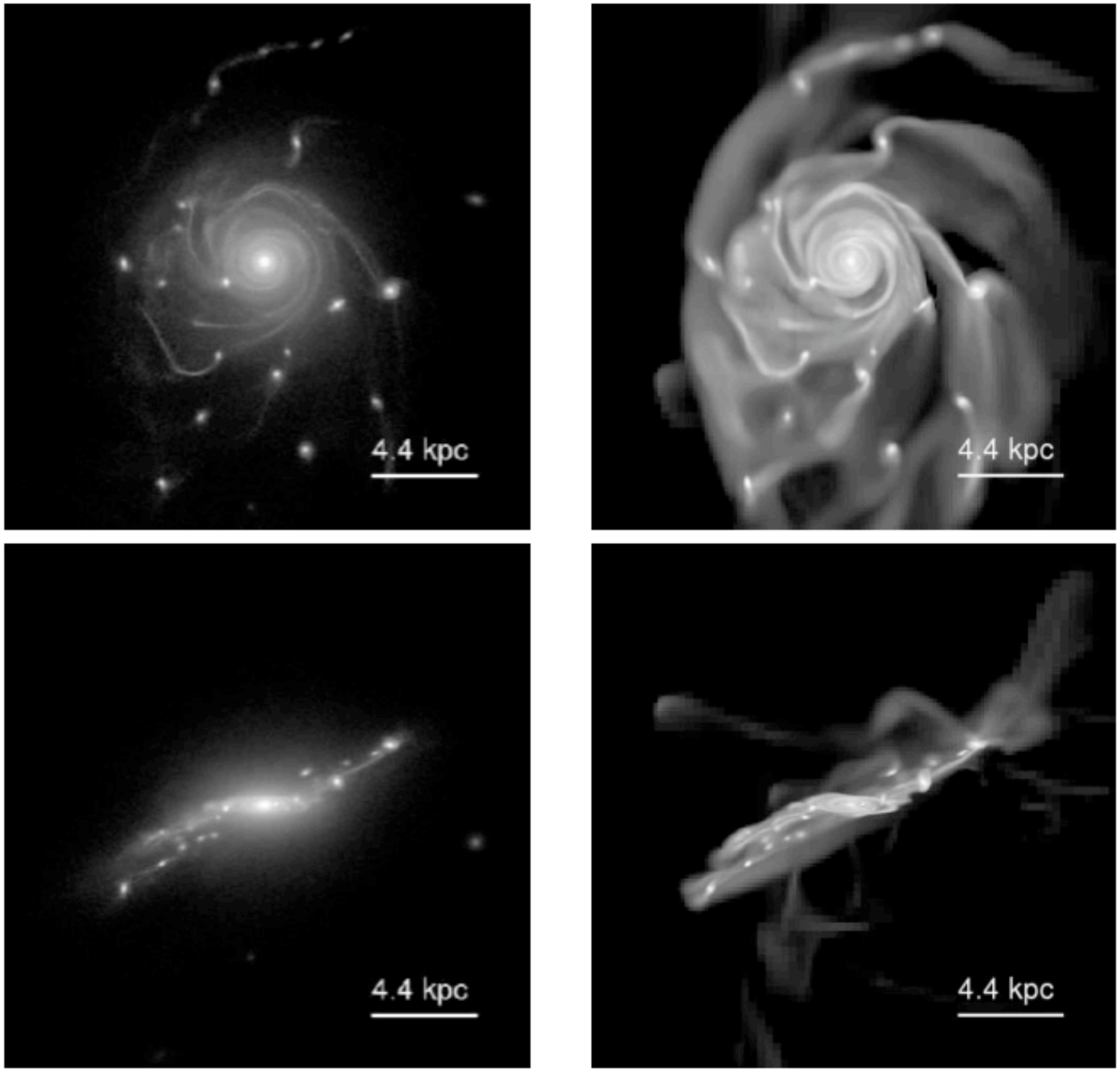
Agertz *et al.*, in prep.

Towards resolving the clumpy ISM

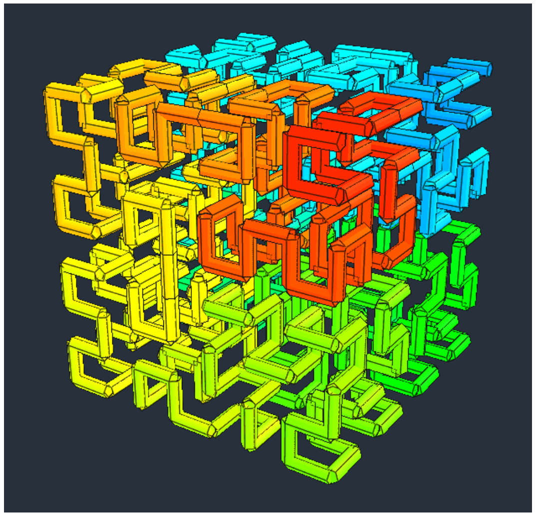
Cosmological simulation with `RAMSES`: low T metal cooling and 40 pc resolution

10^{12} Msol halo from the Via Lactea simulation
Diemand *et al.* 2006

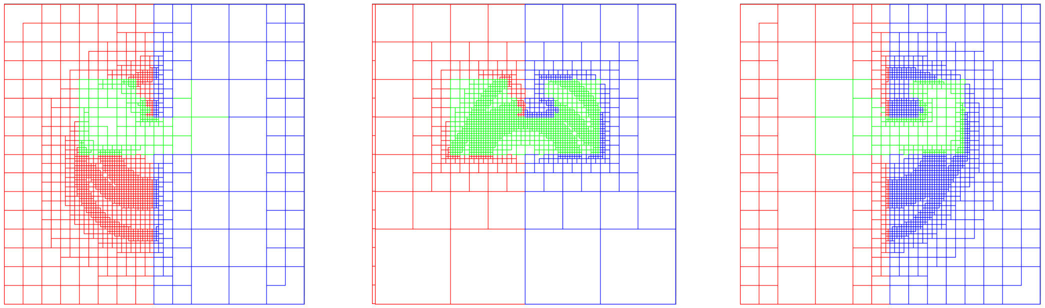
We observe for the first time disc fragmentation in a cosmological simulation.
Agertz *et al.* 2009



Domain decomposition for parallel computing



Parallel computing using the MPI library with a domain decomposition based on the *Peano-Hilbert curve* for adaptive tree-based data structure.



Peano-Hilbert binary hash key is used for domain decomposition (MPI).

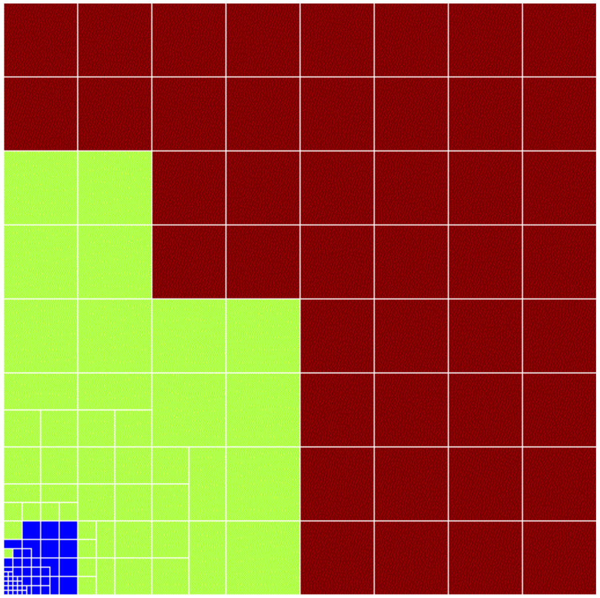
Hilbert ordering for optimal data placement in local memory (OpenMP).

Data compression based on 6D Hilbert indexing.

Implemented in our 2 codes:

- `PKDGRAV` (TREE + SPH) by J. Stadel
- `RAMSES` (PIC + AMR) by R. Teyssier

Weak-scaling up to 20,000 core.



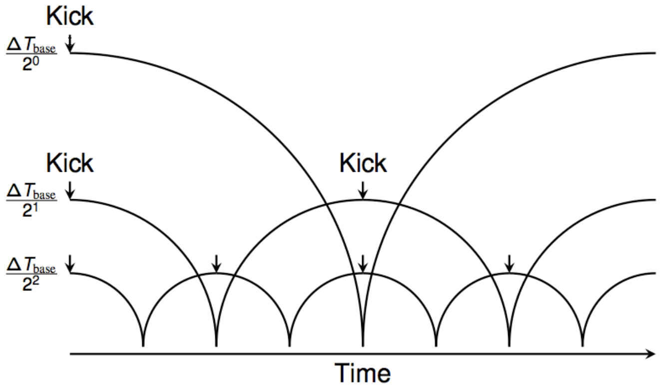
Dynamical load balancing

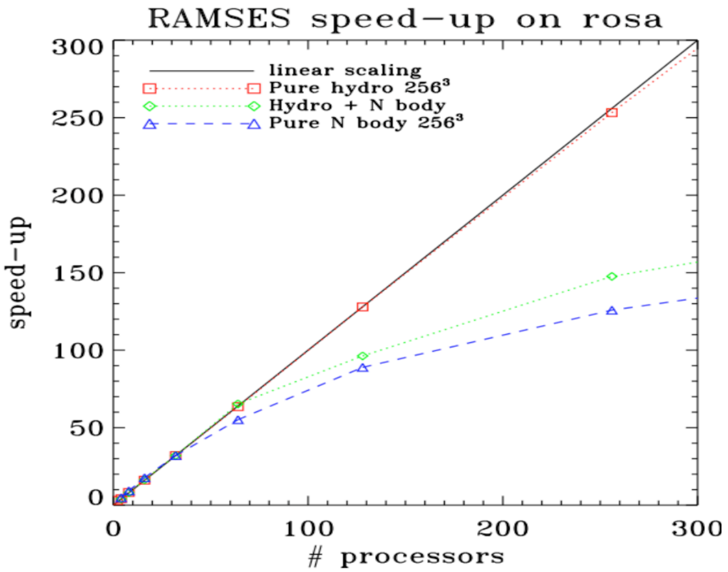
Load-balancing issue

Scaling depends on problem size and complexity.

Large dynamic range in density implies large dynamic range in time steps

Main source of load unbalance: multiple time steps and multiple species (stars, gas, DM).



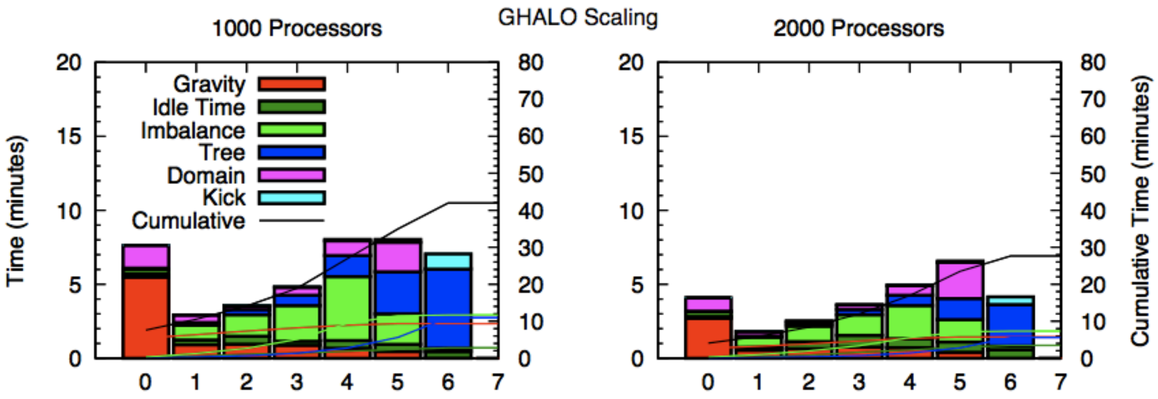


Strong-scaling example.

Problem: load balancing is performed globally.

Intermediate time steps particles are idle.

Solution: multiple tree individually load balanced



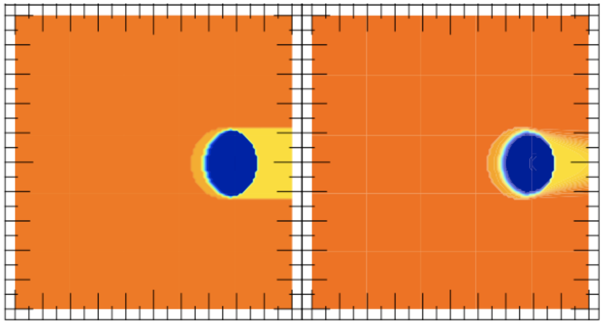
Radiative Transfer with GPU

A radiation transfer scheme with a local Eddington tensor approximation (M1 scheme)
Aubert & Teyssier, MNRAS, 2008

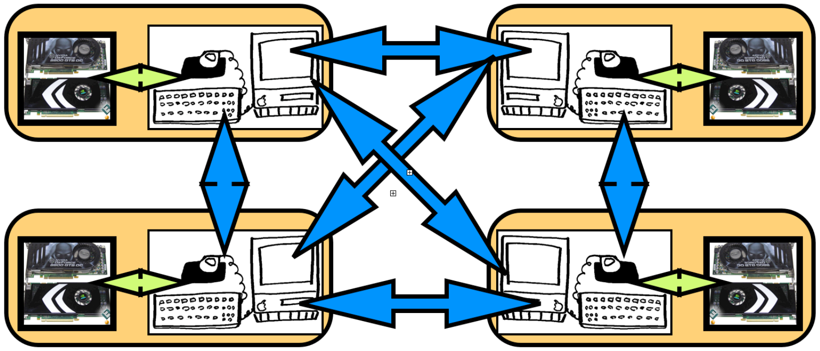
$$\frac{\partial N_\nu}{\partial t} + \nabla \mathbf{F}_\nu = -\kappa_\nu c N_\nu + S_\nu, \qquad \chi = \frac{3+4|\mathbf{f}|^2}{5+2\sqrt{4-3|\mathbf{f}|^2}}.$$

$$\frac{\partial \mathbf{F}_\nu}{\partial t} + c^2 \nabla \mathbf{P}_\nu = -\kappa_\nu c \mathbf{F}_\nu, \qquad \mathbf{D} = \frac{1-\chi}{2}\mathbf{I} + \frac{3\chi-1}{2}\mathbf{u}\otimes\mathbf{u},$$

Hyperbolic system with wave speeds close to c: use implicit or explicit time integration (ATON).



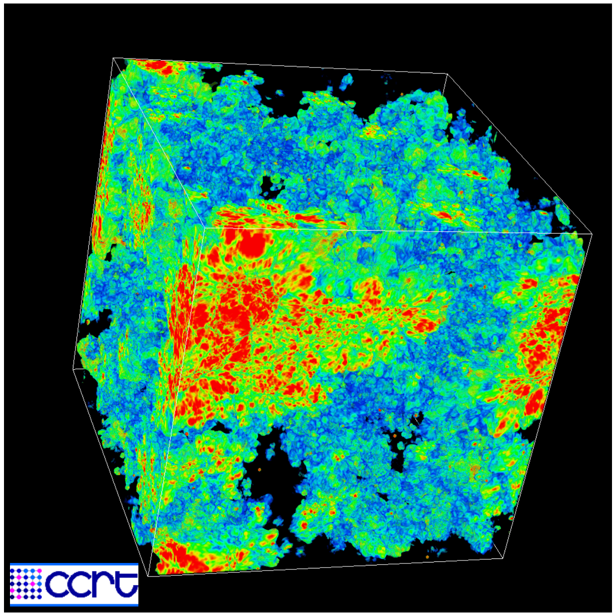
Photoionization with shadowing effect



Brute force explicit scheme using GPU acceleration (100x) on a Cartesian grid (Cuda + MPI)
Aubert & Teyssier, ApJ, 2010

Running a galaxy formation simulation on the host (384 core) with radiative transfer performed on 192 Tesla GPU in CCRT.

Cosmological reionization from first galaxies

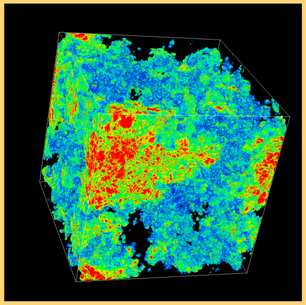


GPU computing

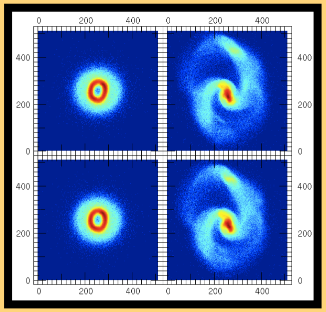
Acceleration with GPU coprocessors works well for cosmological radiative transfer: brute force strategy (explicit hyperbolic solver on a Cartesian grid) Typical acceleration ~100 compared to CPU. MPI-GPU is efficient.

Work in progress: coupling `CUDATON` with `RAMSES`.

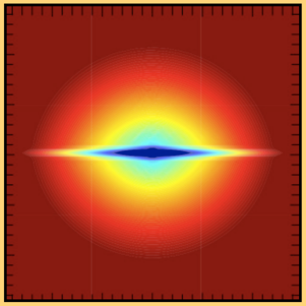
Several astrophysical codes under development with cuda, OpenCL…



Cosmological Radiative transfer x100



PM Gravity + Hydrodynamics x30



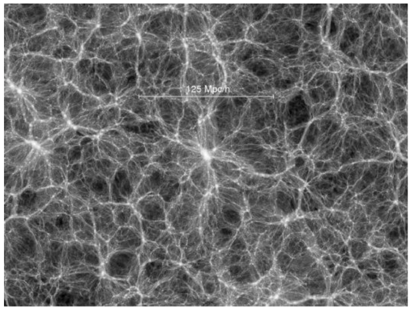
Tree code solver for gravity x?

Aubert *et al.*, ICCS, 2009
Kestener *et al.*, HPCTA, 2010

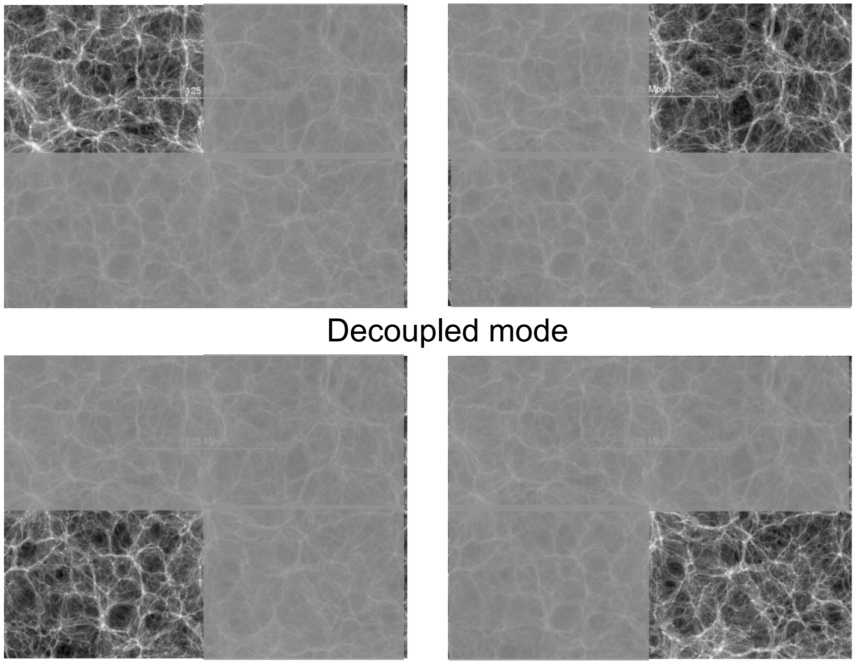
Fault-tolerant computing

Very large clusters with more than 10^6 cores will show small time-to-failure.

Because gravity is a long-range force, present-day simulations need to access the whole computational volume (fully-coupled mode).
A fault-tolerant code needs to relax this constraint: distant regions need to be decoupled.



Fully-coupled mode



Decoupled mode

Idea: use the “zoom-in” technique to segment the computational volume into independent zoom simulations. Distant particles are grouped together into massive particles and evolved locally: maximize data locality at the prize of degraded accuracy and overheads.

Fault-tolerant computing

Challenge: design an efficient scheduling middleware to schedule the jobs.

Optimize buffer region geometry for a given target force accuracy. Use multipole expansion around each sub-domain.

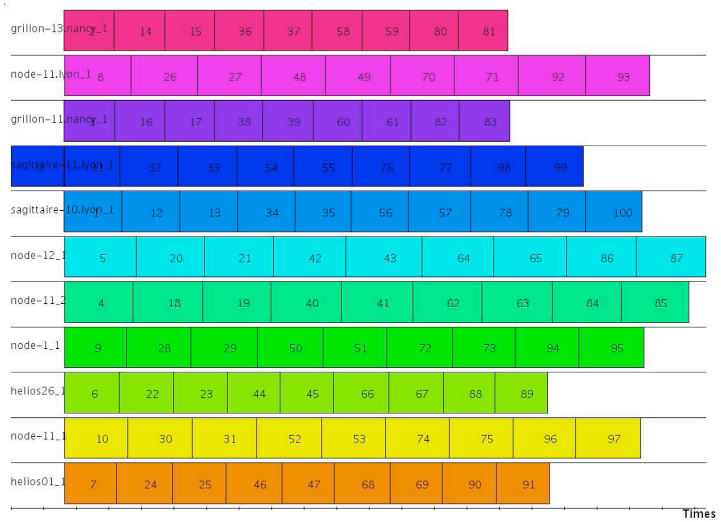
Optimize the computational load across the system: “filling up the Gantt chart”.

This will require an efficient file system.

Grid computing as a laboratory for fault-tolerant computing.

We used the DIET grid middleware to run a large scale experiment on Grid5000, the French research grid.

We obtained a 80% success rate on 3200 cores deployed over 13 sites. The main cause of failure was file system related (2 sites lost).

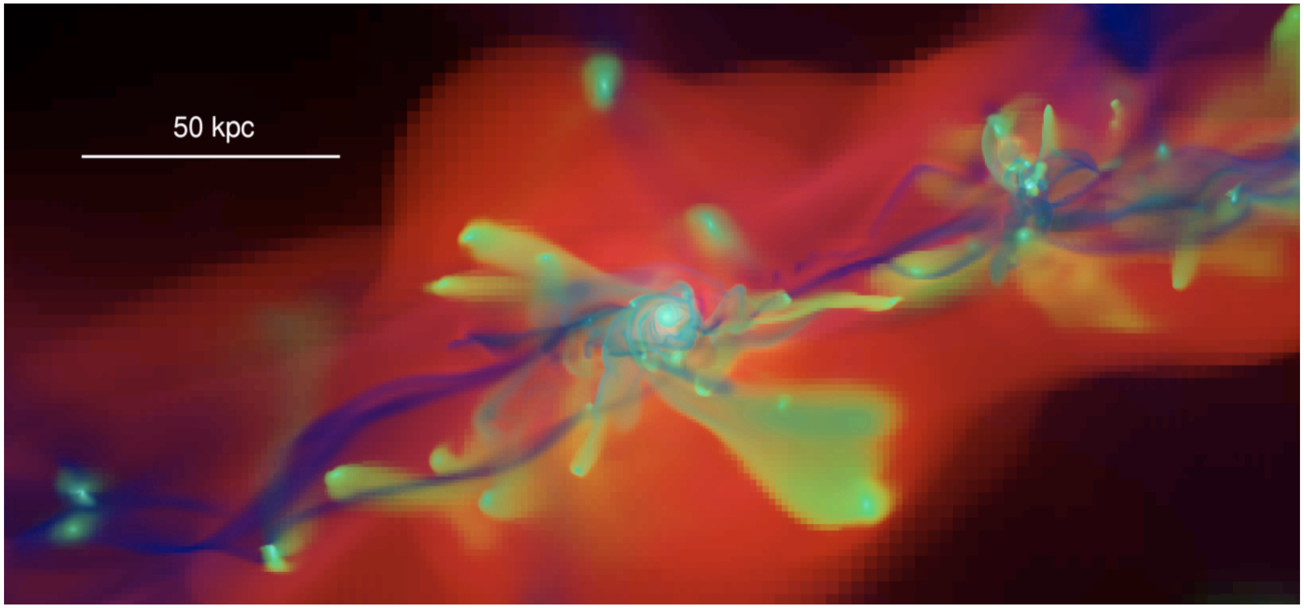


Caniou *et al.*, Fourth HPGC, 2007

Visualization

Cosmological data are based on both particles and AMR grids.

Use of the VTK library with Paraview plugins
AstroViz: A Parallel Visualization Tool for Astrophysical Simulations (Christine Moran)
Current solution: convert AMR cells into particles
Importing particle and AMR data into Visit (in collaboration with Jean Favre).



Issue to be solved:
- unstructured octree AMR grid to be supported.
- 3D parallel rendering of particle data.
- quick data exploration versus final data presentation

Project tasks and team

WP1: Multiple Tree gravity solver and development of `NEW_CODE`

WP2: OpenMP and MPI hybrid parallelization of `RAMSES` and `PKDGRAV`
GPU acceleration for radiation, chemistry and gravity solvers

WP3: Fault-tolerant scheduler and automatic zoom-in generator

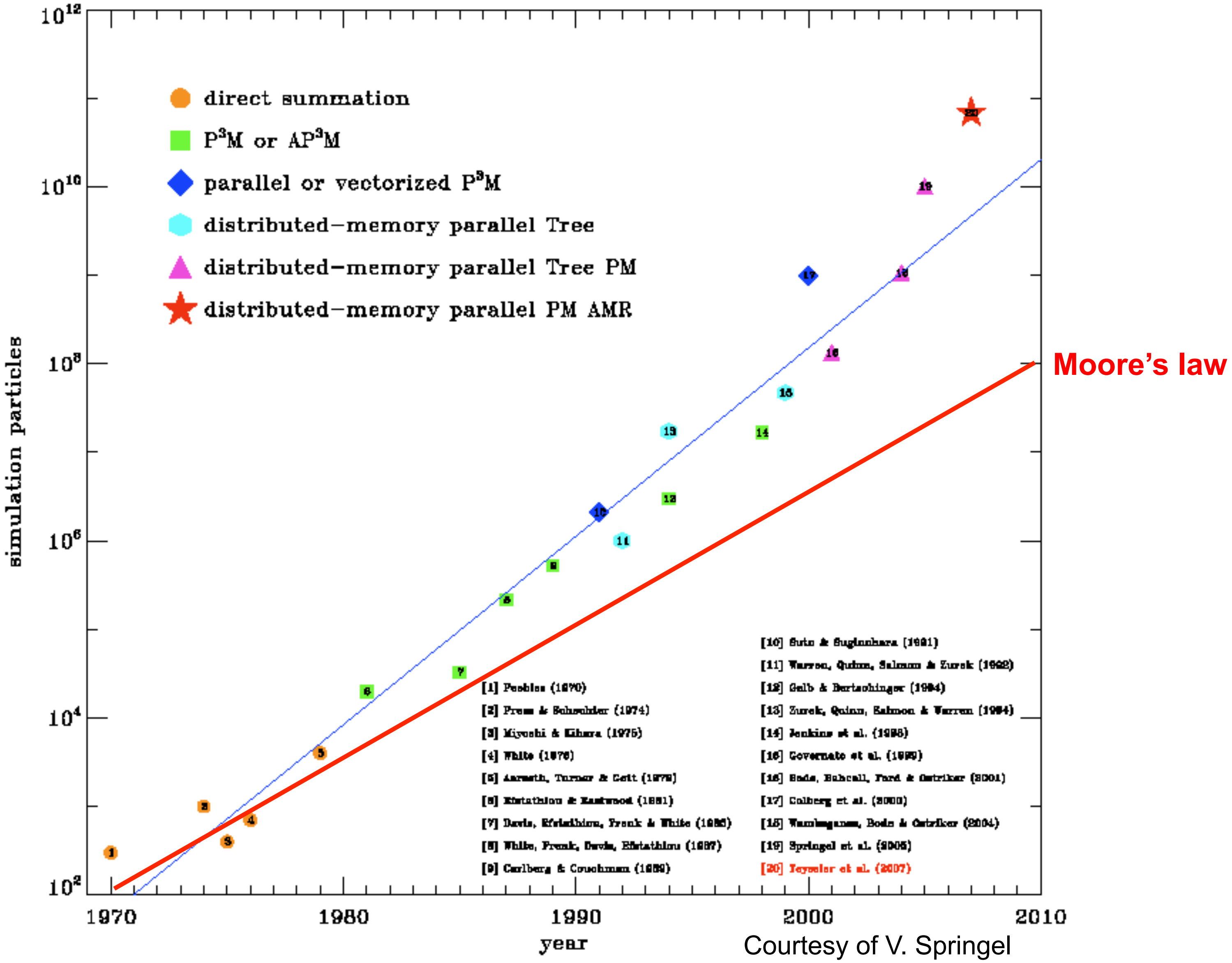
WP4: Parallel data visualization
Parallel I/O and data compression
Parallel halo finder

Time allocated at CSCS:
- High-Impact project 2009
- Production project 2010

Thank you !

- George Lake (PI)
- Romain Teyssier (co-I)
- Ben Moore (co-I)
- Joachim Stadel (co-I)
- Jonathan Coles (postdoc)
- Markus Wetzstein (postdoc)
- Rok Roskar (postdoc)
- Michael Busha (postdoc)
- Doug Potter (PhD student)
- Christine Moran (PhD student)
- Sarah Nickerson (PhD student)

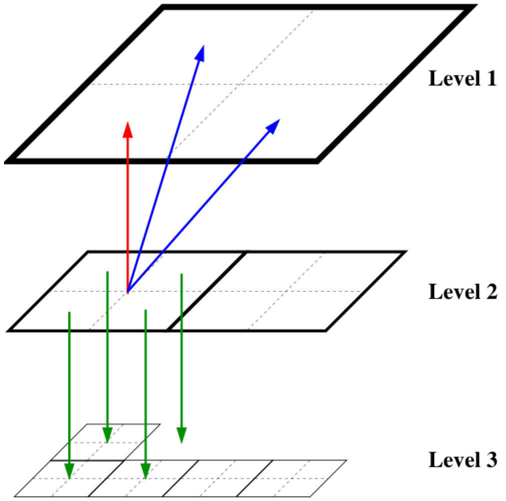
Cosmological N body simulations



Courtesy of V. Springel

RAMSES: a parallel AMR code

- Graded octree structure: the cartesian mesh is refined on a cell by cell basis
- Full connectivity: each oct have direct access to neighboring parent cells and to children octs (memory overhead 2 integers per cell).
- Optimize the mesh adaptivity to complex geometry but CPU overhead can be as large as 50%.



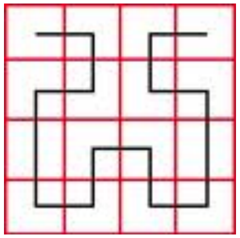
N body module: Particle-Mesh method on AMR grid (similar to the ART code). Poisson equation solved using a multigrid solver.

Hydro module: unsplit second order Godunov method with various Riemann solvers and slope limiters.

Time integration: single time step or fine levels sub-cycling.

Other: Radiative cooling and heating, star formation and feedback.

MPI-based parallel computing using time-dependant domain decomposition based on Peano-Hilbert cell ordering.



Download at http://irfu.cea.fr/Projets/Site_ramses

PKDGRAV2: JS and Doug Potter

1. Fast Multipole Method (FMM), like W.Dehnen FALCON code, but 5th-order expansion of the potential instead of 3rd. Uses reduced moments.
2. New fast and low “rung-noise” dynamical timestepping algorithm.
3. Memory usage reduced by about 70% to 200 bytes/particle.
4. Use of SSE2/3 and Altivec assembly code for interactions.
5. Over 20 times faster for large simulations than PKDGRAV.
6. New I/O system: HDF5 file support, concept of I/O CPUs (RAM Disk).
7. For Solar System work: Very Active Particles, TreeHermite and TreeSymba! R. Morishima
8. Python interface to many high level functions - Analysis!
9. Built in parallel GRAFIC1 and GRAFIC2 initial conditions generation.
10. *No Hydrodynamics, yet...*