Big Data Visualization for Science: the Splotch approach

Claudio Gheller (ETH Zurich), Marzia Rivi (University of Oxford),
Mel Krokos, Tim Dykes, Ian Cant (Univ. of Portsmouth),
Klaus Dolag & Martin Reinecke (MPI Munich)

In this lecture we will discuss:

- **Why we bother about big scientific data**

In this lecture we will discuss:

- Why we bother about big scientific data
- How we can visualize data of any size using Splotch

316714	101001	42268	42268	2	2100.0000	1665	1668	2	15	39	144	293	1105	67	3
316714	201003	133373	133373	12	12100.0000	4005	4007	2	38	58	245	1142	3142	170	2
316714	301005	27005	27005	5	5100.0000	708	708	2	5	8	30	118	503	34	0
316714	401007	19004	0	4	0 .0000	0	0	0	0	0	0	0	0	0	0
316714	501009	46456	45426	5	3100.0000	964	965	0	6	7	62	194	547	40	1
316714	601011	11357	4250	2	1100.0000	29	29	1	0	3	2	14	6	3	0
316714	701013	21784	21784	4	4100.0000	835	836	2	3	13	114	153	523	27	1
316714	801015	117498	117498	7	7100.0000	7210	7234	13	54	207	633	1671	4188	444	24
316714	901017	36864	36864	4	4100.0000	1264	1269	3	14	28	148	232	781	58	5
316714	1001019	21923	3220	3	1100.0000	184	184	0	2	2	11	38	122	9	0
316714	1101021	37070	37070	5	5100.0000	732	732	3	2	16	127	167	376	41	0
316714	1201023	15982	13682	4	1100.0000	26	26	0	0	0	4	7	13	2	0
316714	1301025	28616	27027	5	4100.0000	367	367	0	3	7	15	89	234	19	0
316714	1401027	14027	11901	3	2100.0000	173	174	0	1	4	10	67	84	7	1
316714	1501029	14367	13873	3	2100.0000	217	217	0	1	7	4	40	141	24	0
316714	1601031	42610	42610	6	5100.0000	795	798	1	15	31	43	249	407	49	3
316714	1701033	52163	52163	7	7100.0000	1797	1802	2	11	15	69	285	1382	33	5
316714	1801035	14033	14033	2	2100.0000	145	145	0	0	2	17	45	79	2	0
316714	1901037	11706	11706	2	2100.0000	286	286	3	3	0	24	93	147	16	0
316714	2001039	37556	35596	4	3100.0000	1044	1045	2	6	12	166	167	668	23	1
316714	2101041	13692	13692	3	3100.0000	304	304	2	2	5	60	86	137	12	0
316714	2201043	75302	75302	3	3100.0000	1829	1846	0	19	12	70	425	1149	154	17
316714	2301045	49073	47973	10	8100.0000	1102	1102	2	10	25	141	174	671	79	0
316714	2401047	46960	46960	2	2100.0000	2989	3018	4	25	64	411	670	1642	173	29
316714	2501049	58694	16737	4	2100.0000	483	483	1	0	4	13	63	390	12	0

In this lecture we will discuss:

- Why we bother about big scientific data
- How we can visualize data of any size using Splotch
- **The "magic" of parallel processing**

Vs.

In this lecture we will discuss:

- Why we bother about big scientific data
- How we can visualize data of any size using Splotch
- The “magic” of parallel processing

And, finally……

- **We will learn how to use Splotch**

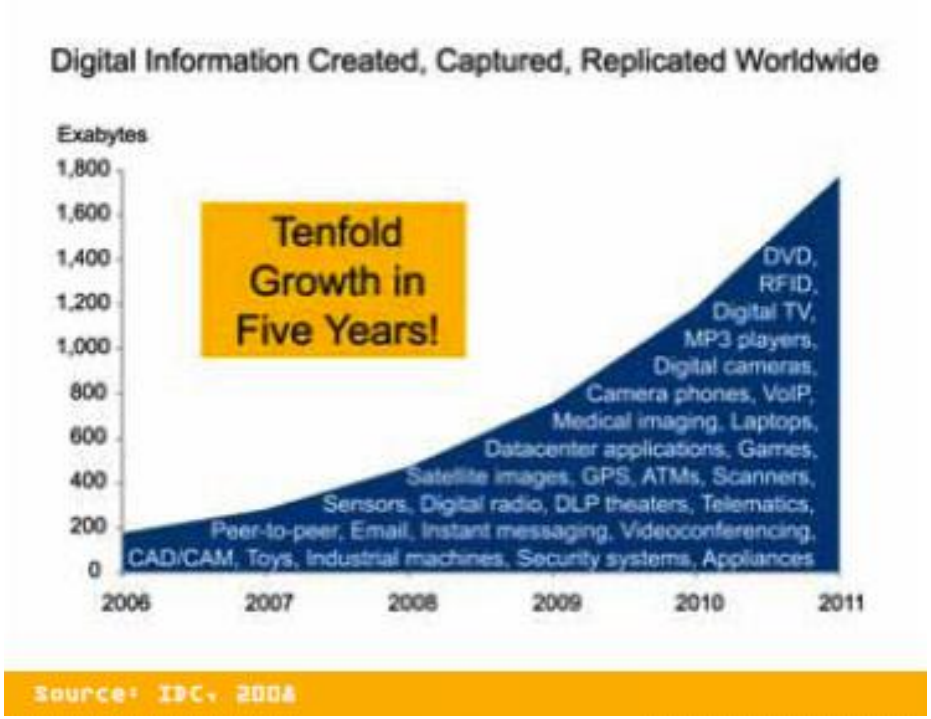
Terminal — bash — 102×25

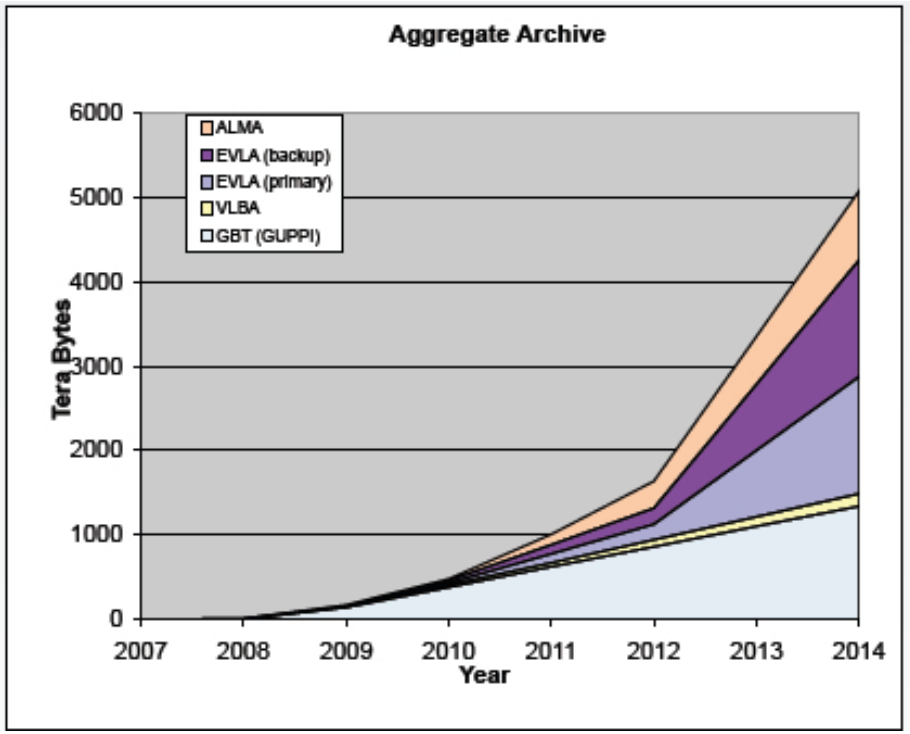
bash

```
Meride:splotch-6.0 cgheller$ ./Splotch5-mac demo.par

+---------+
| splotch |
+---------+

Vector math: SSE2
OpenMP active: max. 8 threads.
MPI: not supported by this binary

building color maps (2)...
 loading 3 entries of color table of ptype 0
 loading 4 entries of color table of ptype 1

reading data ...
Error encountered at reader/gadget_reader.cc, line 220
(function void gadget_reader(paramfile&, int, std::vector<particle_sim, std::allocator<particle_sim> >
&, std::vector<unsigned int, std::allocator<unsigned int> >&, std::vector<vec3_t<float>, std::allocato
r<vec3_t<float> > >&, int, double&, double&))

could not open input file! <test/snap1_092>

terminate called after throwing an instance of 'PlanckError'
Abort trap
Meride:splotch-6.0 cgheller$
```

The data tsunami

- Experiments, observations, numerical applications produce huge amount of data: the data volume approx. doubles each year, leading to the "Data Tsunami/Deluge/Avalanche…"
- This data must be properly stored and managed.
- Furthermore, the data must be accessible and proper tools for its exploration, processing and analysis should be available.



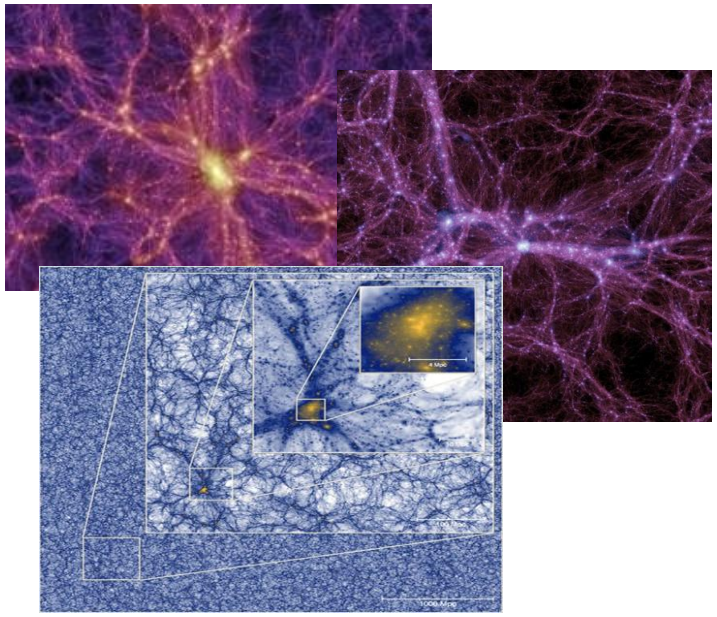


A couple of examples: Huge observational data - SKA

- The SQUARE KILOMETERS ARRAY (SKA) is one of the most ambitious projects in astronomy
- In terms of complexity (and costs) it is comparable to CERN LHC
- The instrument will generate an exabyte (10^{18}) of data every day (twice the information sent around the internet on a daily basis and 100 times more information than the LHC produces

A couple of examples: Huge simulations - Cosmological N-bodys

Pure gravity (N-body):	Size:
10^{11} particles	10^{11} elements
For each particles we save (at least) 3D position and velocity: 6 variables	$6x10^{11}$ elements
Each variable is a float number: 4 bytes	$2.4x10^{12}$ bytes = 2.4 TBs
We want to save a number of time steps: e.g. 10	24 TBs



Millennium I	(WMAP1-GAD)	500 /h Mpc	10 billion particles
Millennium II	(WMAP1 –GAD)	100/h Mpc	10 billion particle
Millenium XXL	(WMAP1-GAD)	3 /h Gpc	303 billion particles
Bolshoi	(WMAP7-ART)	250/h Mpc	8 billion particles
Multidark	(WMAP7-ART)	1 /h Gpc	8 billion particles
BigMD	(WMAP7-GAD)	2.5/h Gpc	56.6 billion particles
MICE	(WMAP5-GAD)	7 /h Gpc	8 billion particles
Horizon (FR)	(WMAP3-RAMSES)	2 /h Gpc	68 billion particles
Horizon (KR)	(WMAP5-GOTPM)	10.7 /h Gpc	372 billion.
DEUS (FR)	(WMAP7-RAMSES)	21/h Gpc	550 billion particles
JUBILEE	*(WMAP7-CP3M)*	*6/h Gpc*	*216 billion particles*

How do we squeeze all the information from data?

- **Various tools** offer powerful instrument for automatically analyzing large volumes of data, for classification, association, clustering, etc. (e.g. data mining or statistical tools)
- In general, data analysis is characterized by accurate and sophisticated algorithms that often:
  - scale as N^2 or even N^3 (non-linear behavior)
  - are complex and computationally expensive
  - cannot be optimized/parallelized (not suitable for HPC system)

But an extremely accurate approach is not always necessary…

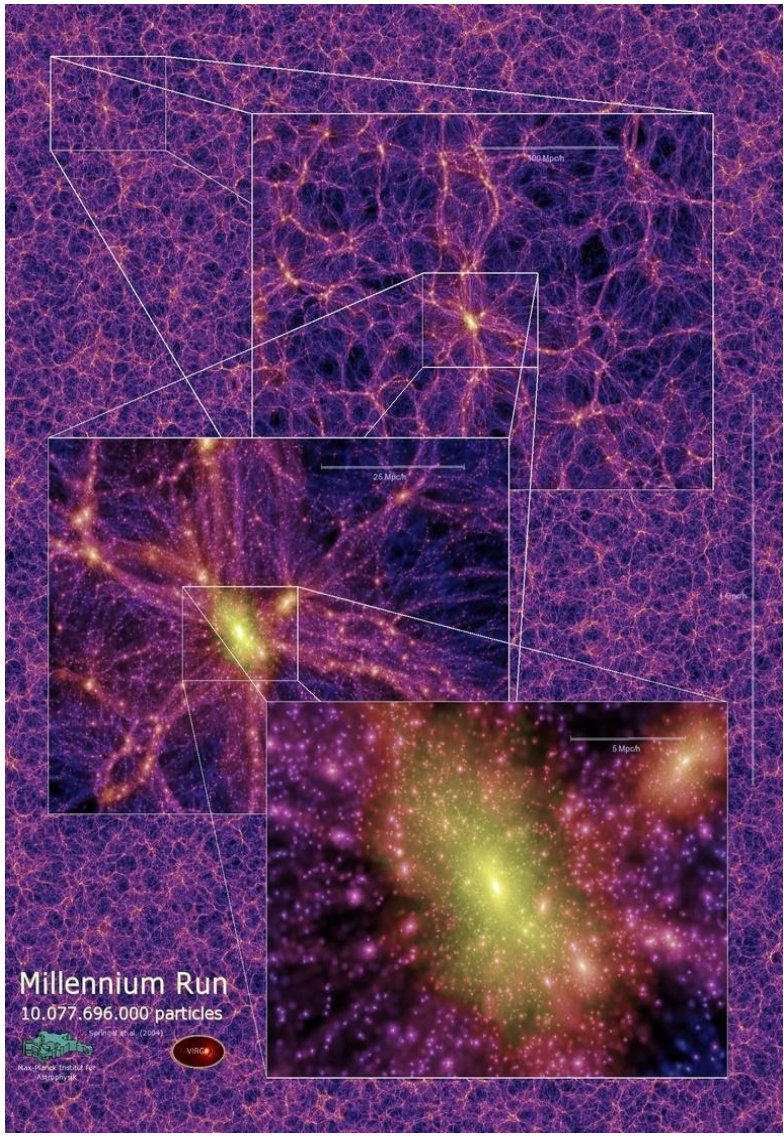
…visualization power…

some problems require a overall data exploration approach, as that provided by visualization…

Visualization offers an intuitive and immediate insight into data

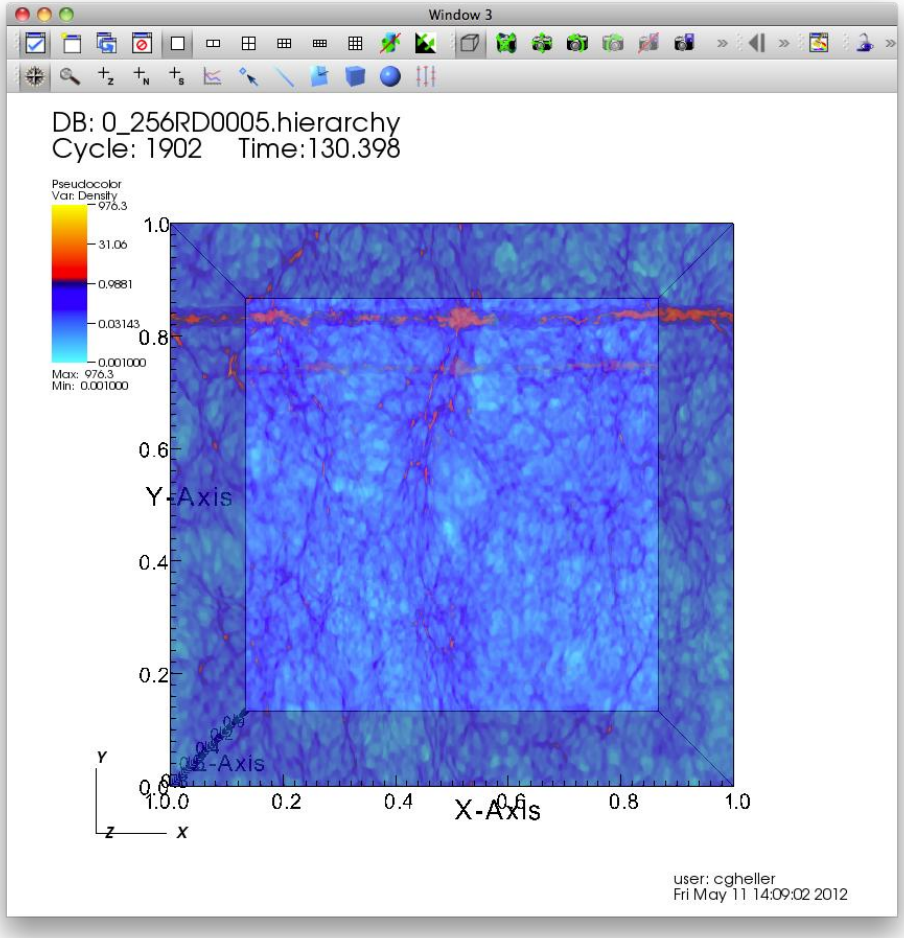
What takes hours for a CPU can take a glance for the human eye!!!

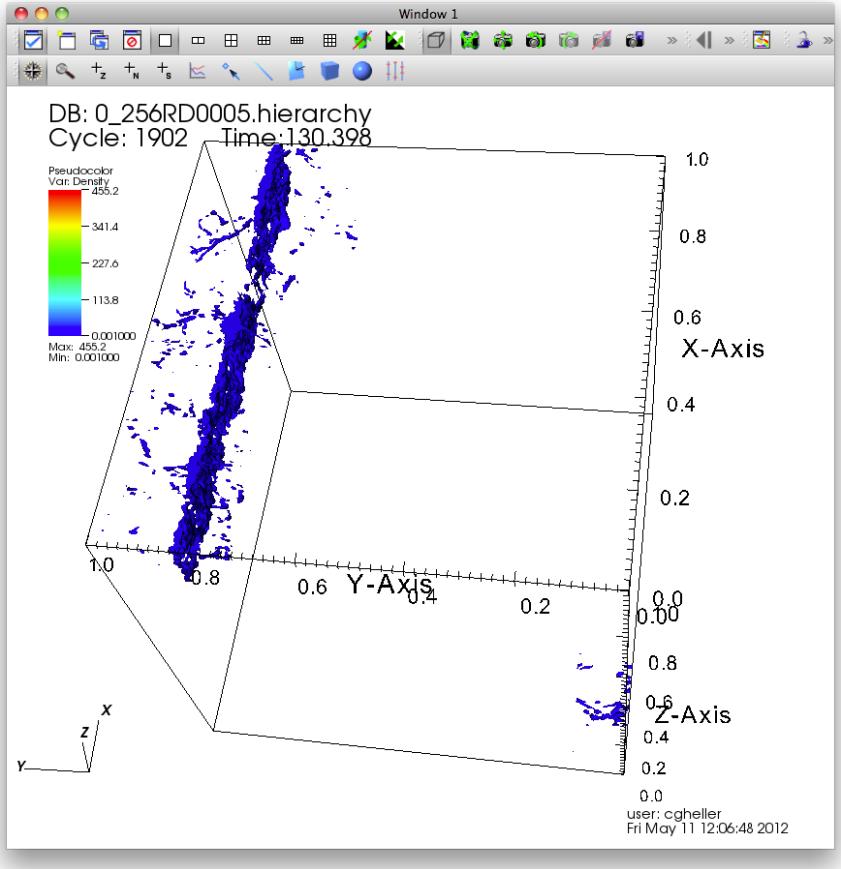
The visualization process plays a fundamental role in understanding data.



A couple of examples: data inspection

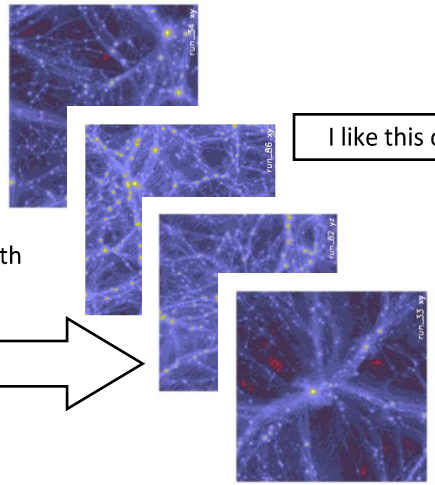
Problems in the data can clearly emerge looking at it…





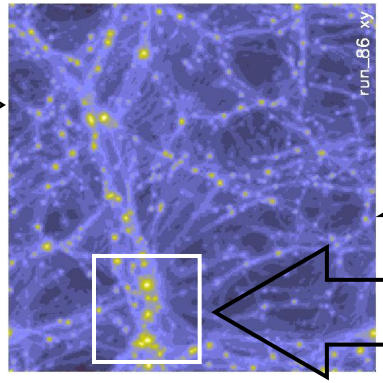
A couple of examples: find interesting objects/features

Search for simulations with Lambda>0.7



I like this one

It's too large !!!

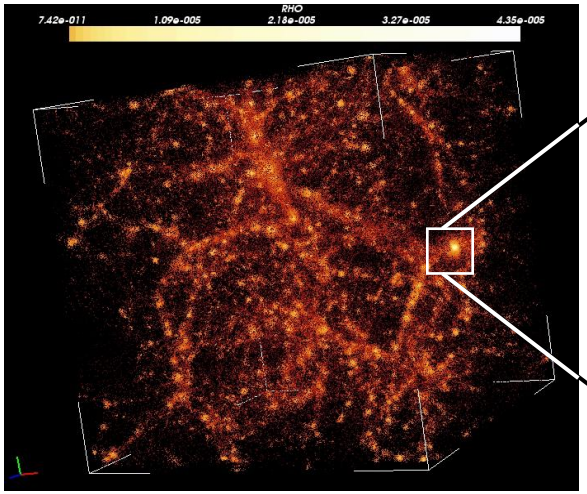


Metadata

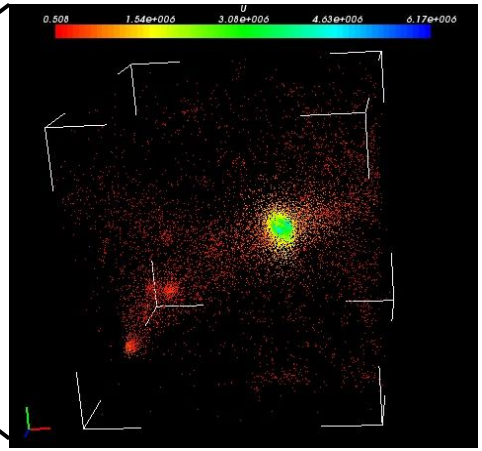
Binary data file

Let's select a sub-region!!!

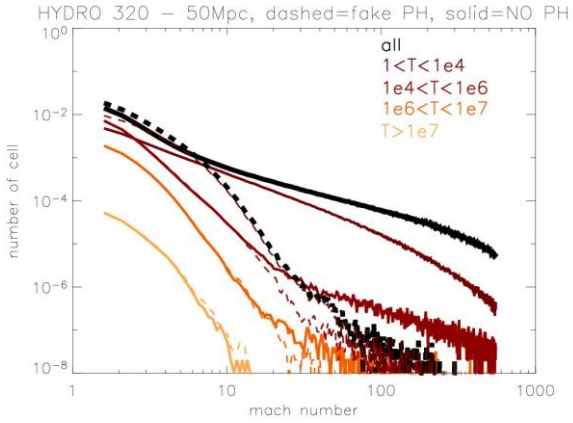
Data is too large!!!



Extract a sub region… it is still large



Perform the analysis on-site
Finally I have a jpeg… cannot be too large!!!



Visualization tools

Many products are available for graphics and visualization, even free/open source. However, not so many are designed for **SCIENTIFIC visualization**.

We need tools that:

- Are **suitable for scientific data** (e.g. sophisticated modeling is seldom necessary, efficient volume rendering or iso-volume/surface technique are much more effective)
- can **handle terabytes of data** in reasonable times (this rules out ANY PC, 32 bit software and promotes high performance computers, multicore, multinode, multiGPU)

Other, relevant features are:

- Be user-friendly possibly with simple but effective GUIs
- Support client-server approach, to avoid data movement
- Produce nice and accurate images

What's special with Splotch

- Our main focus is on **huge datasets** (giga to tera bytes – and beyond…) that cannot be visualized interactively and/or using "standard" visualization facilities.
- We want to use **HPC resources**, exploiting hybrid architectures that allow to load and process data in an acceptable time.
- Sort of "brute force approach", specialized on **POINT LIKE** data sources (like particle distributions…)

Splotch web site:

http://www.mpa-garching.mpg.de/~kdolag/Splotch

https://sites.google.com/site/claudiogheller/visualization

Splotch papers:

K. Dolag, M. Reinecke, C. Gheller, S. Imboden, "Splotch: Visualizing Cosmological Simulations", *New Journal of Physics*, 10(12) (2008)

Z. Jin, M. Krokos, M. Rivi, C. Gheller, K. Dolag, M. Reinecke, "High-Performance Astrophysical Visualization using Splotch", *Procedia Computer Science* 1 (2010) pp. 1775-1784

M. Rivi, C. Gheller, M. Krokos, "GPU Accelerated Particle Visualization with Splotch", submitted to Journal of Parallel and Distributed Computing

Point like data

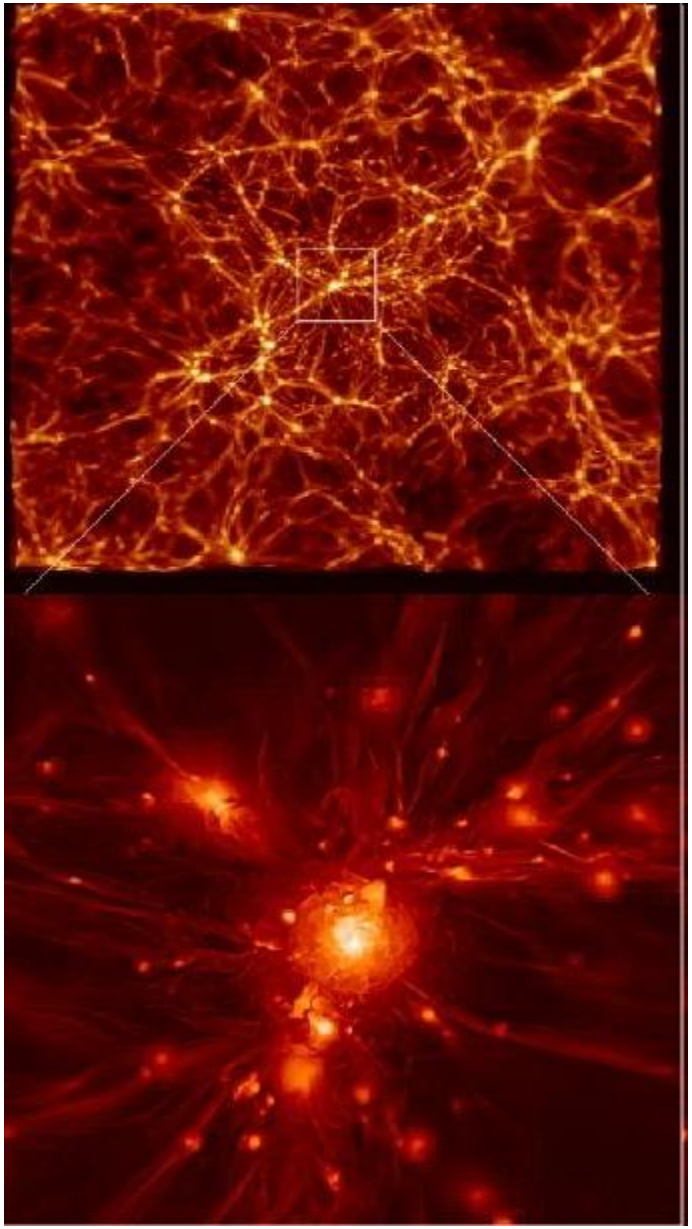
- An important class of astrophysical (more generally – scientific)**simulations use a point like (particles) discretization** of the evolving fluid (N-body and SPH simulations)
- In astrophysics This numerical approach is used in simulations of the large scale structure of the universe, evolution of galaxy clusters, galactic dynamics, star clusters, formation of planetary systems
- Each particle has a specific **position and velocity and various properties** (mass, temperature, pressure, entropy, metallicty…..)
- Particles dynamics is solved with different approaches, all dominated by **gravitational forces**
- **Mesh based data** (in particular adaptive mesh data) can be treated as special (regularly distributed in space) particles

Splotch basics

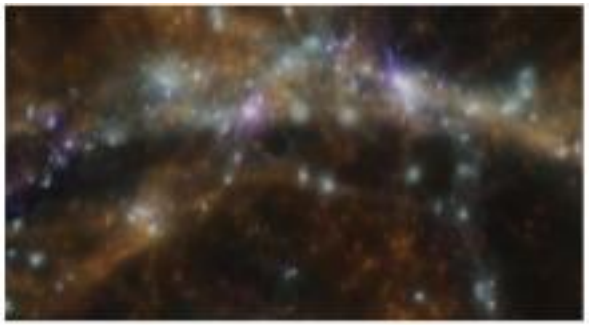
Splotch is a **ray-casting algorithm** for effective visualization of large-scale point-like datasets, based on solution of the **radiative transport equation**

$$\frac{d\mathbf{I}(\mathbf{x})}{dx} = (\mathbf{E_P} - \mathbf{A_P}\mathbf{I}(\mathbf{x}))\rho_\mathbf{P}(\mathbf{x})$$

where the density is calculated smoothing the particle quantity on a "proper" neighborhood, by a Gaussian distribution function.

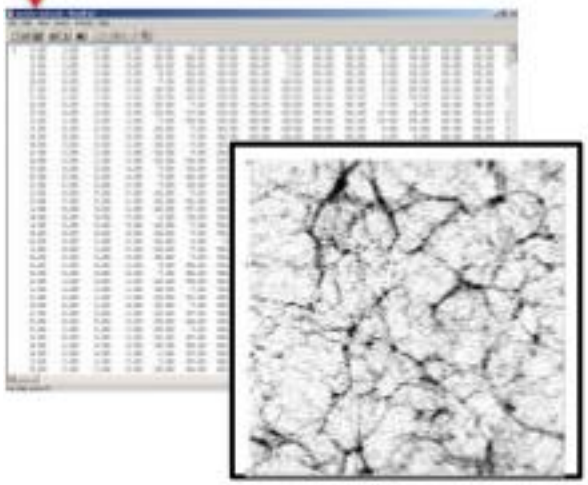


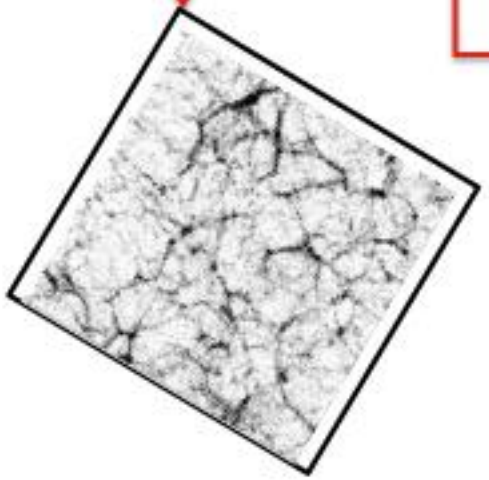
How does Splotch work?

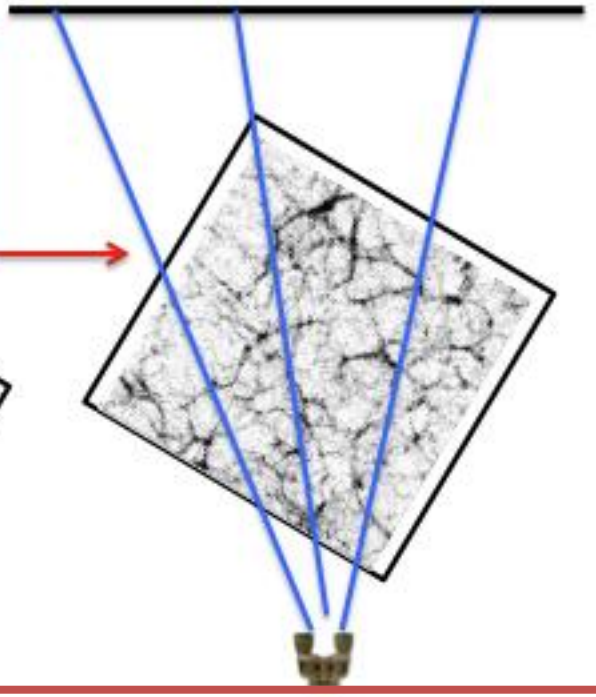


Main steps:

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



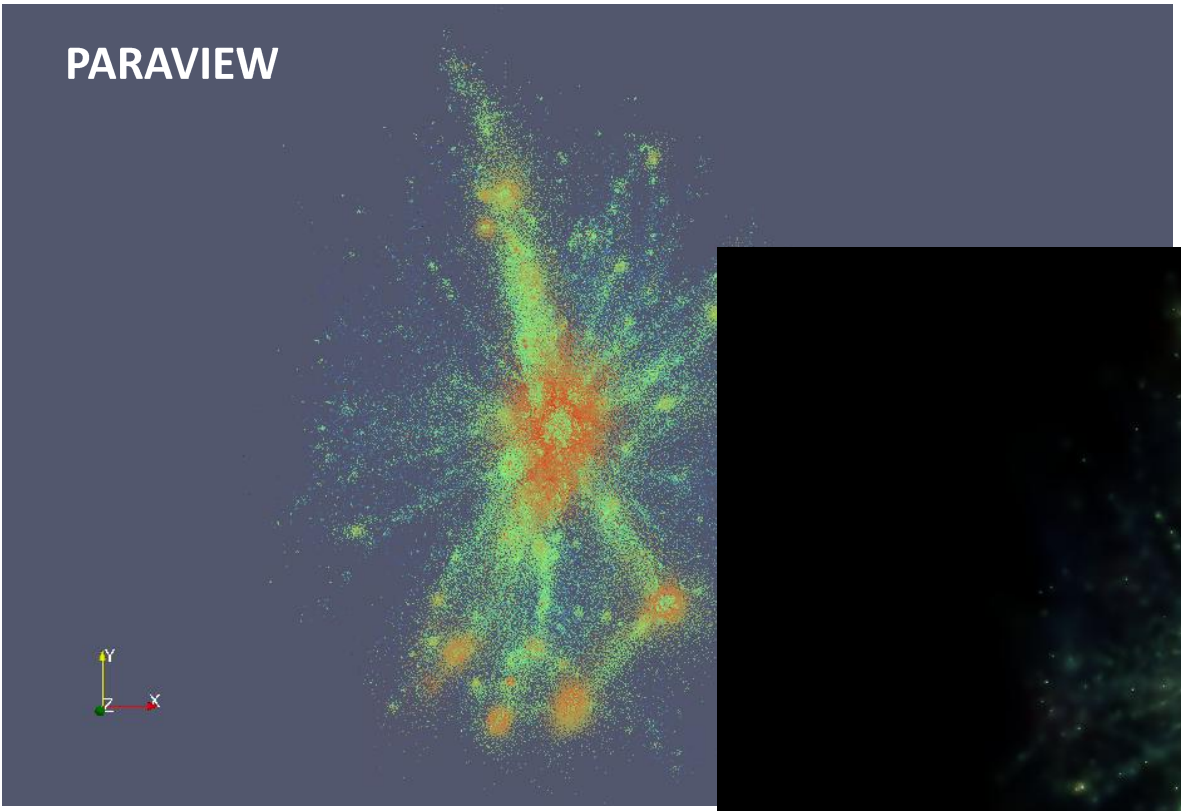


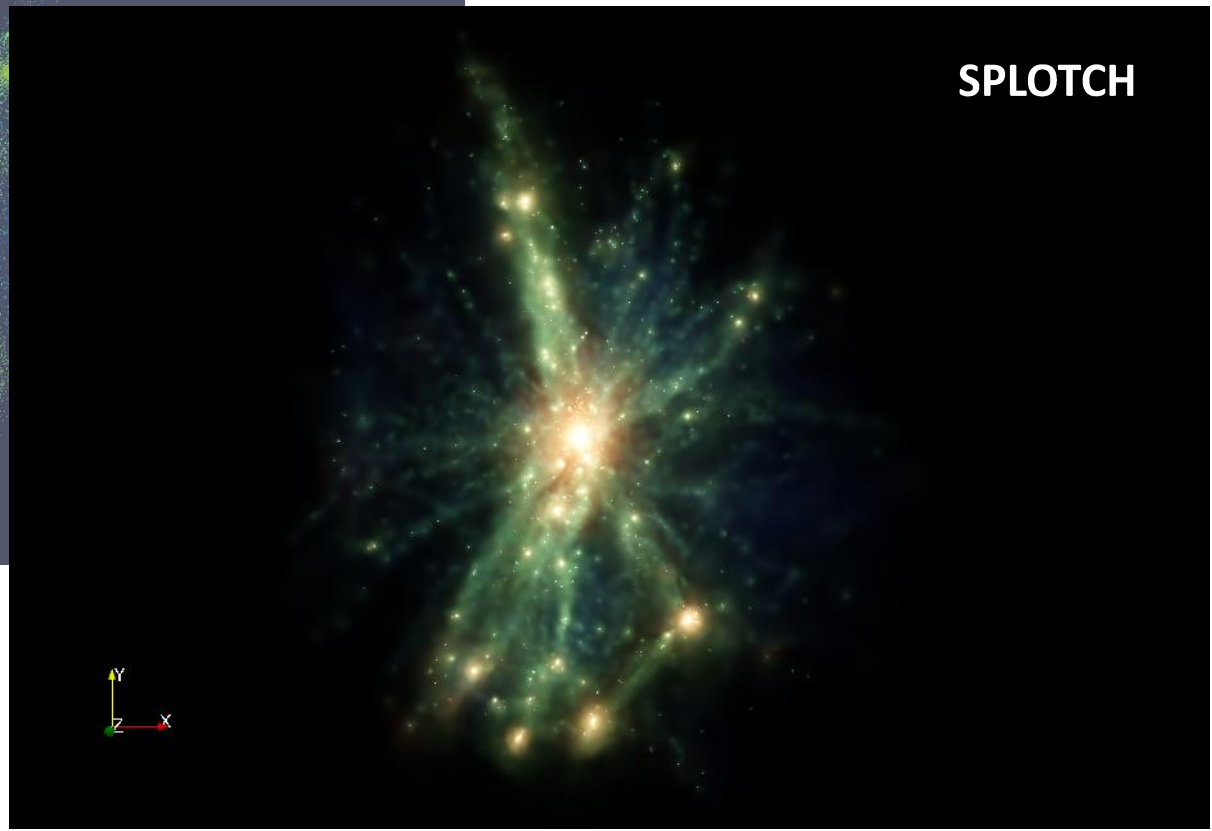


Designed for huge data: most of HPC architectures supported;
Simple: completely self contained, no external dependencies
Completely Open Source

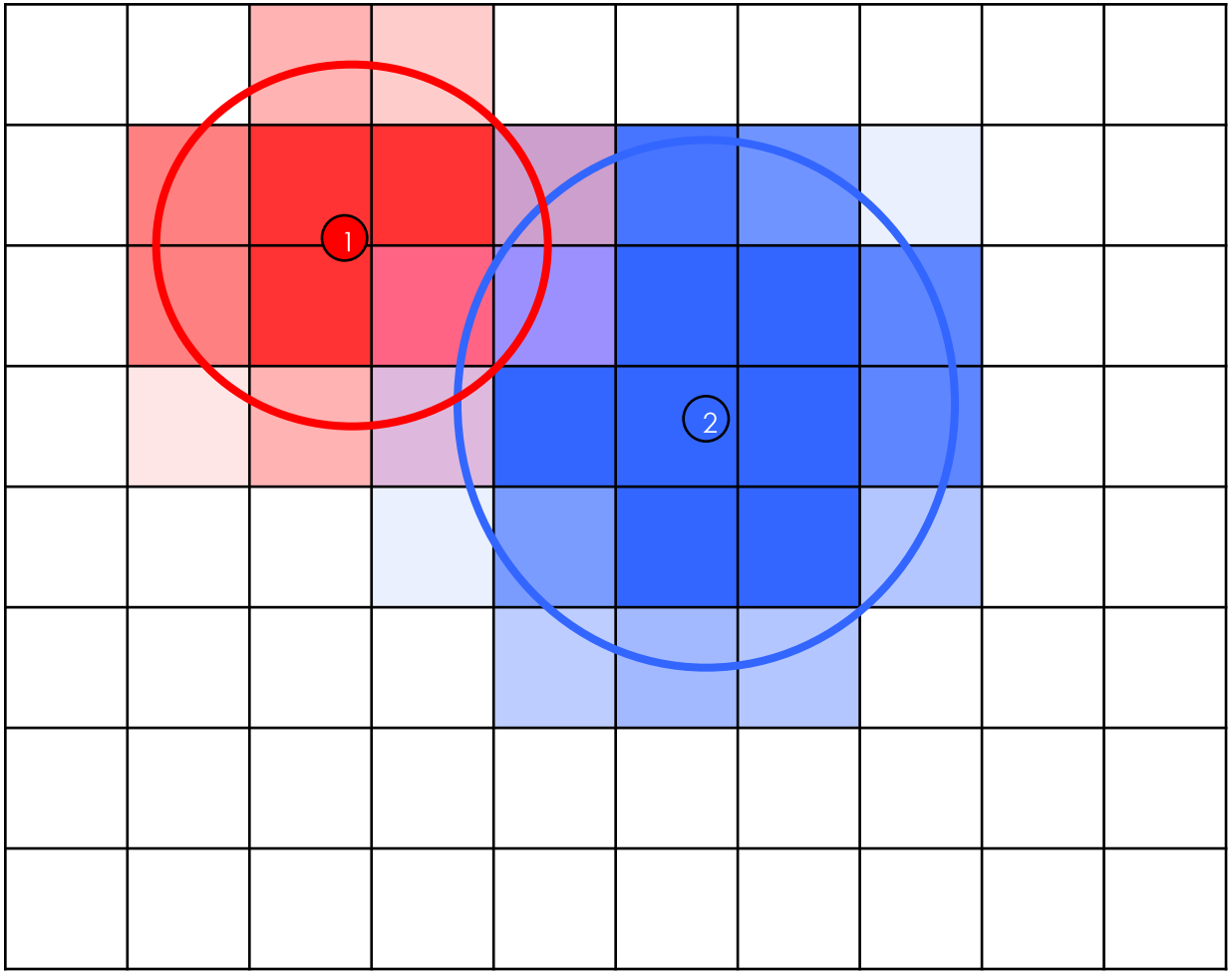
Particle visualization

Splotch was born for an effective visualization of cosmological N-body simulations. It is specialized o particle visualization…





Notable challenges 1: Smoothing particles

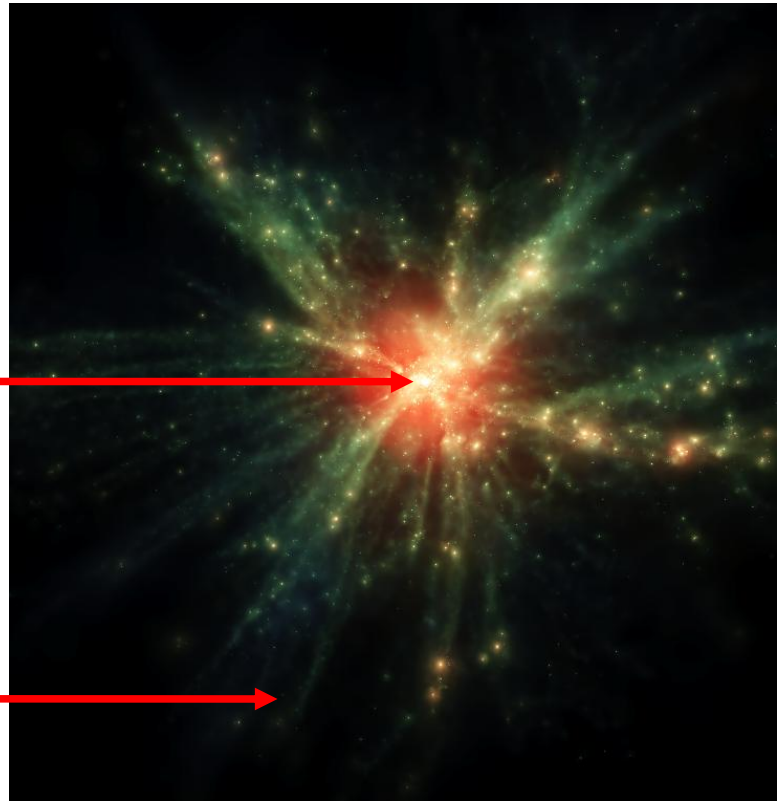


Note that, differently from traditional ray casting algorithms, splotch cannot easily predict how much data will affect a pixel…

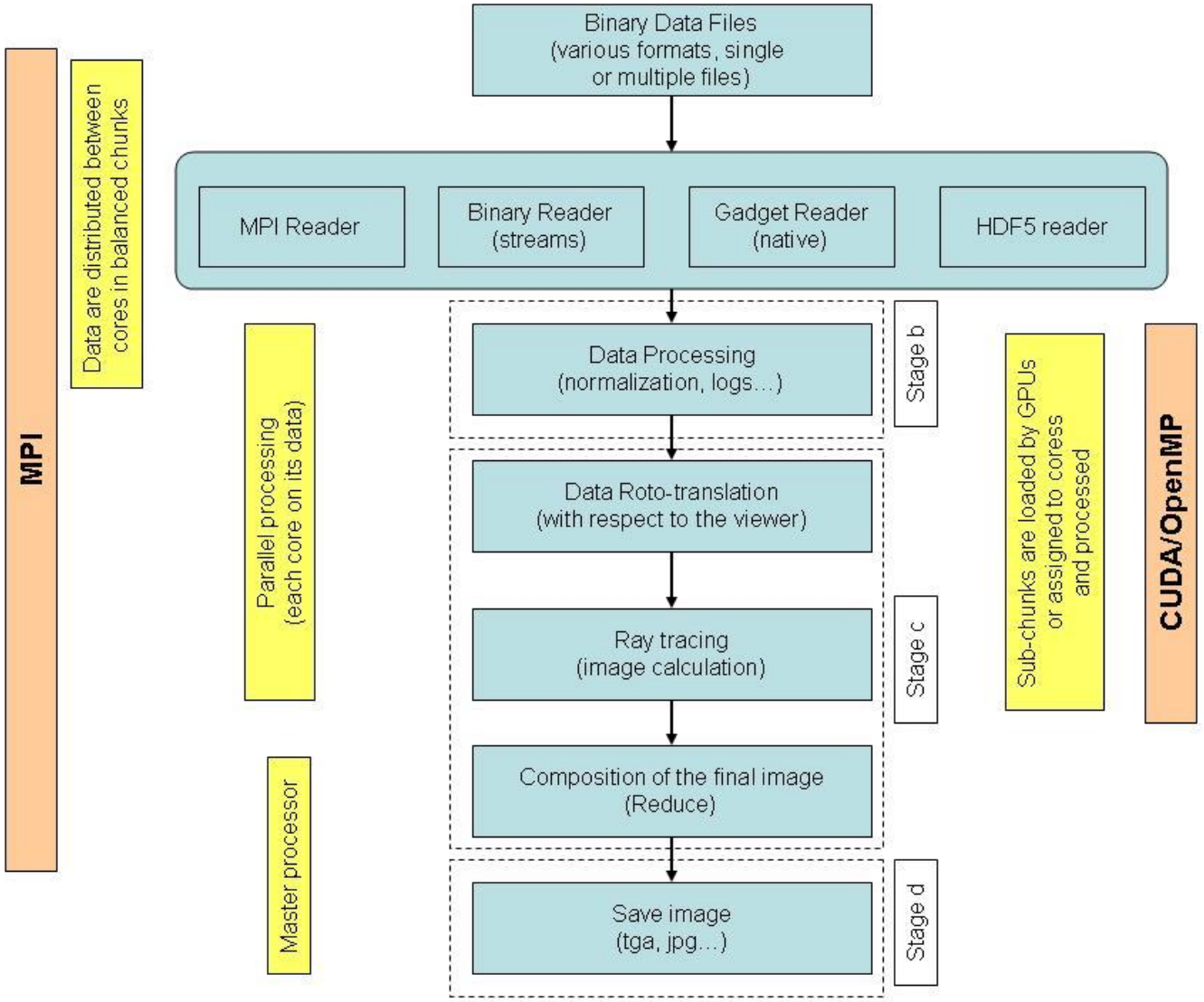
Notable challenges 2: Load balancing

High concentration of particles in small area

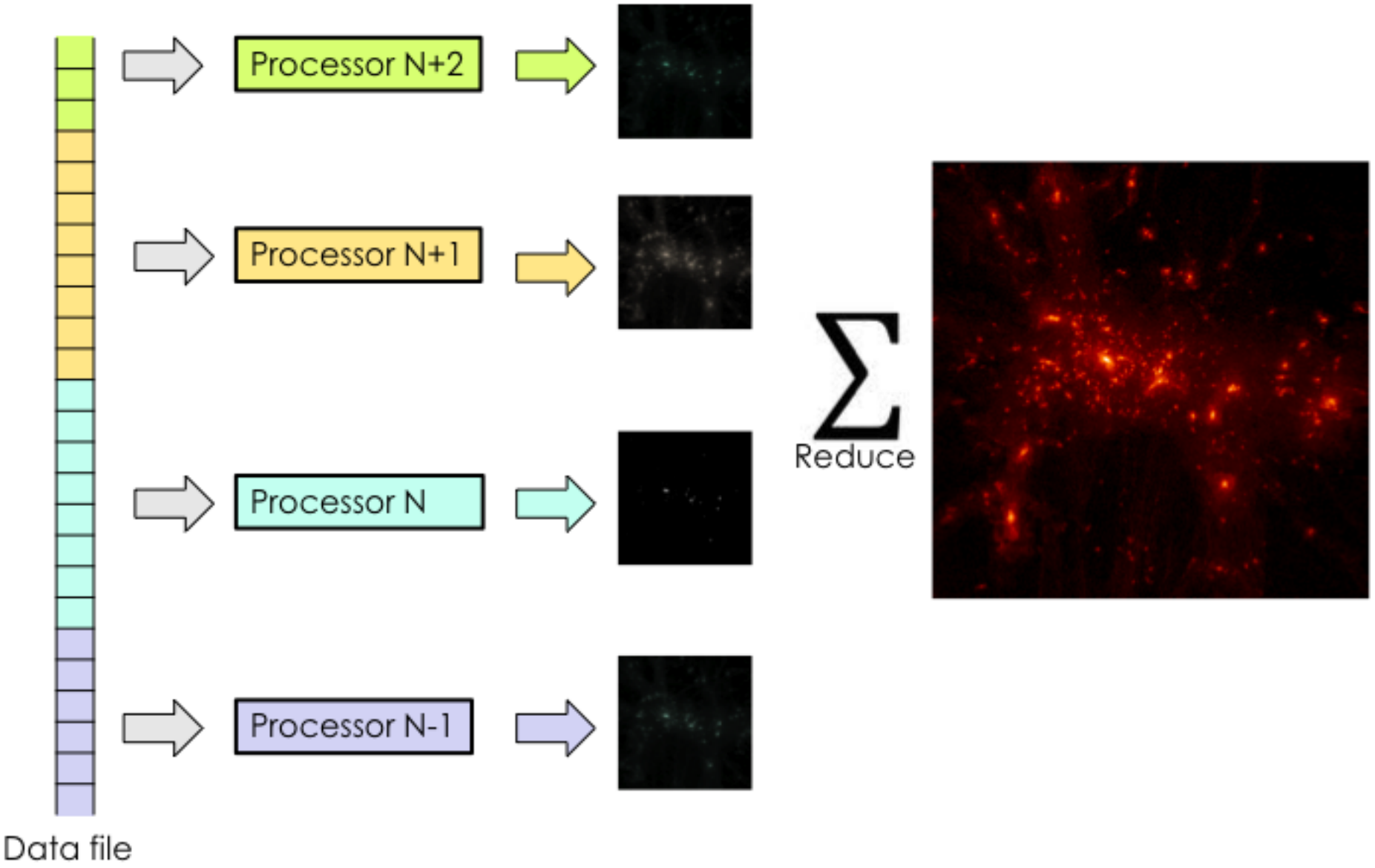
Low concentration of particles spread across large area



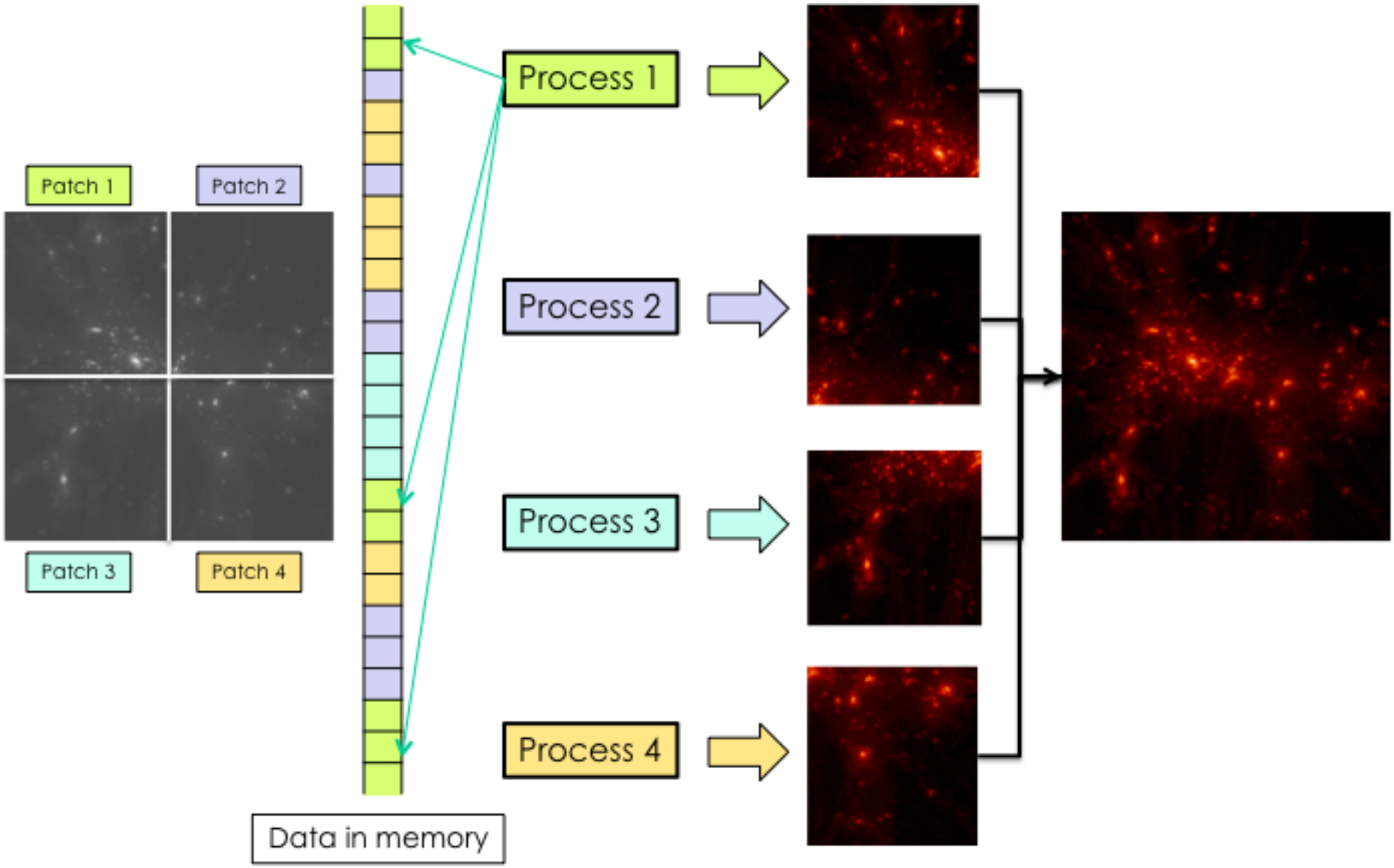
Splotch workflow



Splotch in parallel: 1, MPI implementation for distributed memory systems

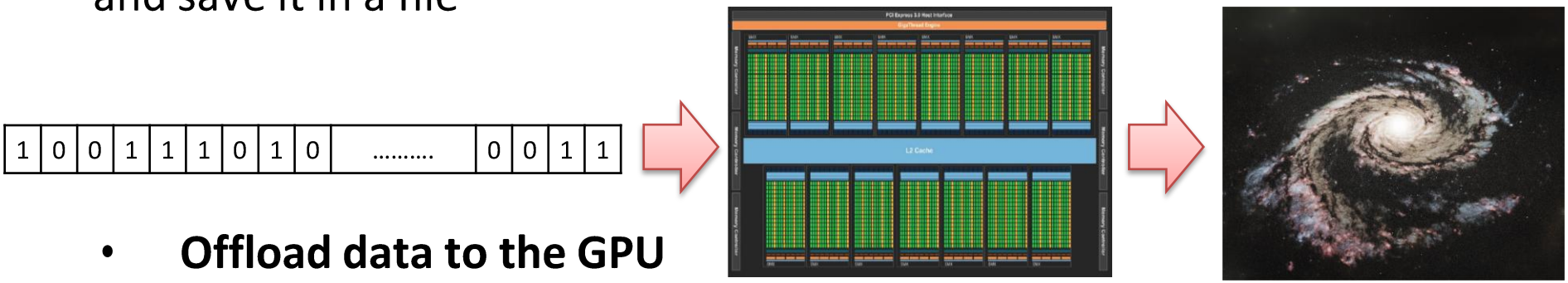


Splotch in parallel: 2, OpenMP implementation for shared memory systems



Splotch in parallel: 3, CUDA GPU implementation

The **CUDA based approach** consists in moving data on the GPU and let the GPU make the integration. Copy back the final image to the CPU and save it in a file



- **Offload data to the GPU**
- **Rasterisation: efficient *one-thread-per-particle* approach**
- **Rendering: a full refactoring of the algorithm was necessary to solve the following issues:**
  - **unbalanced distribution of particles**
  - **each particle access "randomly" the image, therefore possible concurrent access to the same pixel may occur**
- **Copy back the final image to the CPU**

Running Splotch

Currently splotch is a **command line application**. Therefore to run it:

> Splotch.exe parameter_file.par

> mpiexec –n 1024 Splotch.exe parameter_file.par

The **parameter file** is contains ALL the settings needed by splotch

It contains different classes of parameters:

- Data file related parameters (filename, file type, possibly file data structure)
- Variables related parameters (brightness, smoothing length, data type…)
- Scene related parameter (camera position(s), look-at direction(s), orientation(s)…)
- Image related parameters (filename, resolution…)

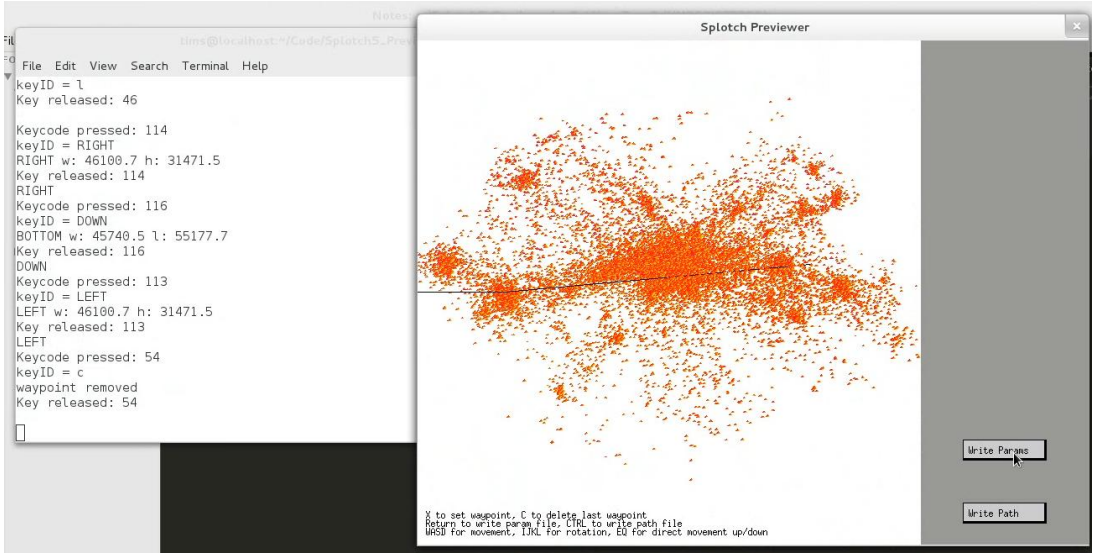
Splotch's Pluses

- **Completely self contained**
  - No dependencies
  - Easy to compile
- **Standard C++ based; portable on any platform**
  - Easily extensible
  - Usable anywhere
  - Scriptable
- **Can exploit (almost) any HPC system**
  - Small time to solution
  - And…
- **Data of ANY SIZE can be processed**
- **Support for animations**

Splotch's Drawbacks

1. Documentation is poor
2. Some practice is required to get a working parameter file
3. Trial and error approach (we have the big problem of the "black image")
4. Camera paths for animations have to be built by hand
5. No GUI/preview and real-time interaction… BUT

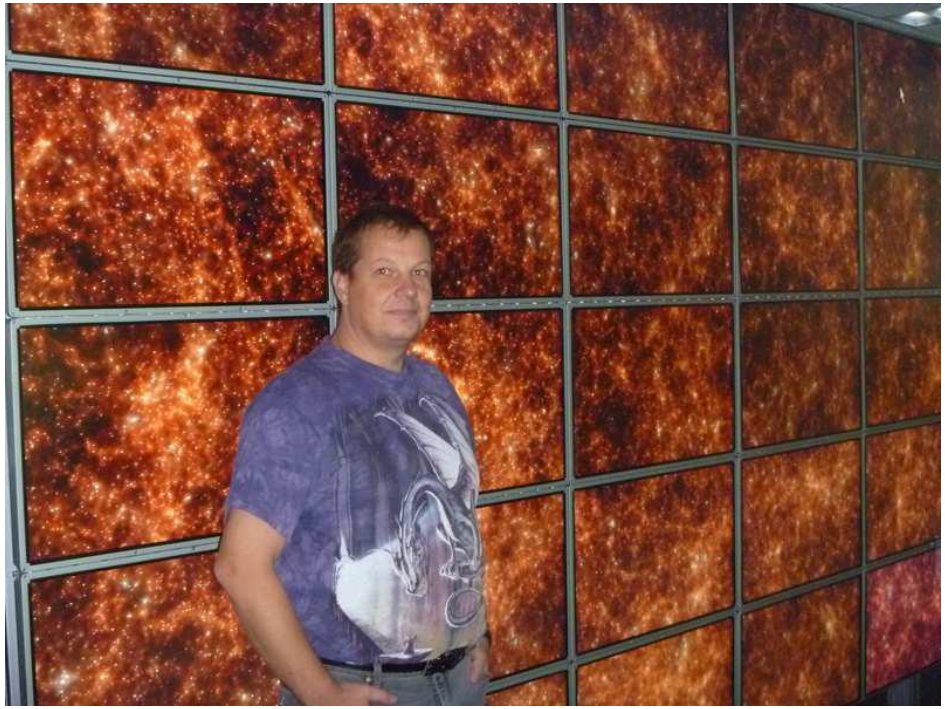
The Splotch Previewer is now available!!!



The Splotch team and on-going activities

- Martin Reineke (MPI Munich): the creator, general maintenance, OpenMP implementation
- Klaus Dolag (MPI Munich): Large scale cosmological simulations, Gadget code support
- Marzia Rivi (Oxford): GPU implementation (CUDA and OpenCL)
- Mel Krokos (Portsmouth): OpenCL implementation and Previewer
- Tim Dykes (Portsmouth): MIC implementation, Previewer, RAMSES reader
- Ian Cant (Portsmouth): Previewer
- Claudio Gheller (ETH-CSCS):

MPI implementation, various readers, GPU and MIC software design



Wrapping-up…

- Data of any size? You can visualize it, and in a reasonable time, with Splotch
- Find difficult to learn how to use Splotch? You can use the brand new previewer
- Do you like Splotch and would like to contribute? Contact one of us!
- **Interested to collaborate with an Internship at CSCS? Have a look at:**

http://www.cscs.ch/working_at_cscs/internships/index.html

BEFORE MOVING ON, SEVERAL EXAMPLES OF SPLOTCH DATA VISUALIZATION

…and now…

Hands-on session

Getting the software

The Splotch package is installed on all the PCs

or (if you wish to try on your laptop – LINUX/UNIX/MAC – limited Windows support)

Download the tarball from:
ftp://ftp.cscs.ch/out/cgheller/splotch/

Basic steps

1. (if downloaded) untar the package (tar xvf splotch-6.0.tar) anywhere on your laptop
2. Edit the Makefile and customize it
3. Compile (make)
4. Create the first image (data sub-directory): ./Splotch5-generic snap.par
5. Look at the image (display demo0000.tga / open demo0000.tga)
6. Let's have a look at the parameter file
7. Analysis of some parameters
8. Try to
    1. Change colors
    2. Change brightness
    3. Change smoothing length
9. Let's have a look at the second data file…….

Animations

The path file:

camera_x	camera_y	camera_z	lookat_x	lookat_y	lookat_z	fidx	brightness0	brightness1	brightness2	brightness3
0.00000	0.00000	-19945.0	0.00000	0.00000	0.00000	0.00000	0.00500000	0.997500	0.005	0.005
0.00000	0.00000	-19890.0	0.00000	0.00000	0.00000	0.00000	0.0100000	0.995000	0.01	0.01
0.00000	0.00000	-19835.0	0.00000	0.00000	0.00000	0.00000	0.0150000	0.992500	0.015	0.015

1. Write your path file according to the coordinates you can get from the .par file
2. Change the parameter file to use the path file
3. Create your small animation