

MILIARDI DI PARTICELLE IN UN'IMMAGINE: VISUALIZZAZIONE HPC CON SPLOTCH



INAF

ISTITUTO NAZIONALE
DI ASTROFISICA

Claudio Gheller

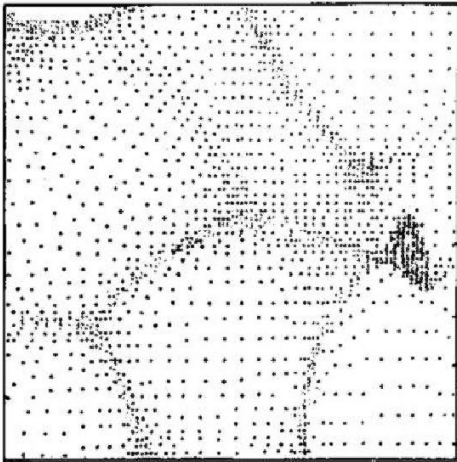
Institute for Radio Astronomy

INAF

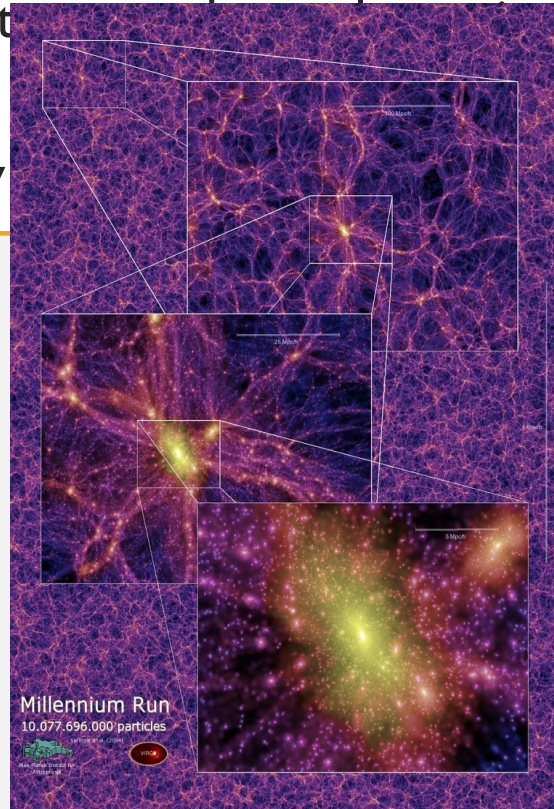
Napoli, 6/10/2026

N-BODY SIMULATIONS

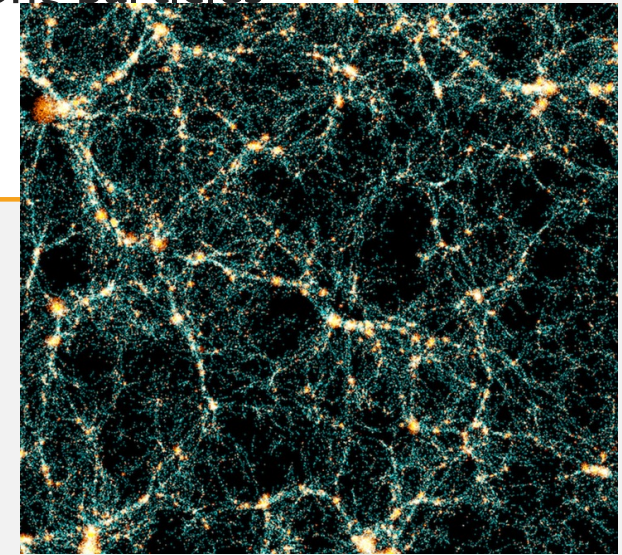
- N-body and SPH method are widely used in CFD and Astrophysics
- These methods sample the fluid with points following their dynamics
- In the simplest case, only gravitational forces among particles



Distribution of particles in a N-body simulation by the Zeldovich group in 1975: 1550 particles

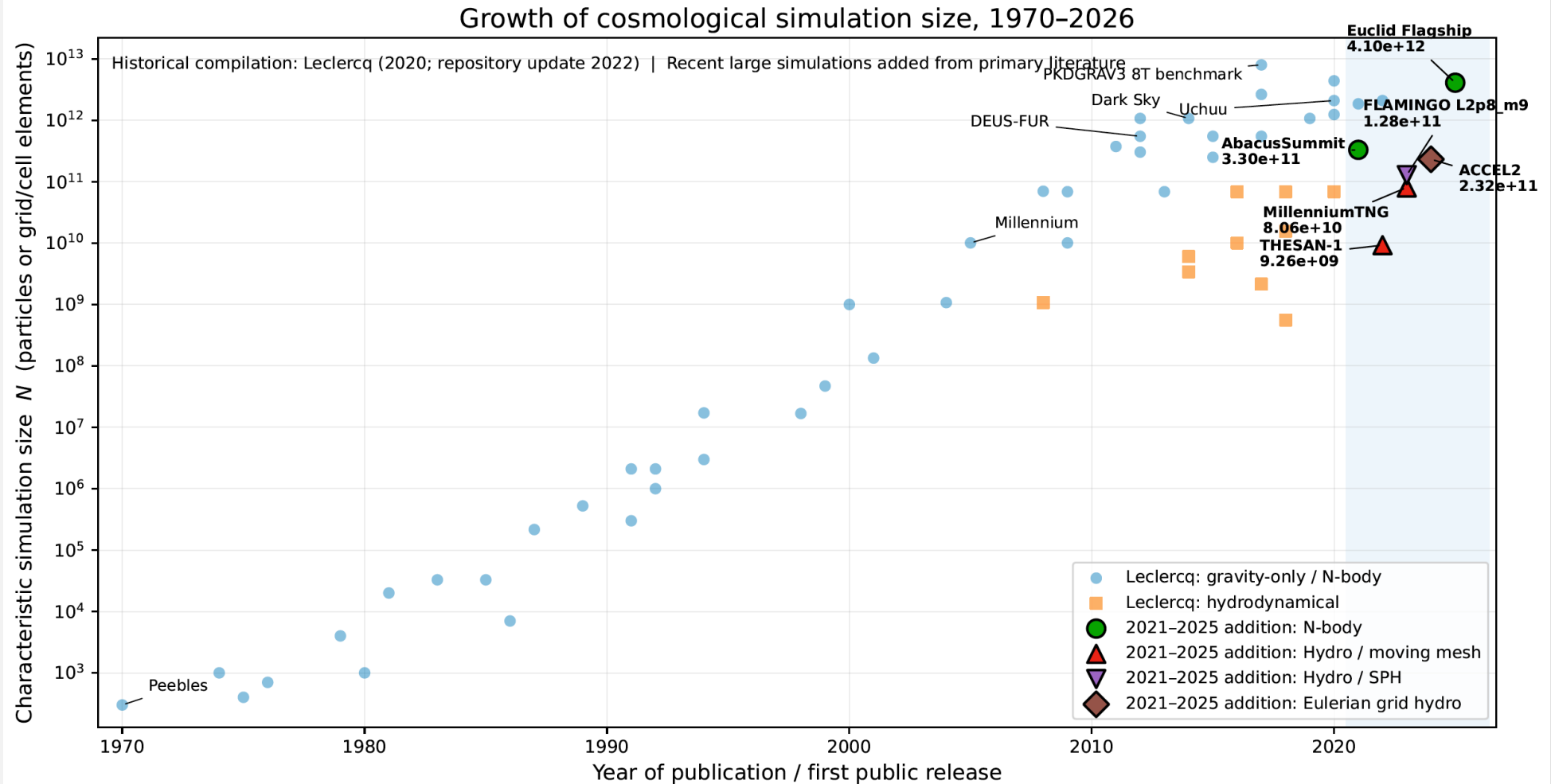


Millennium Simulation (Springel et al. 2005)
10 Billion particles
Gadget code



Euclid Flagship Simulation (Castander et al. 2025)
4.1 Trillion particles
Pkdgrav code

THE HISTORY OF COSMOLOGICAL N-BODY SIMULATIONS



Counting convention: the historical Leclercq N values are preserved. For recent hydro runs, the plotted N is the characteristic one-species/grid count (e.g. 5040^3 for FLAMINGO), not the summed gas+DM+neutrino total. The 2017 8T PKDGRAV3 point is a benchmark; the completed 2025 Euclid Flagship contains $16000^3 = 4.096 \times 10^{12}$ particles.

THE TOP LIST OF COSMOLOGICAL N-BODY SIMULATIONS

Simulation	Year	Physics	Box side	Resolution elements	Code	Main machine
Euclid Flagship	2025	DM-only	3.6 Gpc/h	$16000^3 \approx 4.1$ trillion	PKDGRAV3	Piz Daint
FLAMINGO	2023–26	full hydro + ν	2.8 Gpc	$\approx 3 \times 10^{11}$	SWIFT	COSMA8
FLAMINGO-10k	2024–26	DM-only	2.8 Gpc	$10080^3 \approx 1.02$ trillion CDM + ν	SWIFT	COSMA8
MillenniumTNG	2023	full hydro	740 Mpc	$2 \times 4320^3 \approx 161$ billion	AREPO	SuperMUC-NG
MillenniumTNG large DMO	2023	DM + ν	≈ 3 Gpc	$\approx 10,240^3$ DM + ν	GADGET-4	COSMA8
AbacusSummit	2021–22	DM-only	2 Gpc/h	$6912^3 \approx 330$ billion / run	Abacus	Summit
Uchuu	2021	DM-only	2 Gpc/h	$12800^3 \approx 2.1$ trillion	GreeM	Aterui II
Last Journey	2021	DM-only	5.025 Gpc	1.24 trillion	HACC	Mira
THESAN-I	2022–24	radiation-MHD	95.5 cMpc	$2 \times 2100^3 \approx 18.5$ billion	AREPO-RT	SuperMUC-NG

THE TOP 500 RANK

Rank	Name	Site	Manufacturer	Country	Year	Total Cores	Accelerator/ Co-Processor Cores	Rmax [TFlop/s]	Rpeak [TFlop/s]	Power (kW)	Energy Efficiency [GFlops/Watts]
1	LineShine	National Supercomputing Centre in Shenzhen (NSCS)	Shenzhen Cloud Computing Center	China	2025	13 789 440		2 198 400.00	2 735 824.90	42 220.00	52.07
2	El Capitan	DOE/NNSA/LLNL	HPE	United States	2024	11 340 000	10 260 000	1 809 000.00	2 821 104.00	29 684.62	60.94
3	Frontier	DOE/SC/Oak Ridge National Laboratory	HPE	United States	2021	9 066 176	8 451 520	1 353 000.00	2 055 716.99	24 607.00	54.98
4	Aurora	DOE/SC/Argonne National Laboratory	Intel	United States	2023	9 264 128	8 159 232	1 012 000.00	1 980 006.00	38 698.36	26.15
5	JUPITER Booster	EuroHPC/FZJ	Bull	Germany	2025	4 801 344	3 106 752	1 000 000.00	1 226 280.64	15 793.81	63.32
6	HPC7	Eni S.p.A.	HPE	Italy	2026	3 461 472	3 173 760	571 500.00	861 126.32	8 735.00	65.43
7	Eagle	Microsoft Azure	Microsoft Azure	United States	2023	2 073 600	1 900 800	561 200.00	846 835.20		
8	HPC6	Eni S.p.A.	HPE	Italy	2024	3 143 520	2 930 400	477 900.00	606 965.76	8 460.90	56.48
9	Supercomputer Fugaku	RIKEN Center for Computational Science	Fujitsu	Japan	2020	7 630 848		442 010.00	537 212.00	29 899.23	14.78
10	Alps	Swiss National Supercomputing Centre (CSCS)	HPE	Switzerland	2024	2 121 600	1 372 800	434 900.00	574 841.28	7 124.00	61.05
11	LUMI	EuroHPC/CSC	HPE	Finland	2023	2 752 704	2 566 080	379 700.00	531 505.15	7 106.82	53.43
12	Leonardo	EuroHPC/CINECA	Bull	Italy	2023	1 824 768	1 714 176	241 200.00	306 311.16	7 493.74	32.19

CHALLENGES

Data generated by these simulation is HUGE.

How to process it?

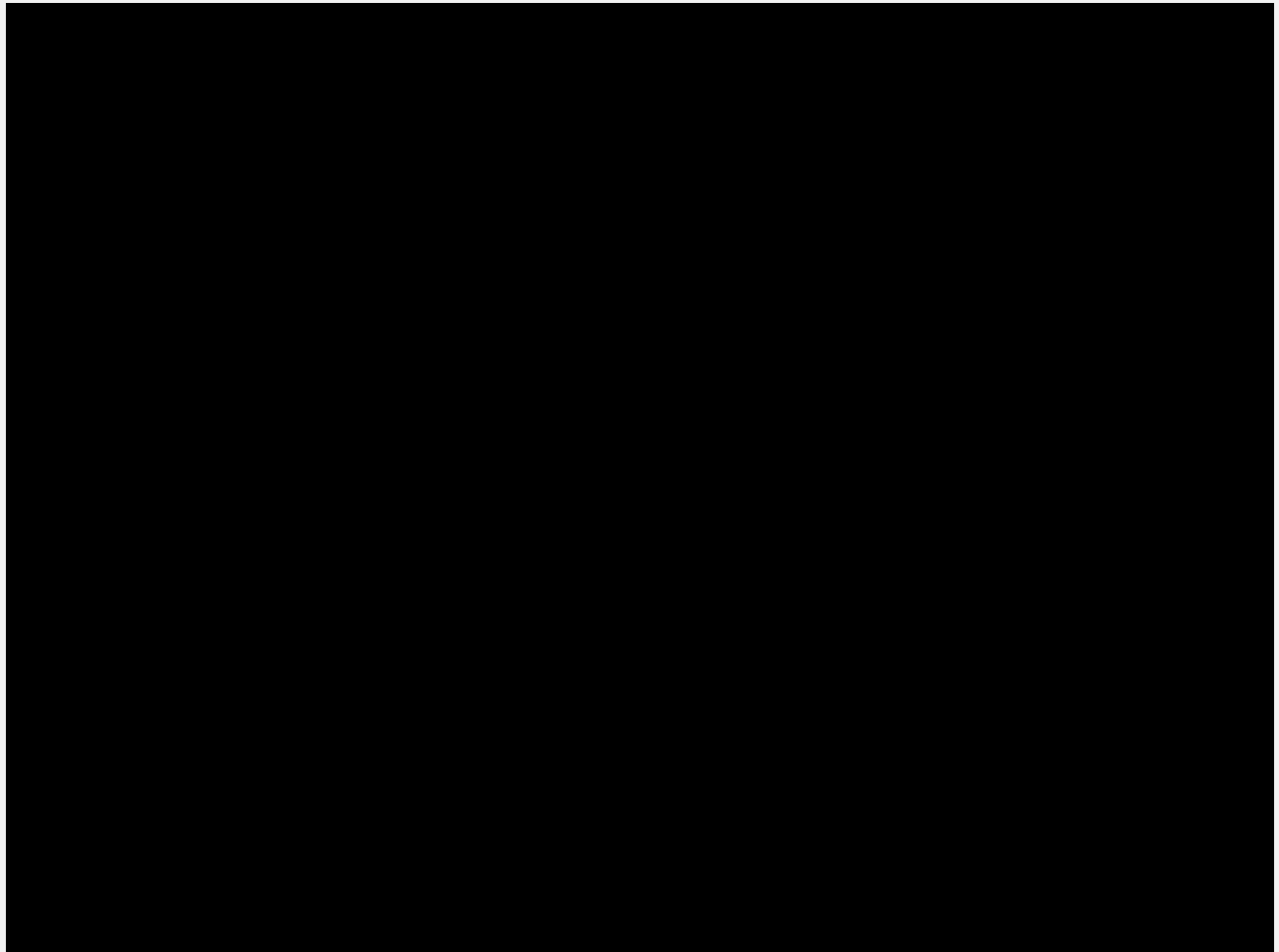
→ **STORAGE SIZE (PB)**

→ **MEMORY SIZE (TB)**

→ **COMPUTING TIME (DAYS)**

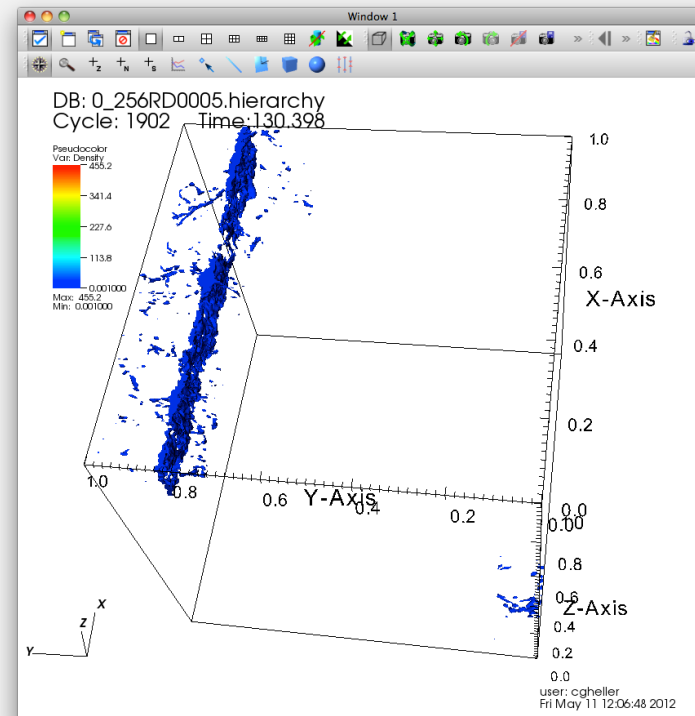
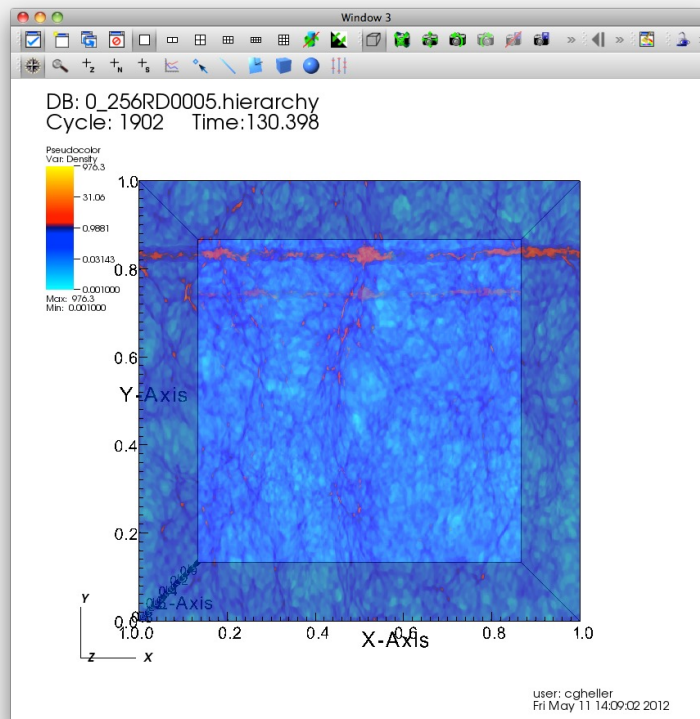
THE POWER OF VISUALIZATION

- Visualization offers an intuitive and immediate insight into data
- What takes hours for a CPU can take a glance for the human eye!!!
- The visualization process plays a fundamental role in understanding data.
- Video by Klaus Dolag



THE POWER OF VISUALIZATION

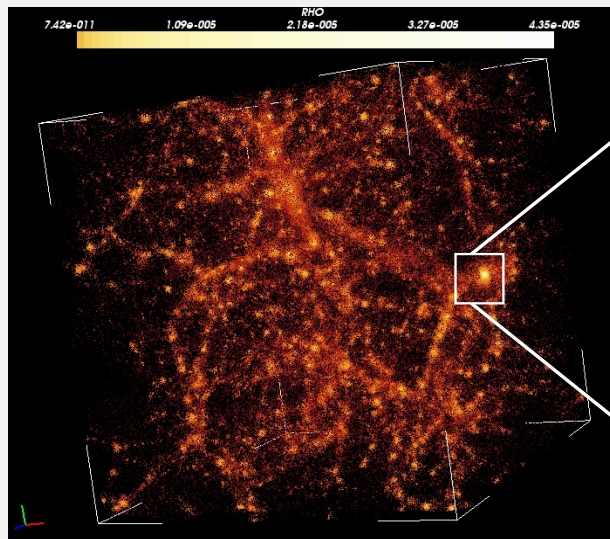
- Problems in the data can clearly emerge looking at it



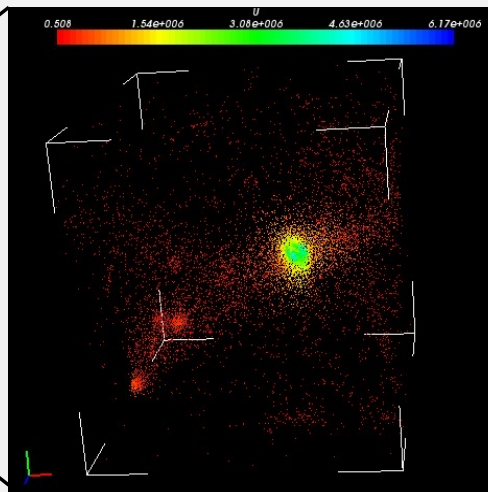
THE POWER OF VISUALIZATION

- Find interesting objects/features

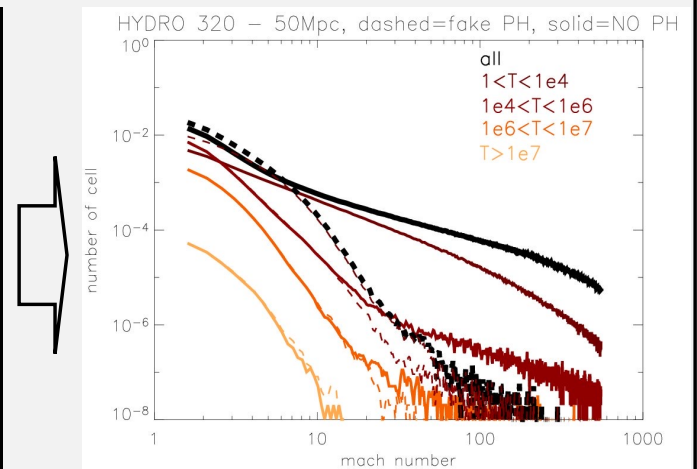
Data is too large!!!



Extract a sub region



Perform the analysis

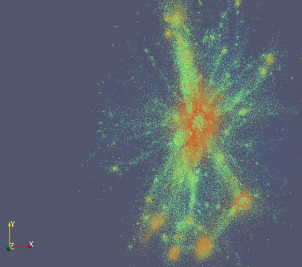


SPLOTCH FOR 3D POINTS VISUALIZATION

Splootch is a **ray-casting algorithm** for effective visualization of **point-like** datasets:

- **High Quality**
- **3D Visualization**
- **of Point Like Scientific Data**
- **of ANY SIZE**

PARAVIEW



SPLOTCH



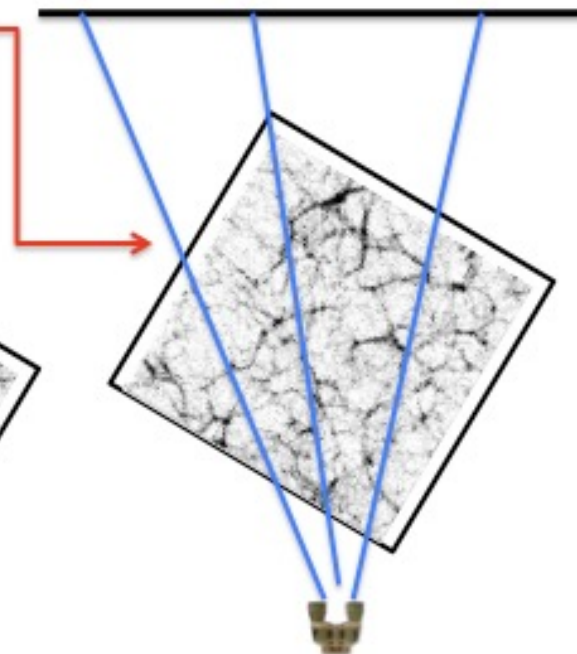
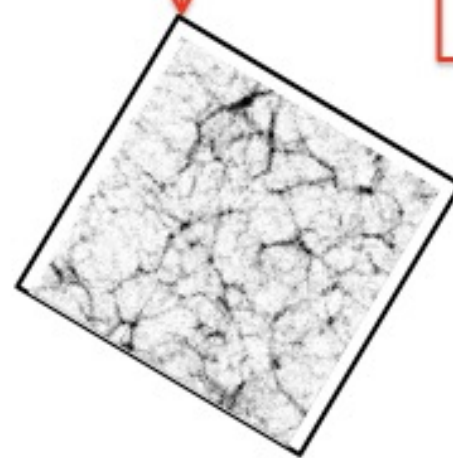
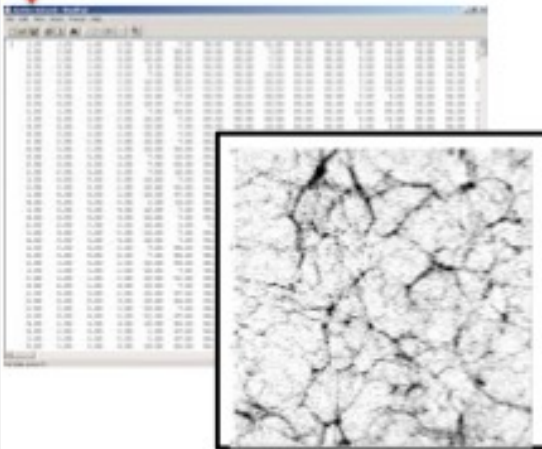
- **Completely open source and self contained**
 - No dependencies
 - Easy to compile
- **Standard C++ based; portable to any platform/compiler**
 - Easily extensible
 - Usable anywhere
 - Scriptable
 - Support for animations
- **Can exploit (almost) any HPC system**
 - Reduced time to solution
 - Big Data (any size?) can be processed

HOW DOES SPLOTCH WORK?

$$\frac{d\mathbf{I}(\mathbf{x})}{dx} = (\mathbf{E}_P - \mathbf{A}_P \mathbf{I}(\mathbf{x})) \rho_P(\mathbf{x})$$

Main steps:

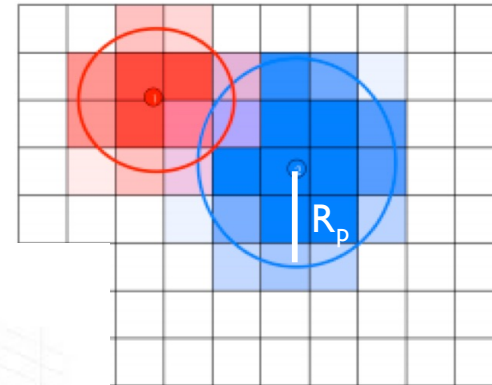
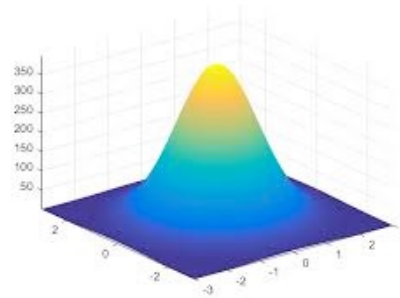
- Read data
- Set the view (rasterization step)
- Ray tracing (rendering step)
- Save the image



THE RENDERING PROCEDURE: WEIGHTS

$$\frac{d\mathbf{I}(\mathbf{x})}{dx} = (\mathbf{E}_P - \mathbf{A}_P \mathbf{I}(\mathbf{x})) \rho_P(\mathbf{x})$$

- Each particle influence a given volume, characterized by a smoothing radius R_p (just like for SPH)
- A smoothing kernel is defined to distribute a particle associated quantity on the volume. The kernel is currently a gaussian:
 $\rho_P(\mathbf{x})$ is calculated
- The sigma of the gaussian is a fraction of R_p
- The contribution of the particle on a pixel is calculated



THE RENDERING PROCEDURE: INTENSITY

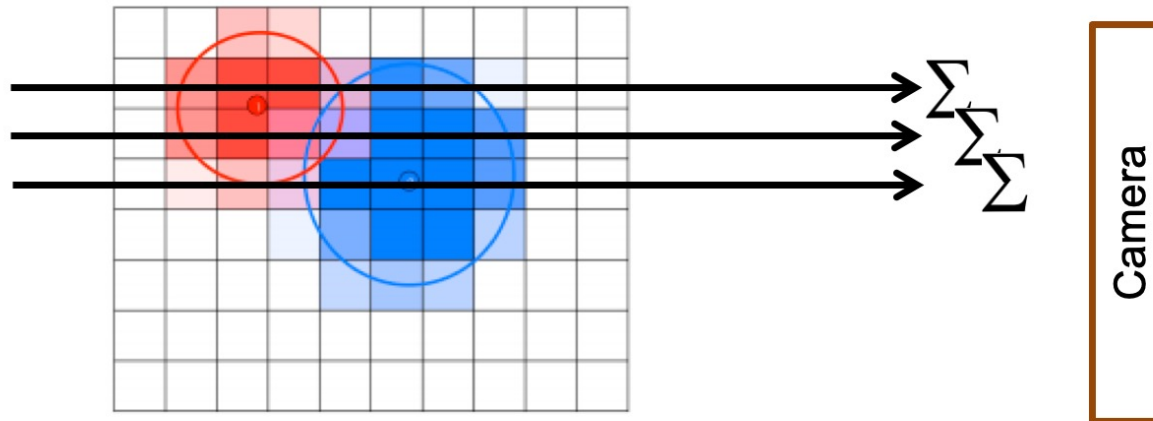
$$\frac{dI(x)}{dx} = (E_p - A_p I(x)) \rho_p(x)$$

- E_p and A_p are the emission and absorption coefficients
- They give the intensity
- In the current implementation $A_p = f(E_p)$
- E_p = available variable (density, temperature)
- Both color tables and RGB are supported (for the latter RT equation is solved for each component and three variables are loaded)

THE RENDERING PROCEDURE: INTEGRATION

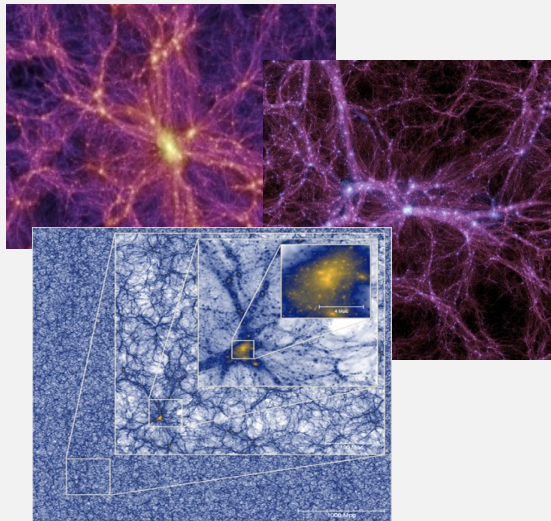
$$\frac{d\mathbf{I}(\mathbf{x})}{dx} = (\mathbf{E}_p - \mathbf{A}_p \mathbf{I}(\mathbf{x})) \rho_p(\mathbf{x})$$

- Particles are sorted with distance from the camera
- Contribution from each particle are summed along the line of sight
- Integration order can be neglected in case of low optical depth



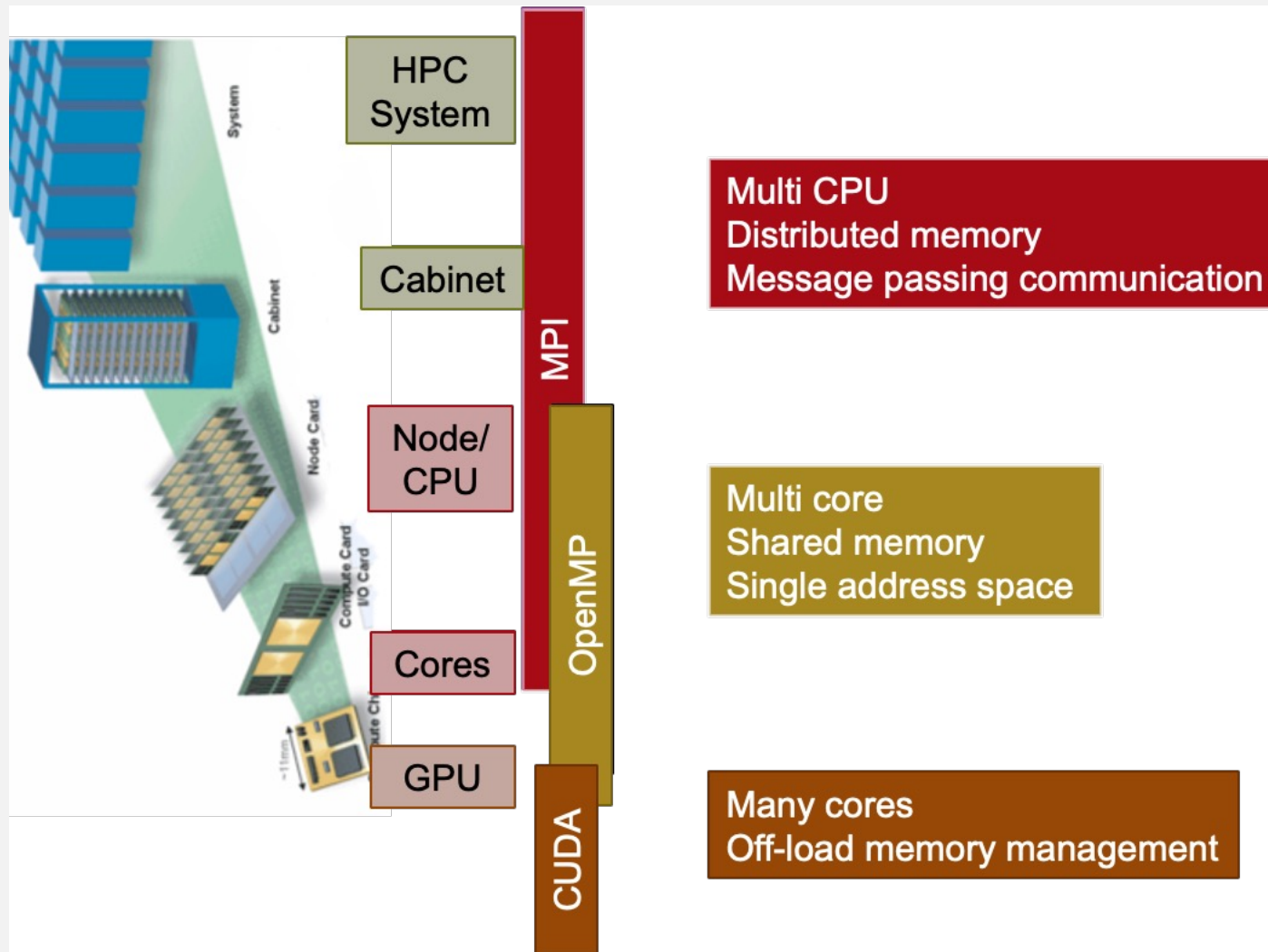
THE NEED FOR HPC FOR N-BODY SIMULATIONS

Pure gravity (N-body):	Size:
10^{12} particles	10^{12} elements
Three 3D coordinates + one variable	4×10^{12} elements
Each variable is a float number: 4 bytes	16 TB of MEMORY!

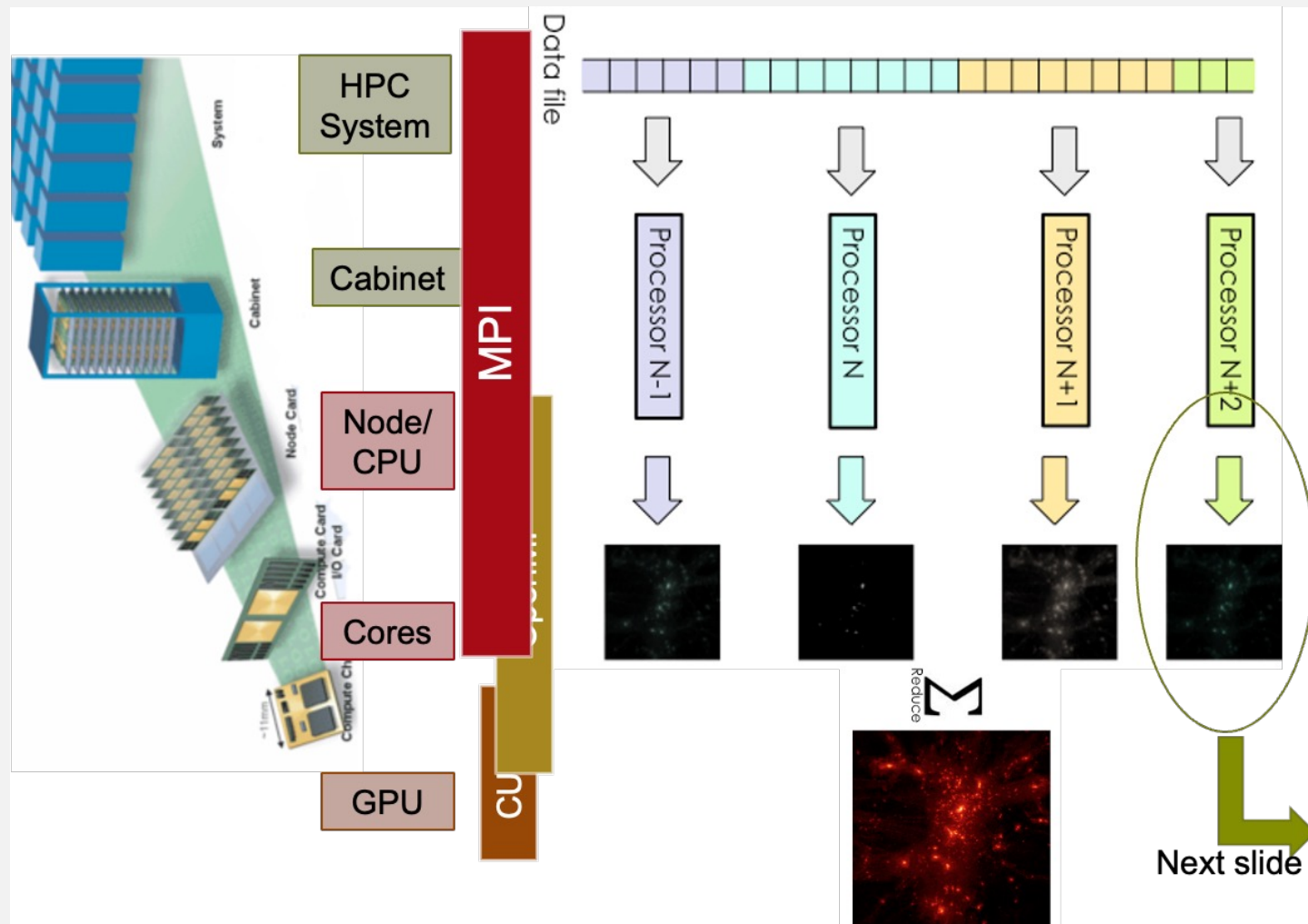


Not for a laptop!!! We need supercomputer

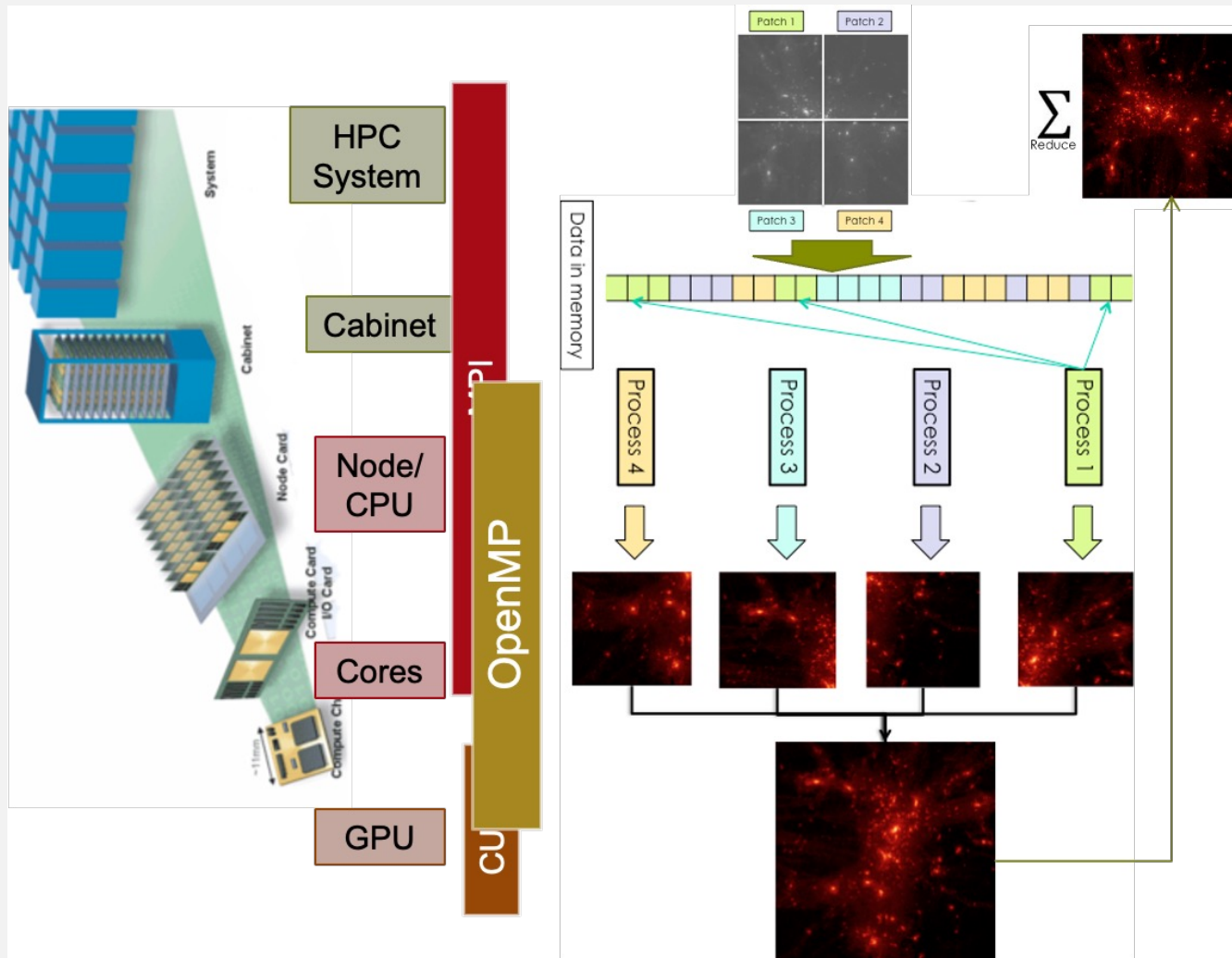
HPC SYSTEMS: THE CURRENT (HYBRID) SCENARIO



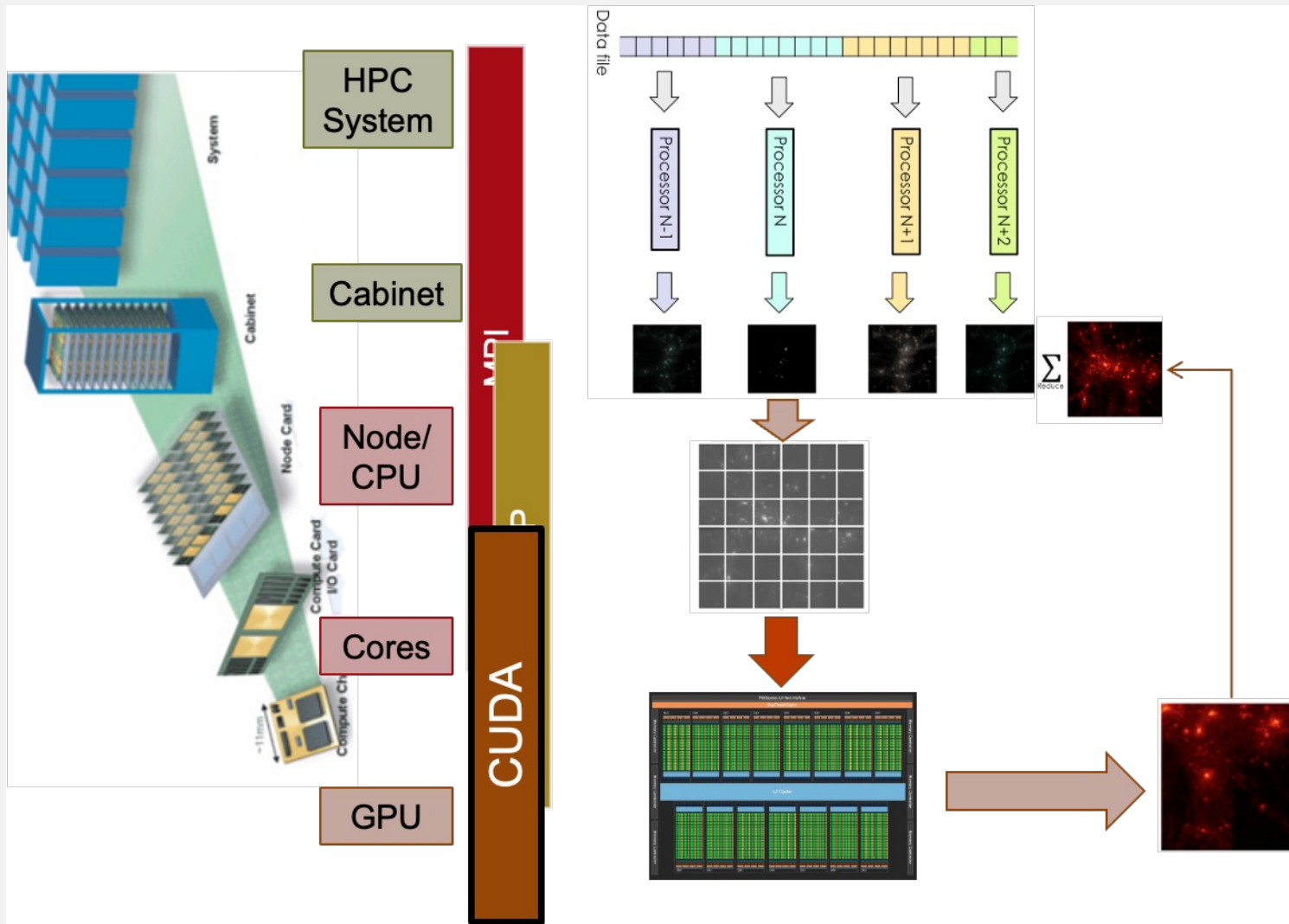
SPLOTCH: MPI IMPLEMENTATION



HYBRID IMPLEMENTATION



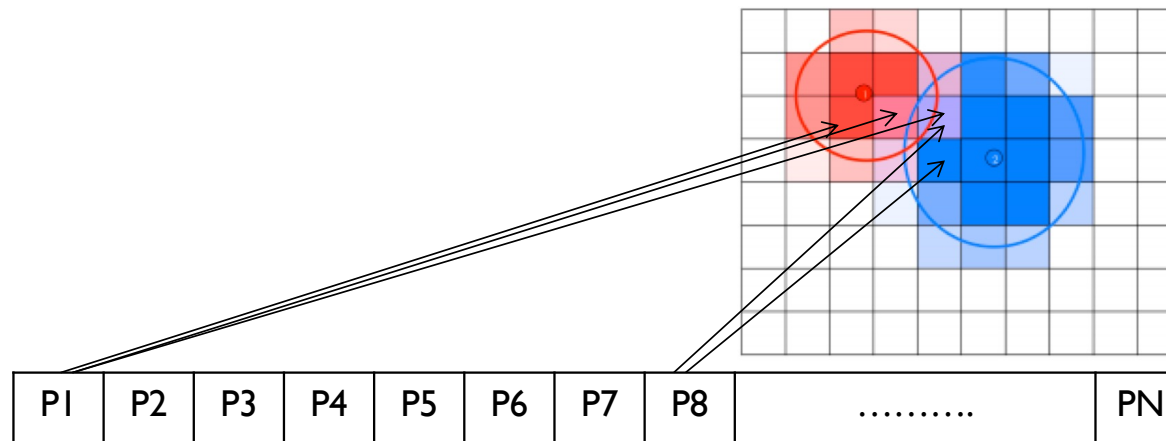
GPU IMPLEMENTATION



GPU RENDERING

One thread per particle:

- Highly unbalanced
- Race conditions prone



One thread per pixel:

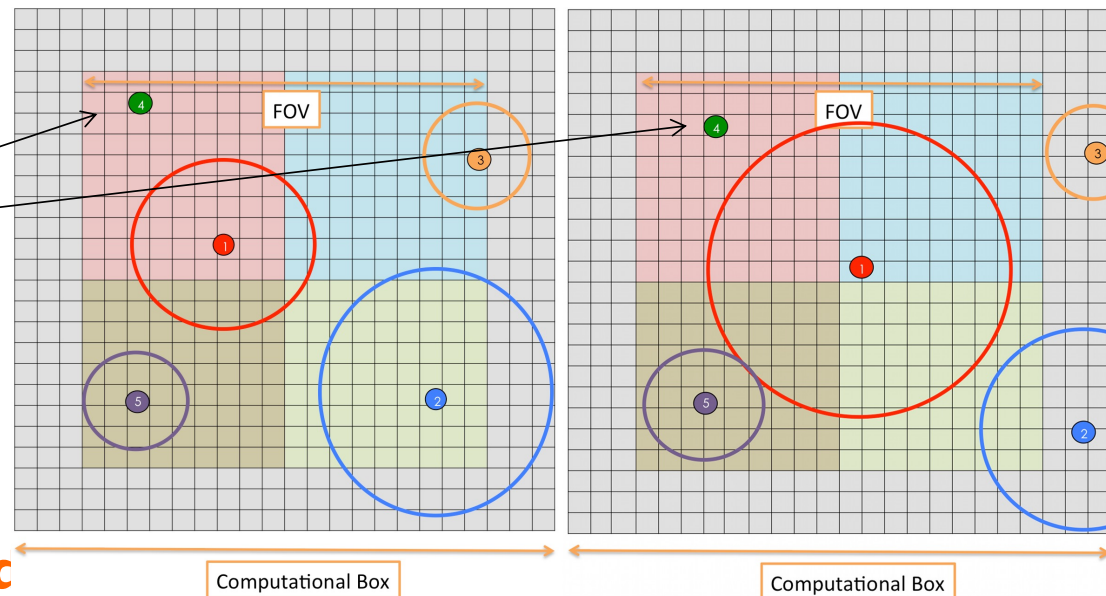
- Highly unbalanced
- Highly inefficient (each thread going through all particles – particles can have any radius)

BOTH SOLUTIONS LEAD TO POOR PERFORMANCE

GPU RENDERING

.Divide the image in rectangular tiles and classify particles depending on their size:

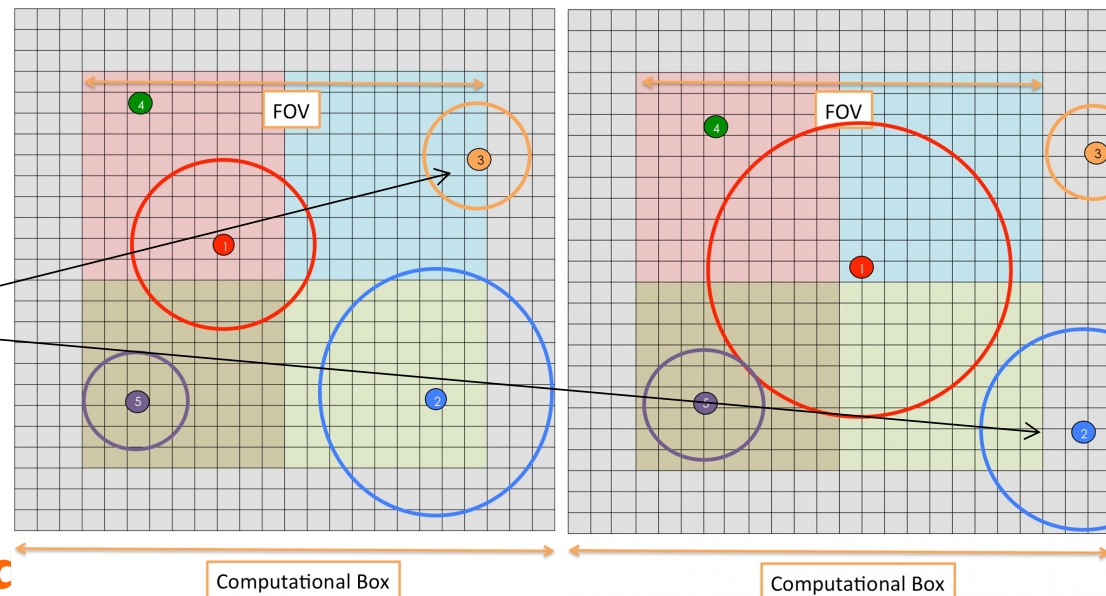
- **Small**
- **Medium**
- **Big**
- **Not processec**



GPU RENDERING

.Divide the image in rectangular tiles and classify particles depending on their size:

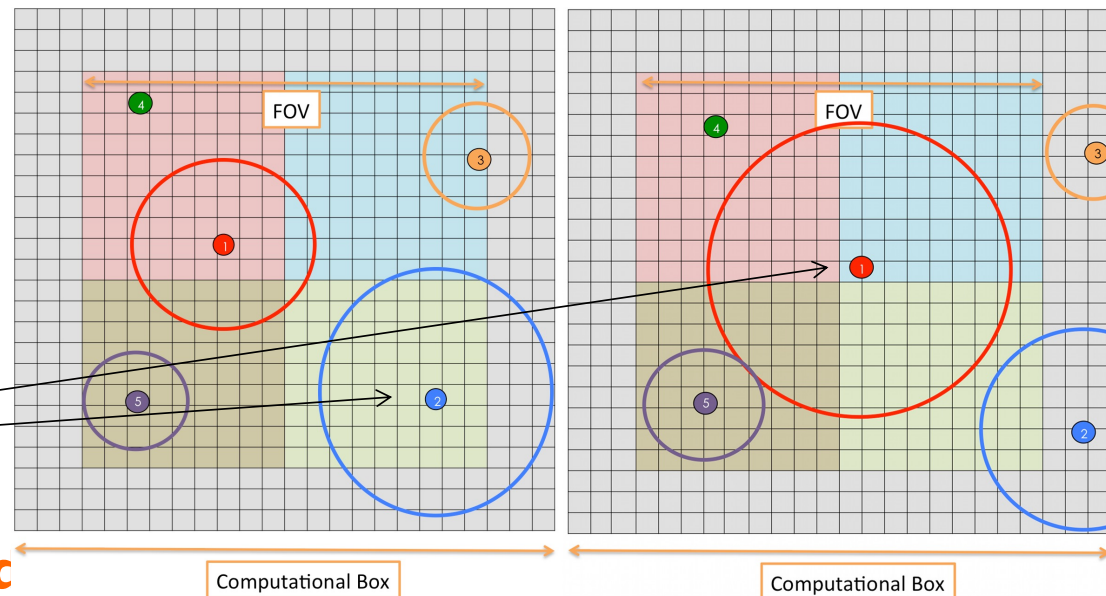
- **Small**
- **Medium**
- **Big**
- **Not processed**



GPU RENDERING

.Divide the image in rectangular tiles and classify particles depending on their size:

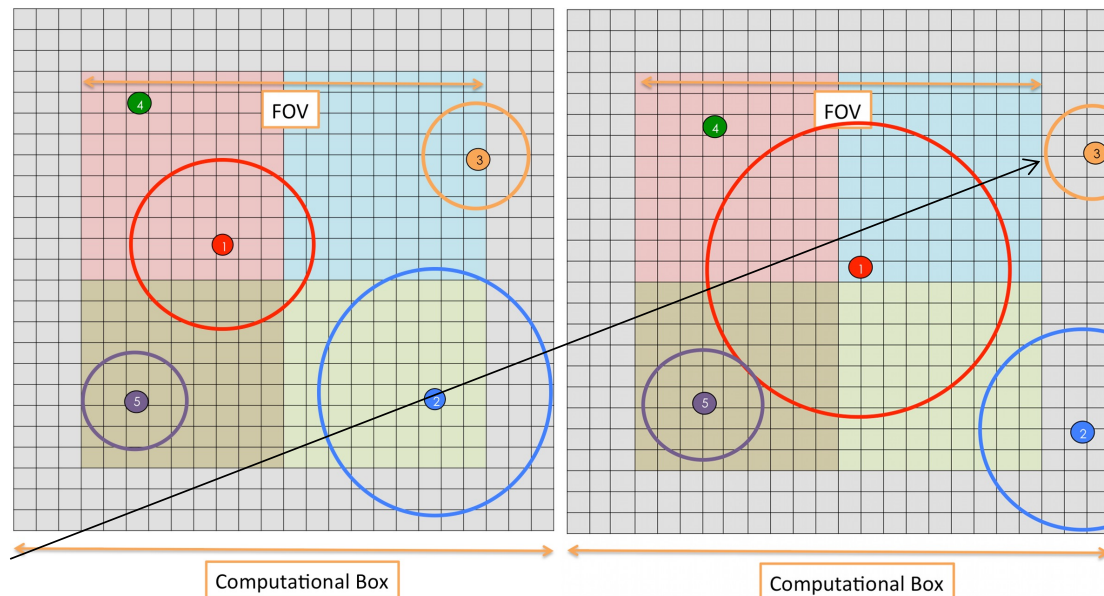
- **Small**
- **Medium**
- **Big**
- **Not processed**



GPU RENDERING

.Divide the image in rectangular tiles and classify particles depending on their size:

- **Small**
- **Medium**
- **Big**
- **Not processed**



GPU RENDERING

Small

- Affect a single pixel
- One thread per particle approach on the GPU

Medium

- Each particle affects only ONE tile (plus borders)
- Each tile is managed by a CUDA block
- One thread per pixel approach
- Tile size is set in order to exploit shared memory (for performance)

Big

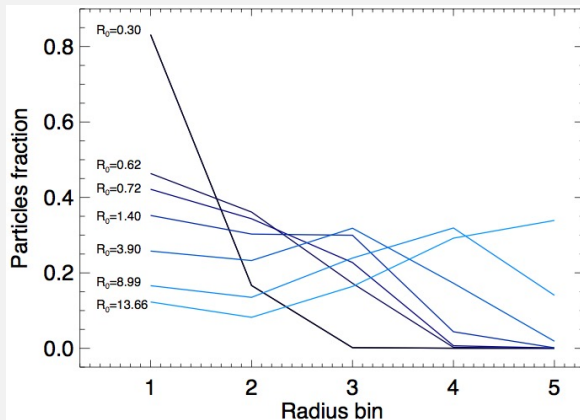
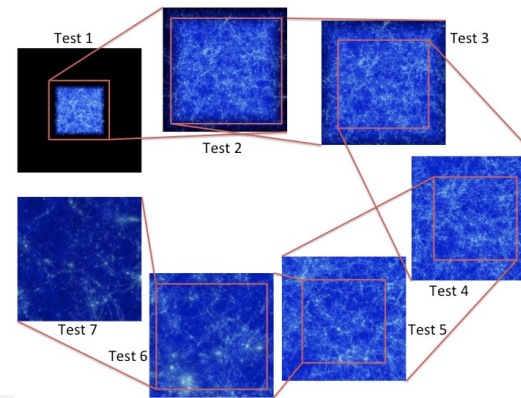
- Affecting more than one tile
- Processed on the CPU, synchronously to the GPU
- Data copy overlapped to GPU computation

CUDA TESTS

DATASET:

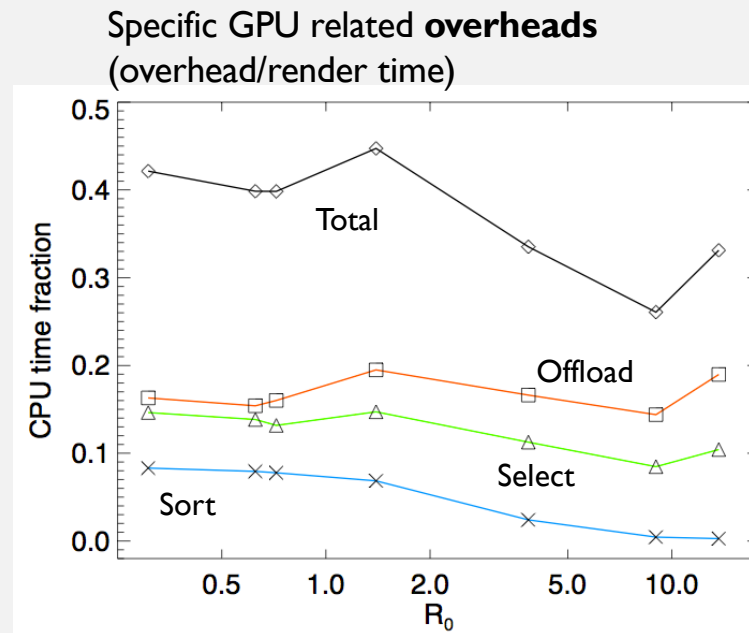
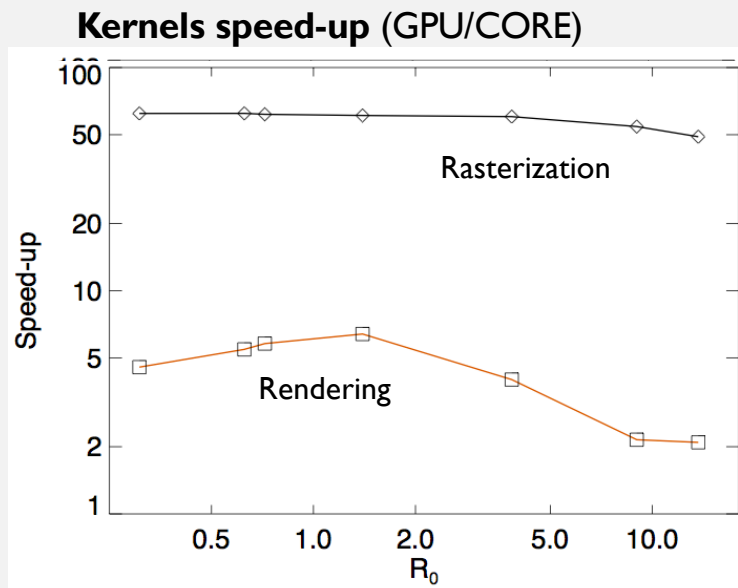
Medium sized cosmological N-body simulation performed with the Gadget code

- 200M particles of dark matter
- 200M particles of baryonic matter
- 10M star particles
- Approx 14 GB required



- Image resolution fixed: $N_{pix}=1000$
- 7 different camera positions closer and closer to the box center
- Increasing average size of the particles: R_0
- Impact on the performance

CUDA TESTS: GPU SPEEDUP AND OVERHEAD

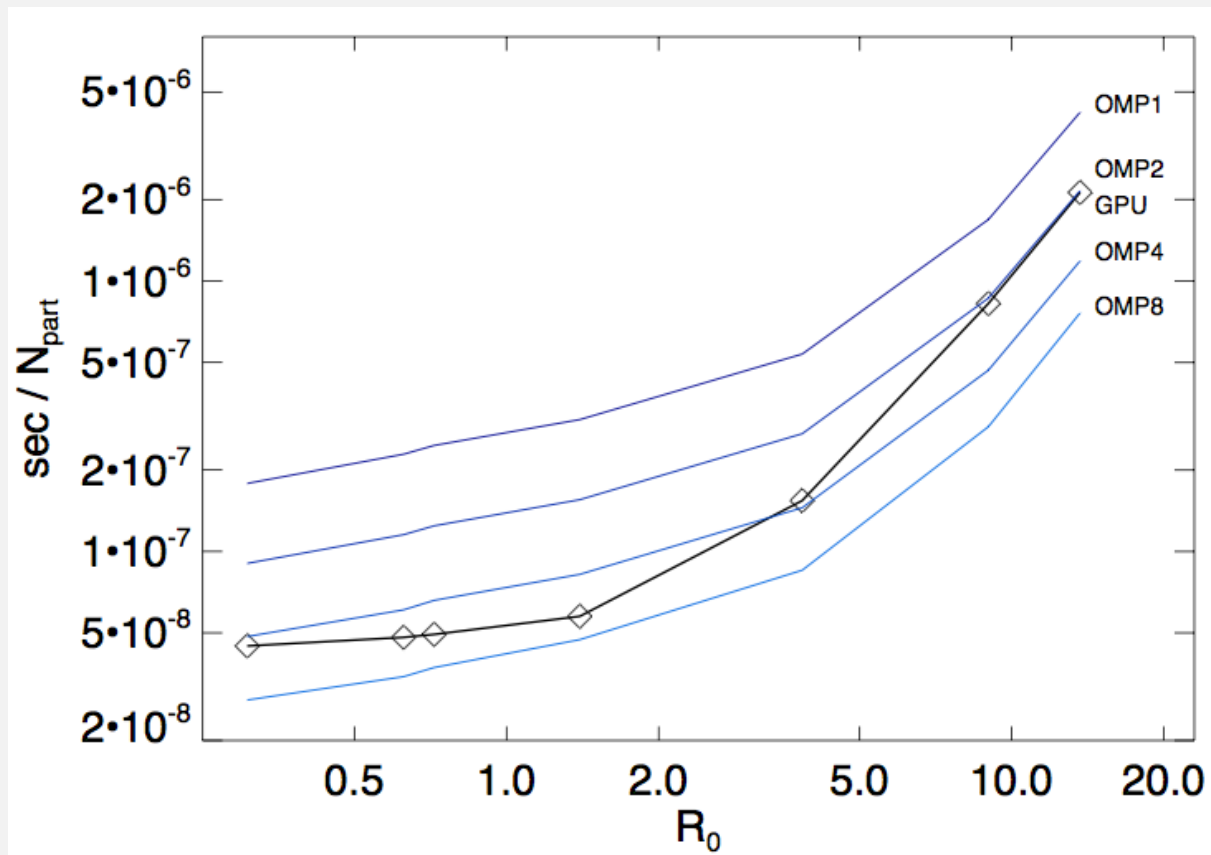


- **Rasterization** is strongly accelerated, speed-up is around **50**
- **Rendering** maximum speed-up is **7** for R_0 around unity
- **Overheads** can be up to **45%** of the total computing time

$$\text{Speed-up} = T_{\text{core}} / T_{\text{gpu}}$$

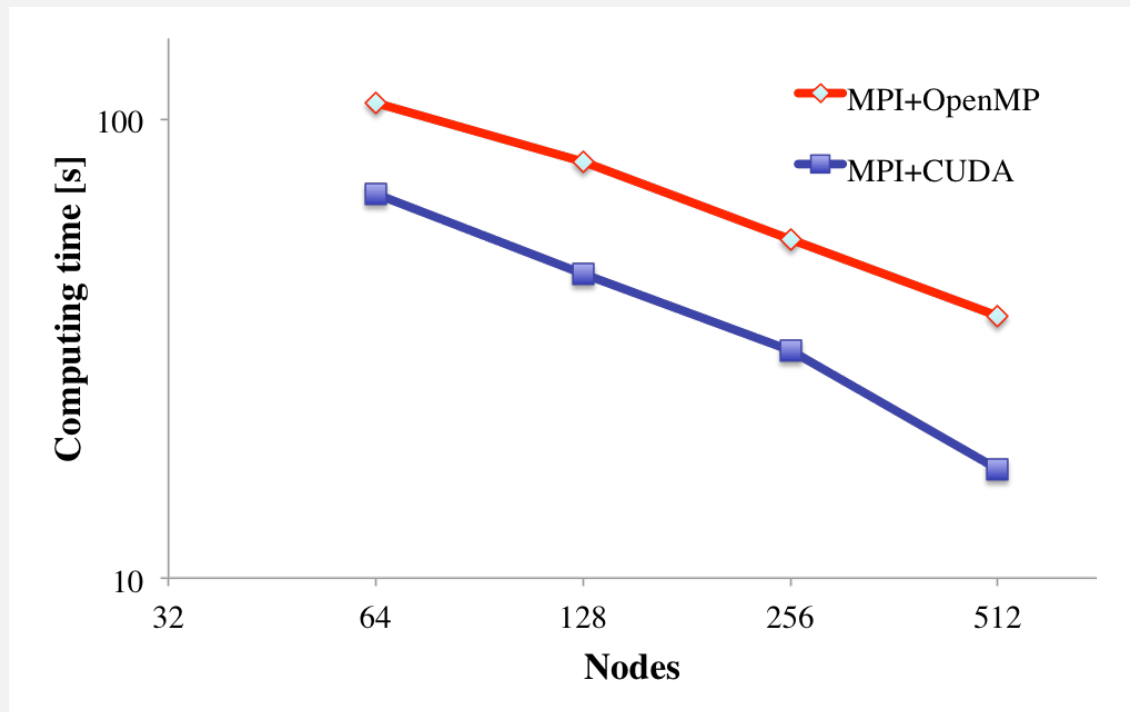
CUDA TESTS: OVERALL SPEED-UP

Overall computing times per particle vs average particle radius



SPLOTCH AT SCALE

The full HPC enabling allows to exploit any kind of HPC systems supporting (in principle) datasets of any size

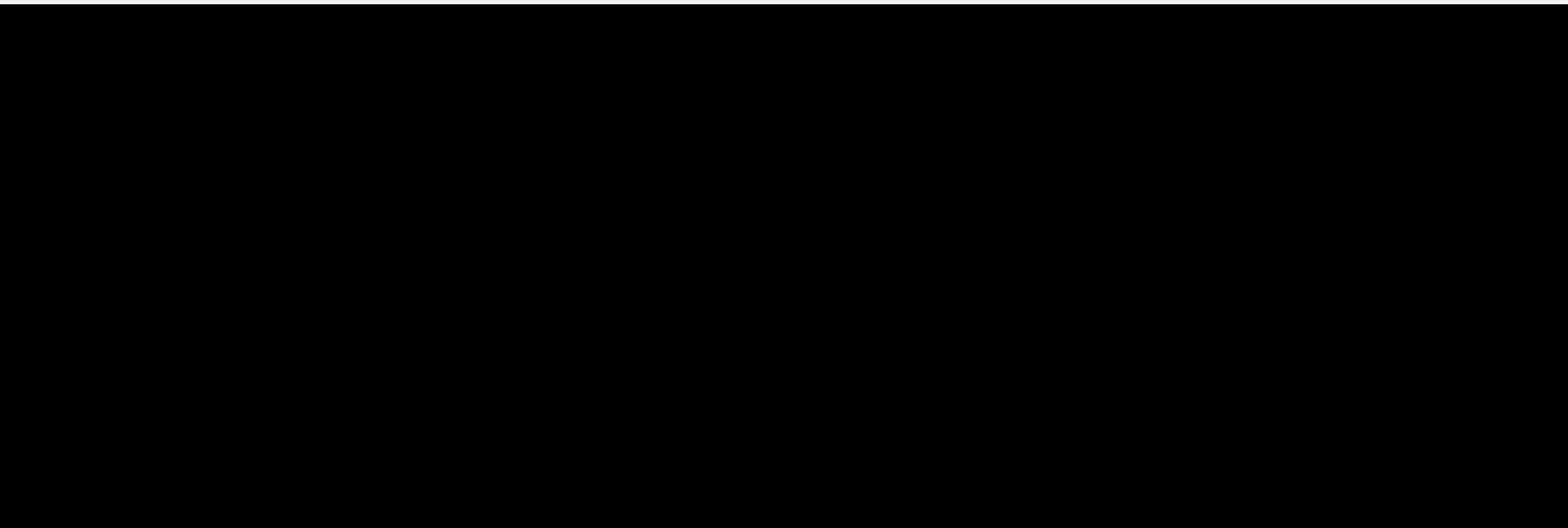


MAGNETICUM SIMULATION

- 187 billion particles – Gadget 3 code
- Dolag et al 2016



THE M83 GROUP



CONCLUSIONS

- **Visualization is an effective way to extract information from data**
- **Full HPC enabling is necessary to visualize modern datasets**
- **Splotch provides an effective solution for particle based datasets**
- **...what next: AI ?**