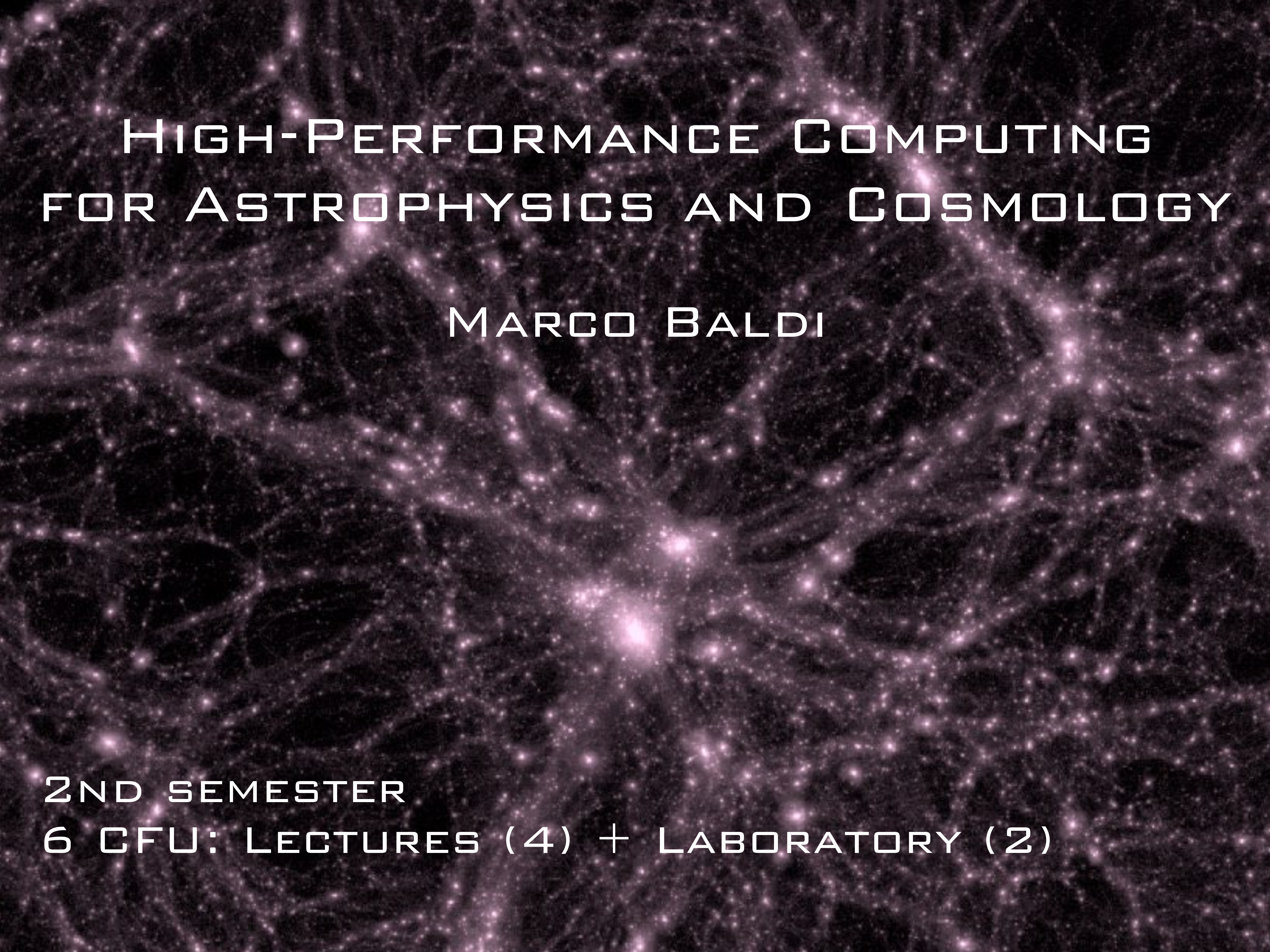


High-Performance Computing for Astrophysics and Cosmology

Marco Baldi

2nd semester

6 CFU: Lectures (4) + Laboratory (2)

Parallel Computing

[i.e. how to do more work in less time…]

Say you have **some work** to do…

Parallel Computing

[i.e. how to do more work in less time…]

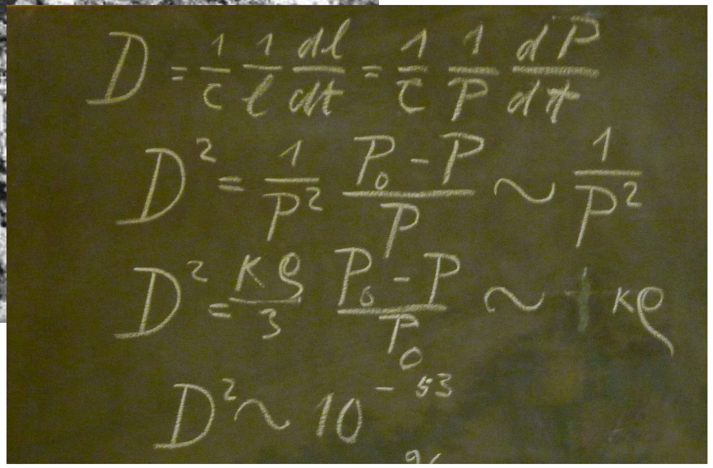
Say you have **some work** to do…

- For **small problem** size the direct solution is often the fastest way to achieve the goal (not efficient, but **easy to implement**)

Parallel Computing

[i.e. how to do more work in less time…]

Say you have **some work** to do…



- For **small problem** size the direct solution is often the fastest way to achieve the goal (not efficient, but **easy to implement**)

Parallel Computing

[i.e. how to do more work in less time…]

Say you have **some work** to do…

- For **small problem** size the direct solution is often the fastest way to achieve the goal (not efficient, but **easy to implement**)
- To **improve efficiency**, you may invest time in **developing tools**

Parallel Computing

[i.e. how to do more work in less time…]

Say you have **some work** to do…

- For **small problem** size the direct solution is often the fastest way to achieve the goal (not efficient, but **easy to implement**)
- To **improve efficiency**, you may invest time in **developing tools**

Parallel Computing

[i.e. how to do more work in less time…]

Say you have **some work** to do…

- For **small problem** size the direct solution is often the fastest way to achieve the goal (not efficient, but **easy to implement**)
- To **improve efficiency**, you may invest time in **developing tools**
- If the problem size increases, you can try to **improve your technology**

Parallel Computing

[i.e. how to do more work in less time…]

Say you have **some work** to do…

- For **small problem** size the direct solution is often the fastest way to achieve the goal (not efficient, but **easy to implement**)
- To **improve efficiency**, you may invest time in **developing tools**
- If the problem size increases, you can try to **improve your technology**

Parallel Computing

[i.e. how to do more work in less time…]

Say you have **some work** to do…

- For **small problem** size the direct solution is often the fastest way to achieve the goal (not efficient, but **easy to implement**)
- To **improve efficiency**, you may invest time in **developing tools**
- If the problem size increases, you can try to **improve your technology**

At some point, you may **reach a limit** in the efficiency of your tool. But the **size** of your problem may still **keep increasing**… what to do?

Parallel Computing

[i.e. how to do more work in less time…]

Say you have **some work** to do…

- For **small problem** size the direct solution is often the fastest way to achieve the goal (not efficient, but **easy to implement**)
- To **improve efficiency**, you may invest time in **developing tools**
- If the problem size increases, you can try to **improve your technology**

At some point, you may **reach a limit** in the efficiency of your tool. But the **size** of your problem may still **keep increasing**… what to do?

GO PARALLEL!

Parallel Computing

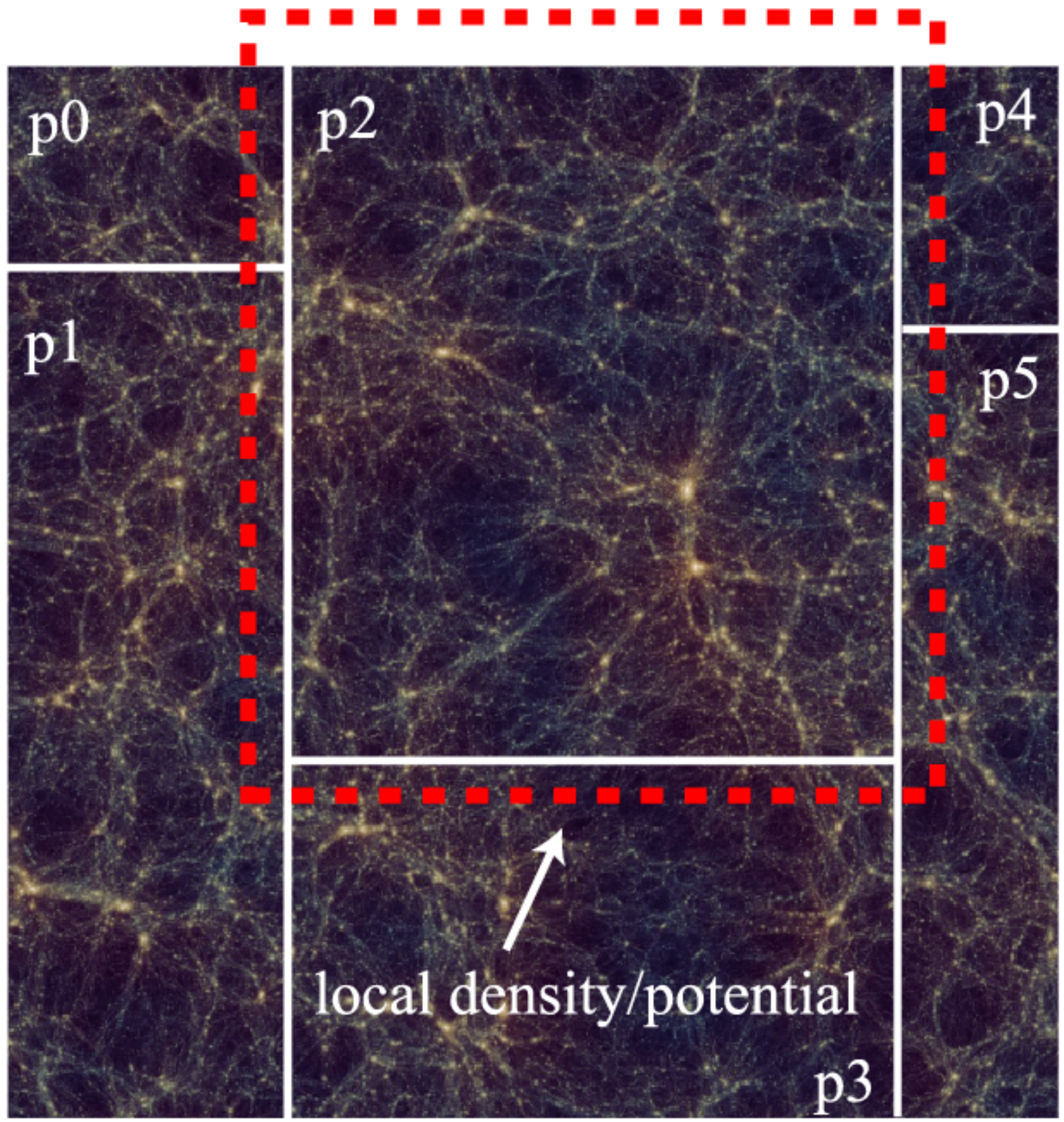
[i.e. how to do more work in less time…]

High-Performance Computing

HPC is a **type of parallel computing** where a large and complex problem is **distributed** over many computational units that perform a given set of operations **on their own portion** of the problem dataset and then **exchange data and instructions** through a **fast network**

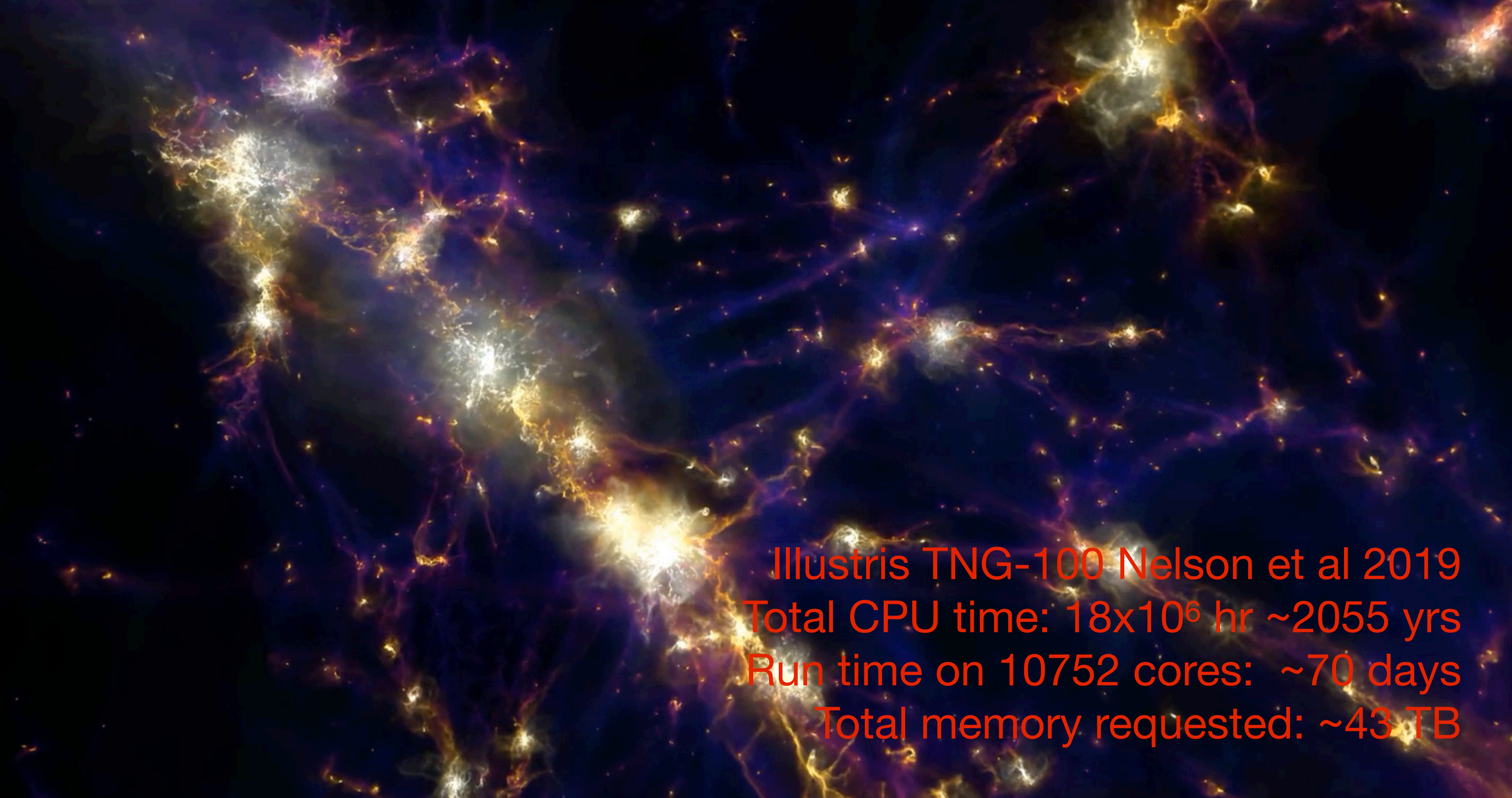
High-Performance Computing

HPC is a **type of parallel computing** where a large and complex problem is **distributed** over many computational units that perform a given set of operations **on their own portion** of the problem dataset and then **exchange data and instructions** through a **fast network**



High-Performance Computing

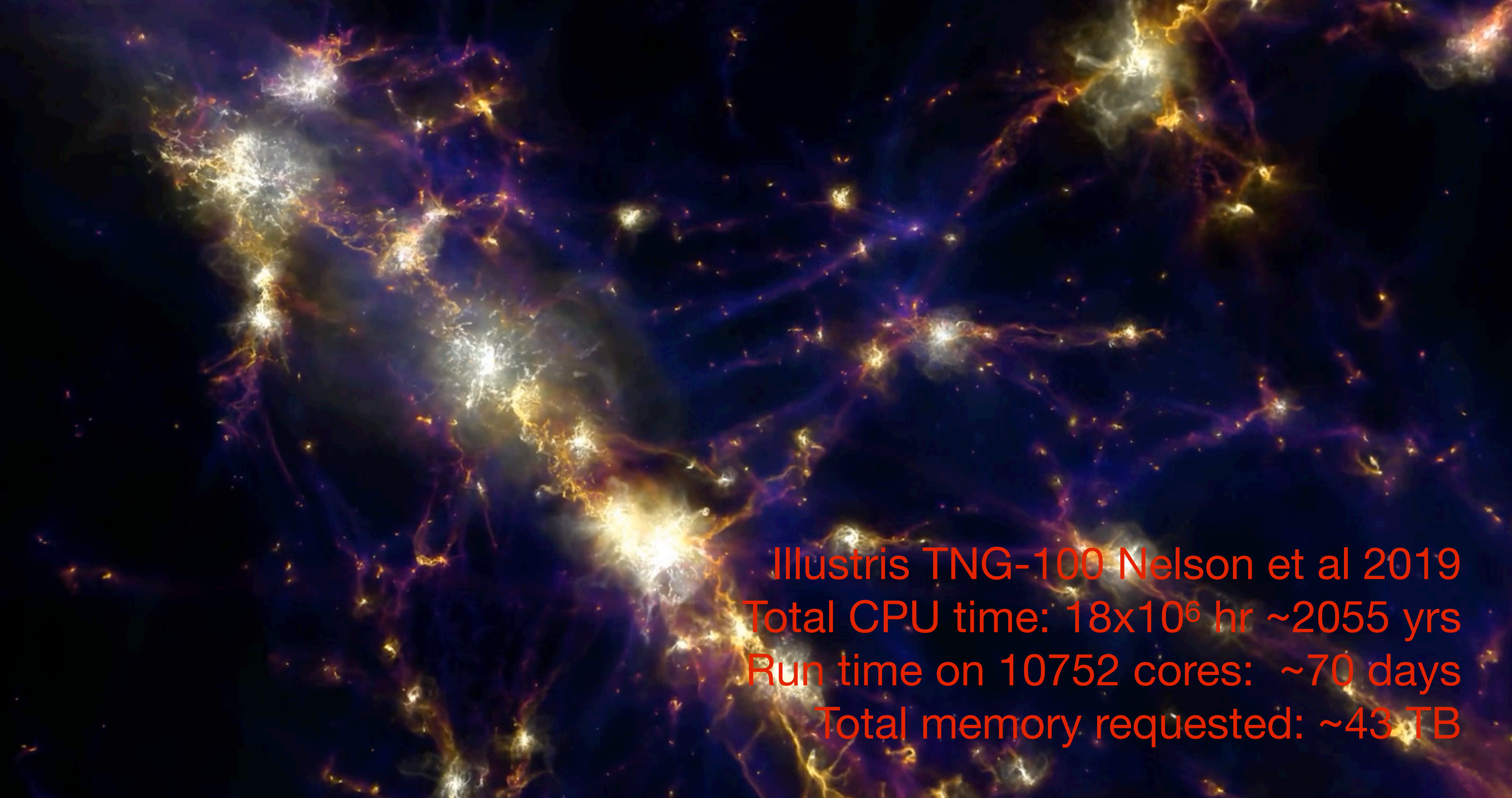
HPC is **particularly important for Astrophysics and Cosmology**
- **Theory** (simulations and modelling)
- **Observations** (data processing and analysis)



Illustris TNG-100 Nelson et al 2019
Total CPU time: 18×10^6 hr ~2055 yrs
Run time on 10752 cores: ~70 days
Total memory requested: ~43 TB

High-Performance Computing

HPC is **particularly important for Astrophysics and Cosmology**
- **Theory** (simulations and modelling)
- **Observations** (data processing and analysis)



Illustris TNG-100 Nelson et al 2019
Total CPU time: 18×10^6 hr ~2055 yrs
Run time on 10752 cores: ~70 days
Total memory requested: ~43 TB

Course Objectives

- Acquire a **general knowledge of Parallel Computing** concepts, terminology, and code design strategies
- Acquire **practical skills** on remote access to shared computing environments, data handling, batch scheduling for parallel jobs
- Acquire basic knowledge of **Message Passing Interface** (MPI) protocols to implement parallel algorithms
- Apply all these skills to some **typical problems in astrophysics** and cosmology (N-body) using the public code Gadget-2
- Embark in **Master Thesis or PhD projects** on Computational Astrophysics/Cosmology

Course Program & Structure

Course Program & Structure

Chapter 1
Introduction to Parallel Computing

1.1 General overview and scientific applications
1.2 Concepts and terminology
1.3 Memory Architectures
1.4 Parallel Programming Models
1.5 Design of parallel algorithms
1.6 **Laboratory** exercises on basic parallelisation strategies

Course Program & Structure

Chapter 1
Introduction to Parallel Computing

1.1 General overview and scientific applications
1.2 Concepts and terminology
1.3 Memory Architectures
1.4 Parallel Programming Models
1.5 Design of parallel algorithms
1.6 **Laboratory** exercises on basic parallelisation strategies

Chapter 2
Practical tools

2.1 Basic Unix Commands
2.2 Working remotely
2.3 Basics of bash scripting
2.4 Regular Expressions
2.5 Batch jobs scheduling
2.6 **Laboratory** exercises on bash scripting and data handling

Course Program & Structure

Chapter 1
Introduction to Parallel Computing

1.1 General overview and scientific applications
1.2 Concepts and terminology
1.3 Memory Architectures
1.4 Parallel Programming Models
1.5 Design of parallel algorithms
1.6 **Laboratory** exercises on basic parallelisation strategies

Chapter 2
Practical tools

2.1 Basic Unix Commands
2.2 Working remotely
2.3 Basics of bash scripting
2.4 Regular Expressions
2.5 Batch jobs scheduling
2.6 **Laboratory** exercises on bash scripting and data handling

Chapter 3
Introduction to MPI parallel programming

3.1 General concepts of Message Passing
3.2 Getting started with MPI
3.2 MPI environment management
3.4 Point-to-Point communications
3.5 Collective communications
3.6 Group and Communicator Management
3.6 **Laboratory** exercises on MPI parallelisation

Course Program & Structure

Chapter 1
Introduction to Parallel Computing

1.1 General overview and scientific applications
1.2 Concepts and terminology
1.3 Memory Architectures
1.4 Parallel Programming Models
1.5 Design of parallel algorithms
1.6 **Laboratory** exercises on basic parallelisation strategies

Chapter 2
Practical tools

2.1 Basic Unix Commands
2.2 Working remotely
2.3 Basics of bash scripting
2.4 Regular Expressions
2.5 Batch jobs scheduling
2.6 **Laboratory** exercises on bash scripting and data handling

Chapter 3
Introduction to MPI parallel programming

3.1 General concepts of Message Passing
3.2 Getting started with MPI
3.2 MPI environment management
3.4 Point-to-Point communications
3.5 Collective communications
3.6 Group and Communicator Management
3.6 **Laboratory** exercises on MPI parallelisation

Chapter 4
HPC in Astrophysics and Cosmology

4.1 Overview of N-body gravity solvers
4.2 N-body parallelisation strategies: domain decomposition and load balancing
4.3 The TreePM N-body code Gadget2
4.4 **Laboratory**: Examples of N-body sims
   4.4.1 Galaxy collisions (and/or)
   4.4.2 Cluster formation (and/or)
   4.4.3 Cosmic Large-Scale Structure

Course Program & Structure

Chapter 1
Introduction to Parallel Computing

1.1 General overview and scientific applications
1.2 Concepts and terminology
1.3 Memory Architectures
1.4 Parallel Programming Models
1.5 Design of parallel algorithms
1.6 **Laboratory** exercises on basic parallelisation strategies

Chapter 2
Practical tools

2.1 Basic Unix Commands
2.2 Working remotely
2.3 Basics of bash scripting
2.4 Regular Expressions
2.5 Batch jobs scheduling
2.6 **Laboratory** exercises on bash scripting and data handling

Chapter 3
Introduction to MPI parallel programming

3.1 General concepts of Message Passing
3.2 Getting started with MPI
3.2 MPI environment management
3.4 Point-to-Point communications
3.5 Collective communications
3.6 Group and Communicator Management
3.6 **Laboratory** exercises on MPI parallelisation

Chapter 4
HPC in Astrophysics and Cosmology

4.1 Overview of N-body gravity solvers
4.2 N-body parallelisation strategies: domain decomposition and load balancing
4.3 The TreePM N-body code Gadget2
4.4 **Laboratory**: Examples of N-body sims
- 4.4.1 Galaxy collisions (and/or)
- 4.4.2 Cluster formation (and/or)
- 4.4.3 Cosmic Large-Scale Structure

Reference teaching material: lecture slides

Course Tools

This course is part of the **Open Physics Hub** project:
https://site.unibo.it/openphysicshub
and will have **direct access to the DIFA HPC cluster**
(laboratory sessions will be performed hands-on on this cluster)

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

OPEN PHYSICS HUB - DEPARTMENT OF PHYSICS AND ASTRONOMY

HOME | PROJECT | INFRASTRUCTURES | ORGANISATION | PARTNERSHIPS | CONTACT US | INTERNAL | OPH NEWSLETTERS

Home / Infrastructures / HPC Cluster "Matrix"

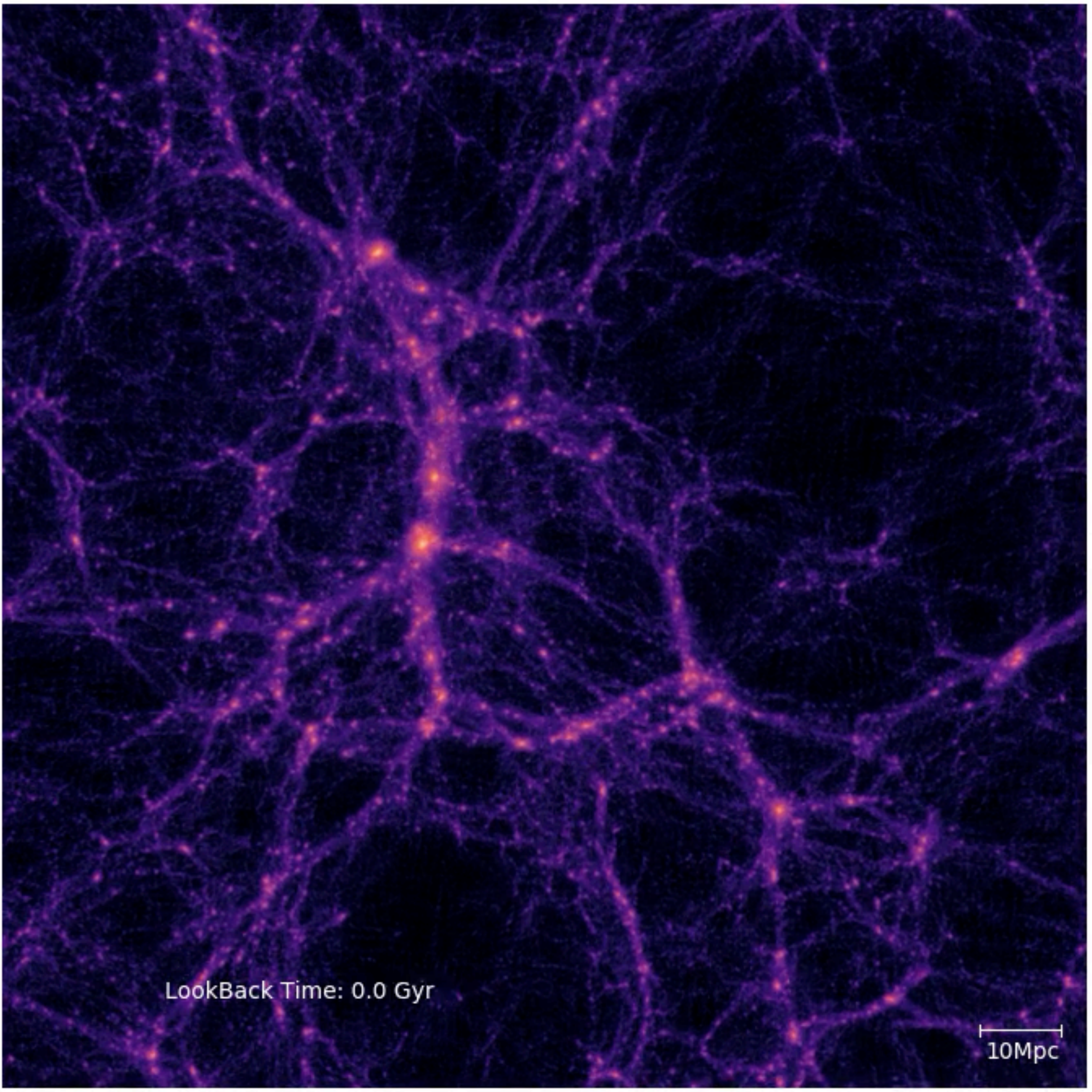
HPC Cluster "Matrix"

OPH has recently deployed and installed a High Performance Computing cluster called **"Matrix" with 1952 virtual cores and 4 GB RAM/core, equipped with 500 TB of disk storage space**, used both for DIFA research activities and innovative teaching courses. More specifically, the "Matrix" computing cluster features:

- **22 compute nodes** featuring multi-core Intel Xeon processors with hyperthreading
- **7.8 TB of RAM** for an average of 8 GB per physical core, 4 GB per thread
- Infiniband Mellanox **100 Gb/s low-latency connection** switch
- **3 disk nodes** with 30 disks of 12 TB/disk

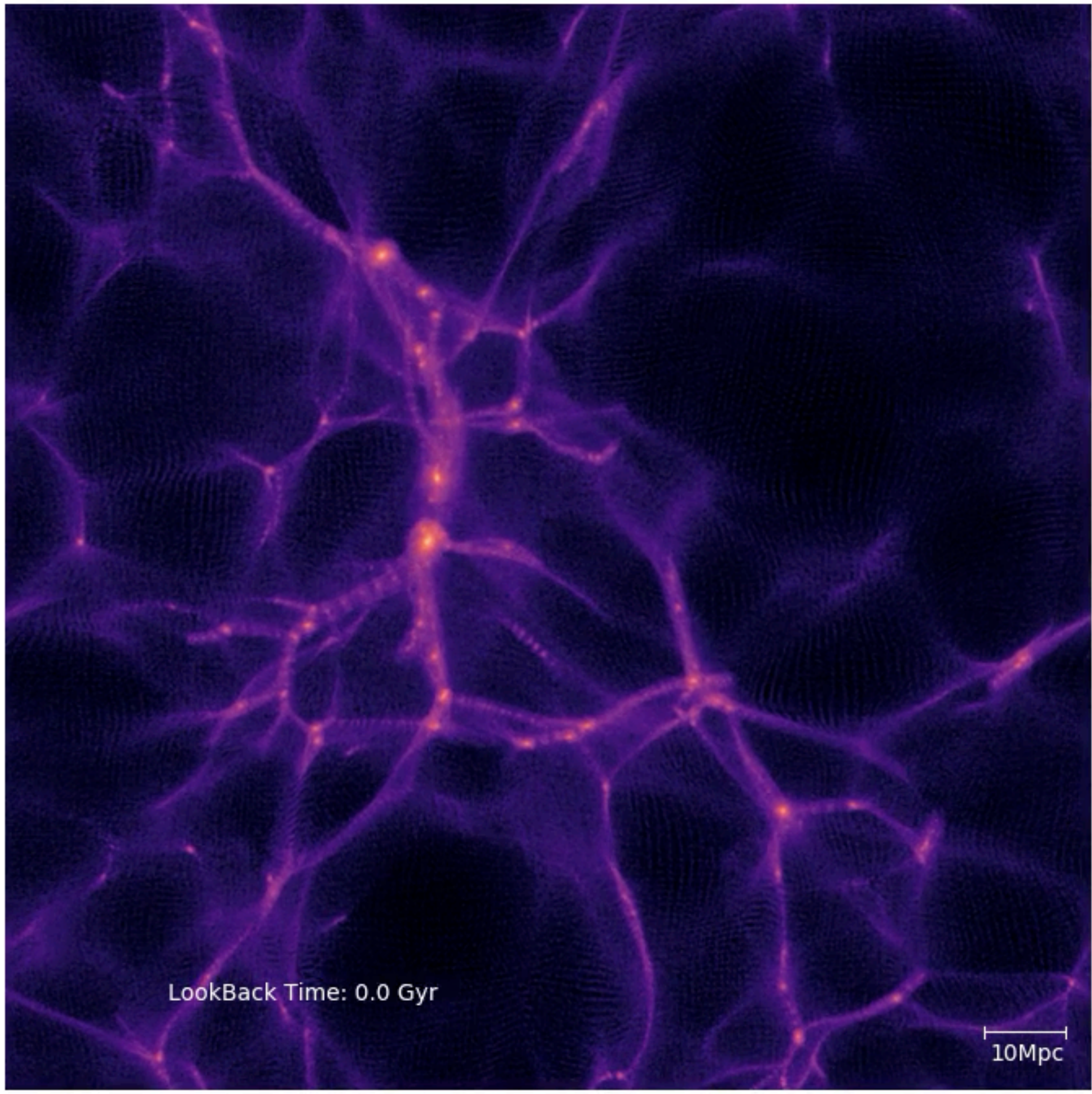
Fundamental Physics & Cosmology

Neutralinos



Gadget3
(Springel 2005)

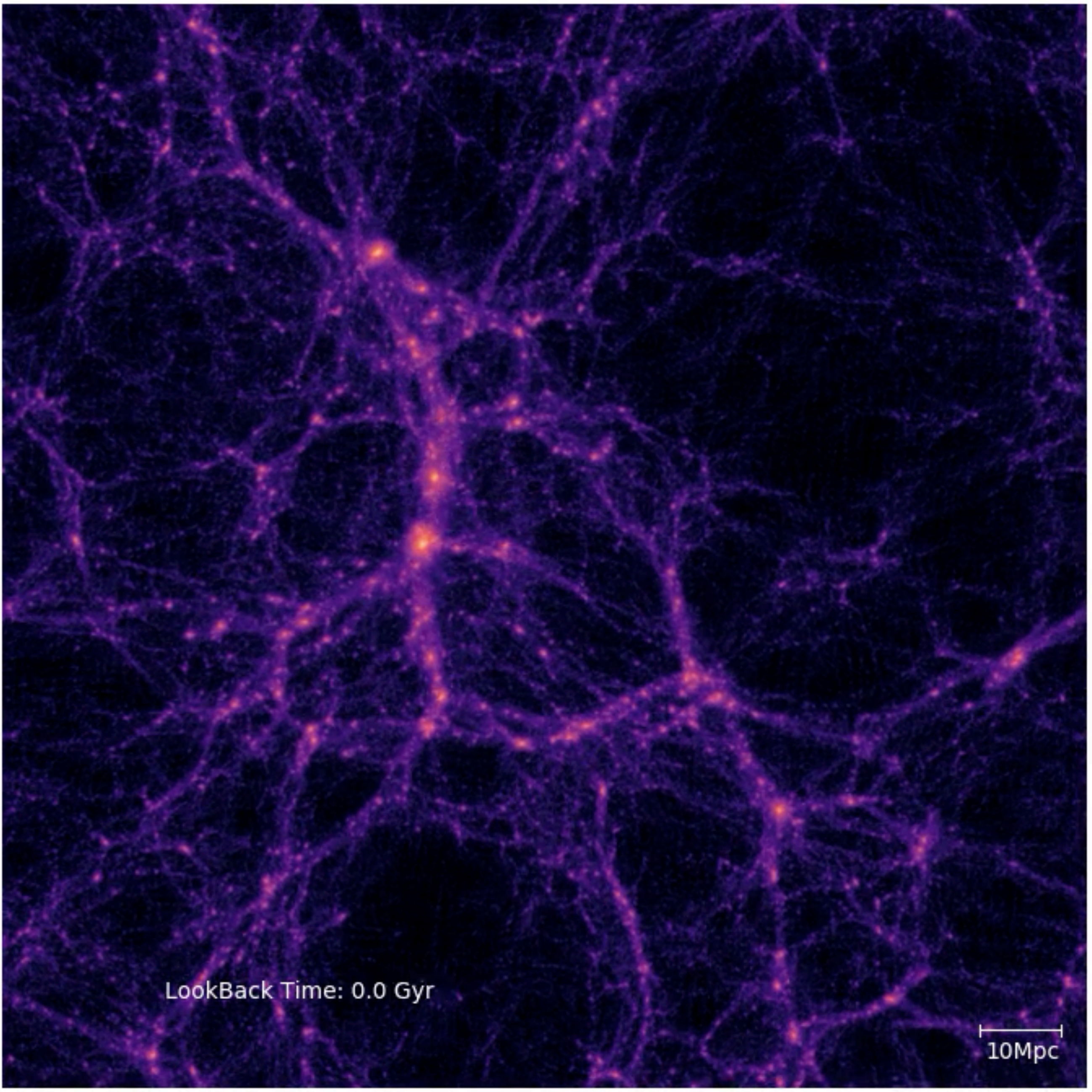
Axions



AX-Gadget
(Nori & Baldi 2018)

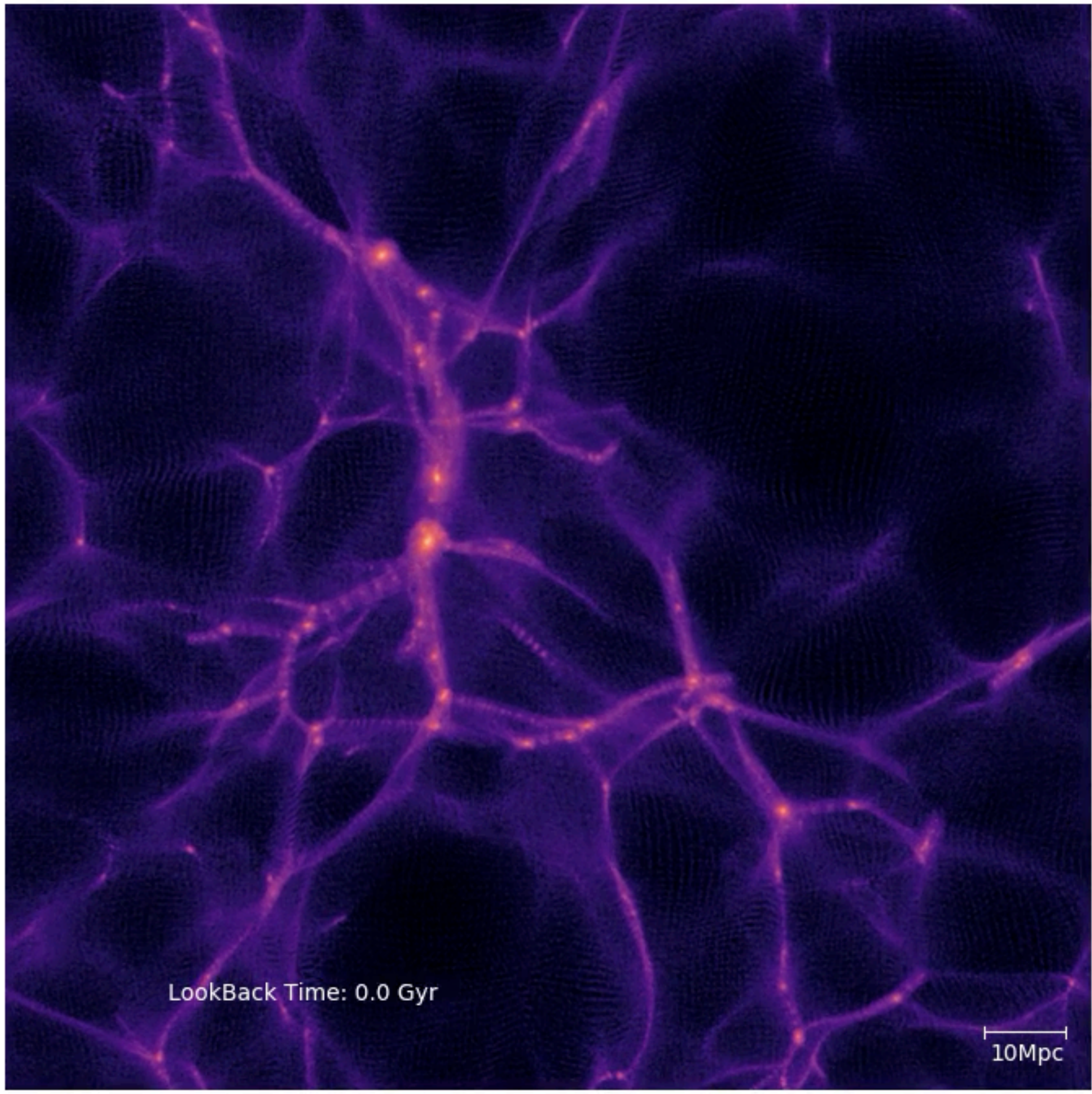
Fundamental Physics & Cosmology

Neutralinos



Gadget3
(Springel 2005)

Axions



AX-Gadget
(Nori & Baldi 2018)

High-Performance Computing for Astrophysics and Cosmology

Marco Baldi

For any additional information:
marco.baldi5@unibo.it
Office 4S5

2nd semester
Course Language: English
6 CFU: Lectures (4) + Laboratory (2)