SPACE

CENTRE OF EXCELLENCE FOR HPC ASTROPHYSICAL APPLICATIONS

VisIVO

Interactive High Performance Visualization

Nicola Tuccari, Eva Sciacca

Co-funded by the European Union

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Belgium, Czech Republic, France, Germany, Greece, Italy, Norway, and Spain under grant agreement No 101093441.

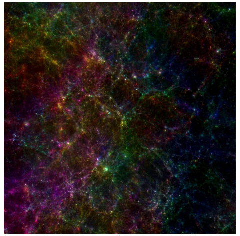
EuroHPC Joint Undertaking

The SPACE Centre of Excellence

Scalable Parallel Astrophysical Codes for Exascale (SPACE)

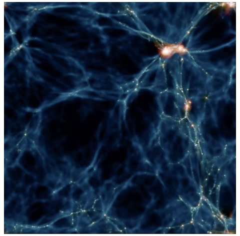
- EU Centre of Excellence focused on Astrophysical and Cosmological (A&C) applications
- Extensively re-engineer A&C codes for the efficient and effective exploitation of exascale computing capabilities
- High-performance data analysis of the data torrent produced by exascale A&C simulation applications with machine-learning and visualization tools.

OpenGADGET



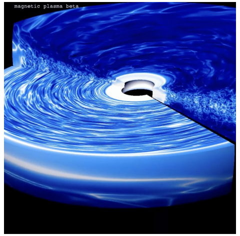
cosmological simulations

CHANGA



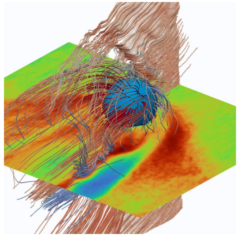
magneto hydro dynamics code

PLUTO



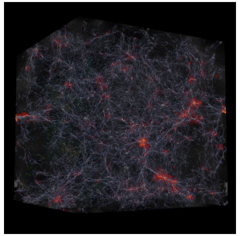
astrophysical plasma simulations

iPic3D



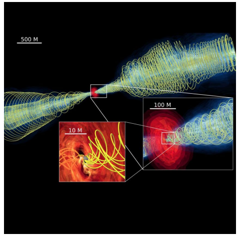
plasma physics

RAMSES



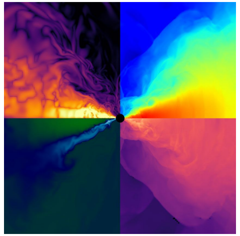
AMR cosmological simulations

BHAC



Black Hole accretion

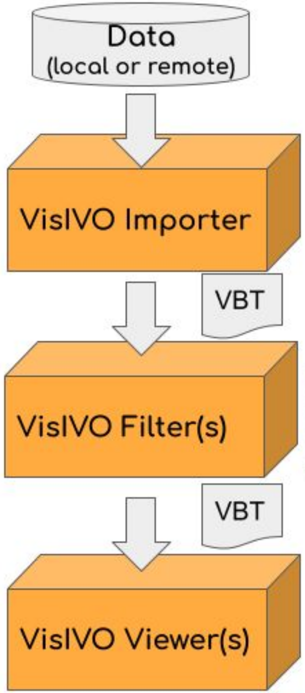
FIL

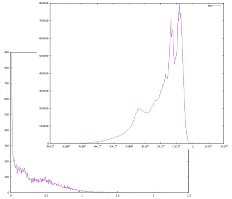


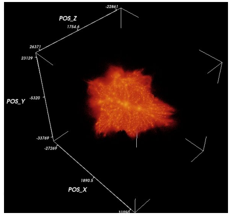
general-relativistic MHD

Visualization Interface for the Virtual Observatory

- Developed and maintained by INAF OACT since 2005
- Performs 3D and multi-dimensional **data analysis** and **knowledge discovery** of a-priori unknown relationships between multi-variate and complex astrophysical datasets.
- VisIVO is deployed in a variety of flavours as follows:
  - **VisIVO Server** - a CLI platform for high performance visualization,
  - **VisIVO Library** - for running complex workflows on DCI, clouds and HPC infrastructures.







https://visivo.readthedocs.io/

Extreme Data Processing and Analysis

Specific Objectives

To identify **use cases and requirements** for Visualisation and Data Analysis

To optimally orchestrate simulation pipelines and data analysis processing with **workflow engines**

To exploit and adapt existing solutions for high performance **visualisation** of data generated on the Exascale systems by the A&C applications

Extreme Data Processing and Analysis

Specific Objectives

To identify **use cases and requirements** for Visualisation and Machine Learning exploitation

To optimally orchestrate simulation pipelines and data analysis processing with **workflow engines**

To exploit and adapt existing solutions for high performance **visualisation** of data generated on the Exascale systems by the A&C applications

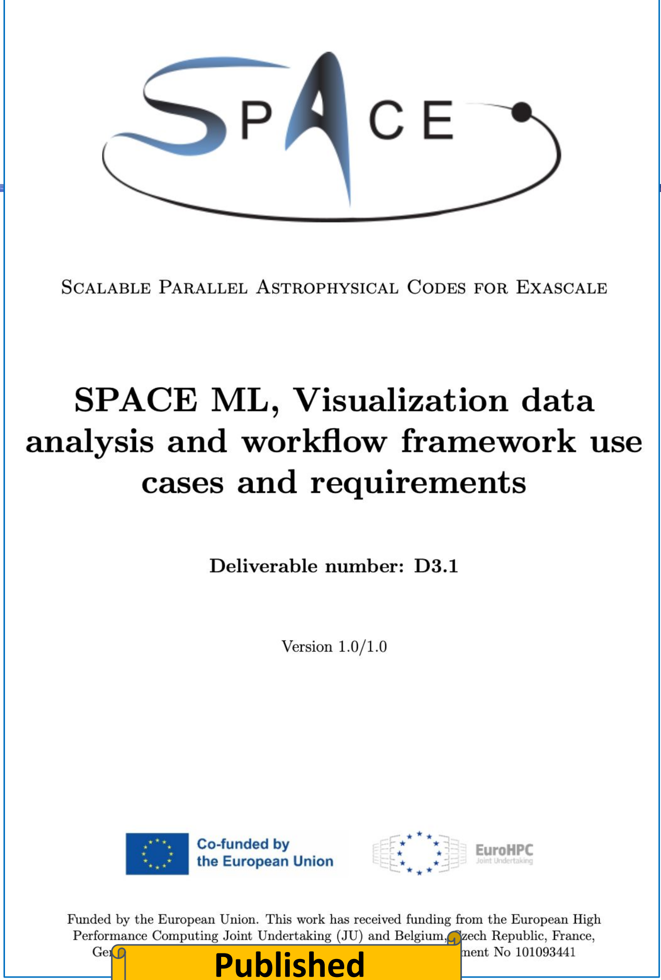
Identification of use cases for visualization and suitable tasks to exploit ML

Developed in collaboration with the scientific partners

- Topology-aware workflows for modular Exascale data analysis

- In-situ parallel visualization through Hecuba
- Interactive, volume-based cinematic visualization
- Interactive, distributed visualization for HPC and cloud environments

- Representation Learning for explorative knowledge discovery in simulations
- Physics-informed reconstruction using synthetic observations of cosmological simulations
- Predicting the effect of radiation on gas cooling rates on-the-fly in hydrodynamical simulations



SPACE

Scalable Parallel Astrophysical Codes for Exascale

SPACE ML, Visualization data analysis and workflow framework use cases and requirements

Deliverable number: D3.1

Version 1.0/1.0

Co-funded by the European Union

EuroHPC Joint Undertaking

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Belgium, zech Republic, France, Ge nent No 101093441

Published December 2023

Extreme Data Processing and Analysis

Specific Objectives

To identify **use cases and requirements** for Visualisation and Machine Learning exploitation

To optimally orchestrate simulation pipelines and data analysis processing with **workflow engines**

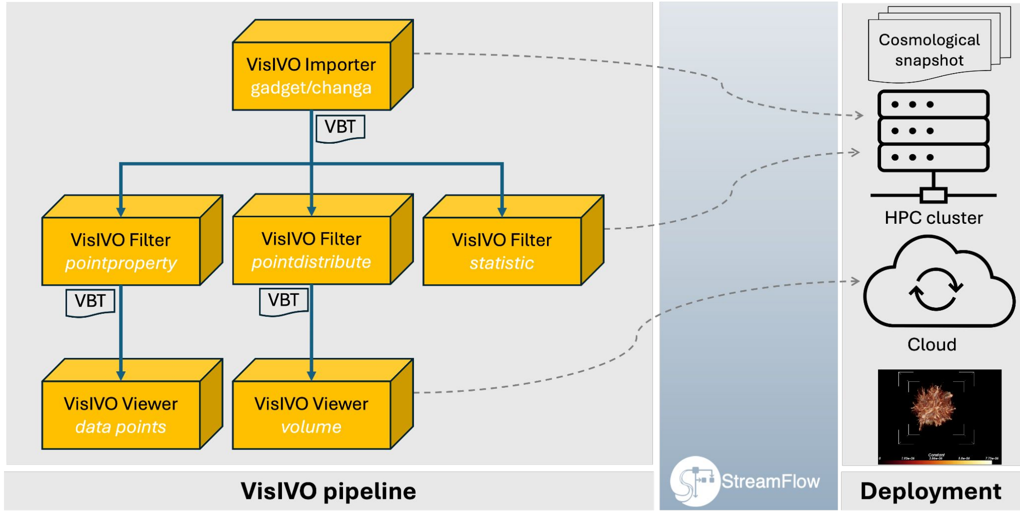
To exploit and adapt existing solutions for high performance **visualisation** of data generated on the Exascale systems by the A&C applications

Hybrid Workflow Model

The main motivations for integrating VisIVO with a Hybrid Workflow Model are:

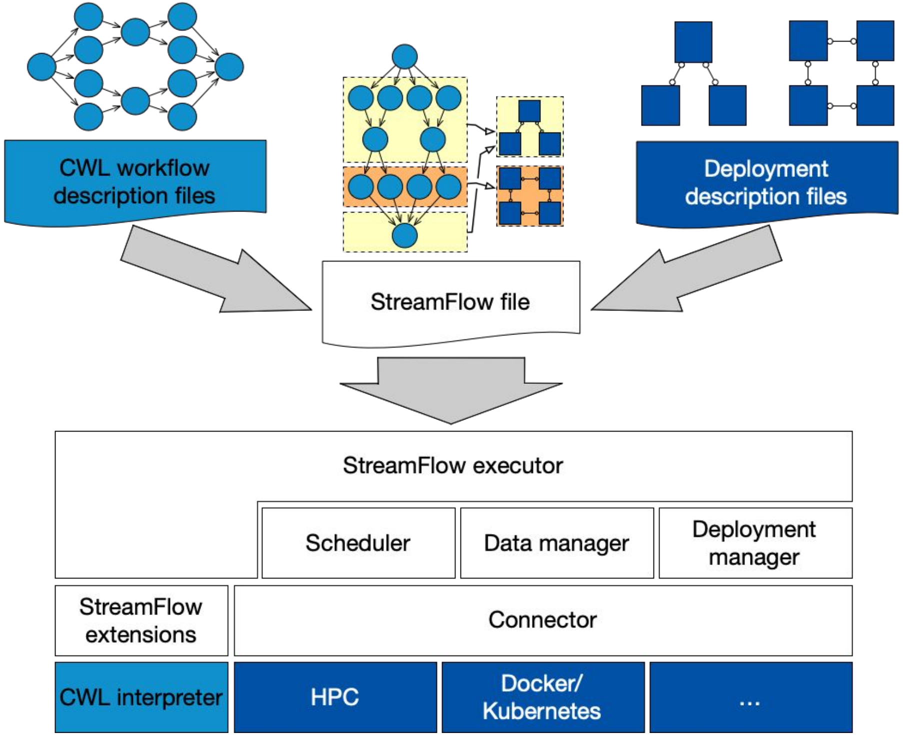
- Enabling execution across various computing infrastructures, such as **mixed HPC-Cloud** resources, for high-performance visualization of large-scale HPC application outcomes in A&C.

- Independent workflow steps can be executed in a **concurrent** fashion on top of multi-agent ecosystems (Cloud/HPC)



A container-native **Workflow Management System** based on the Common Workflow Language (CWL) standard and designed around two main principles:

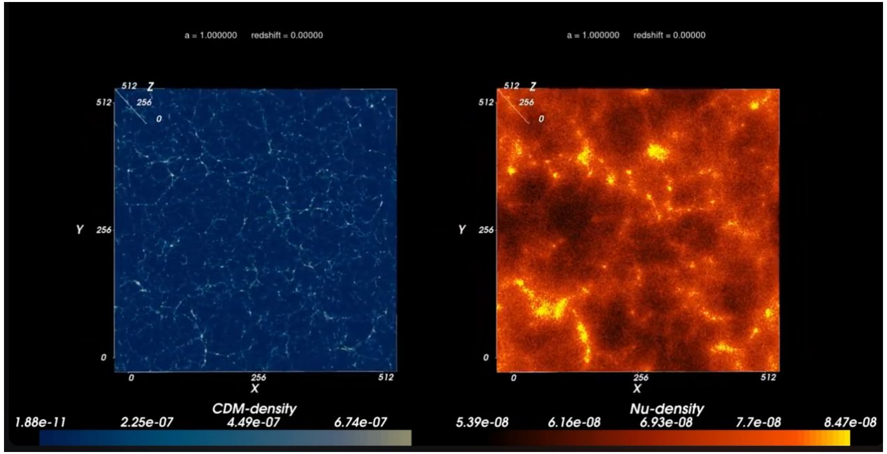
- Allowing the execution of tasks in **multi-container environments**, in order to support **concurrent execution** of multiple communicating tasks in a multi-agent ecosystem;
- Relaxing the requirement of a single shared data space, in order to allow for **hybrid workflow executions** on top of hybrid **cloud/HPC** infrastructures.



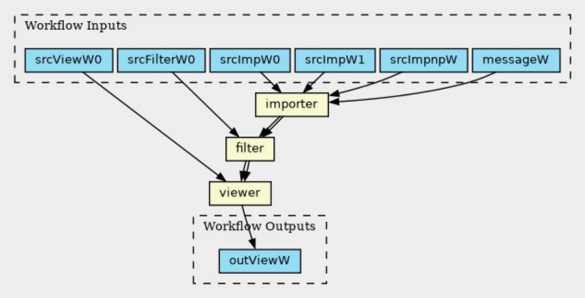
Workflows Rendering Results

- We developed workflows to support the studies of the effects of **massive neutrinos** on the large scale structure of the Universe (DEMNUni simulations) through visualization.

- Cosmological neutrinos strongly affect the evolution of the largest structures in the Universe, i.e., galaxies, galaxy clusters, cosmic voids, and filaments.



Visualization of CDM and neutrinos particles



Extreme Data Processing and Analysis

Specific Objectives

To identify **use cases and requirements** for Visualisation and Machine Learning exploitation

To optimally orchestrate simulation pipelines and data analysis processing with **workflow engines**

To exploit and adapt existing solutions for high performance **visualisation** of data generated on the Exascale systems by the A&C applications

Systematic Code Refactoring

- **Main Objective:** decreasing cyclomatic complexity to improve maintainability and readability while preserving functionality and enabling seamless addition of new features

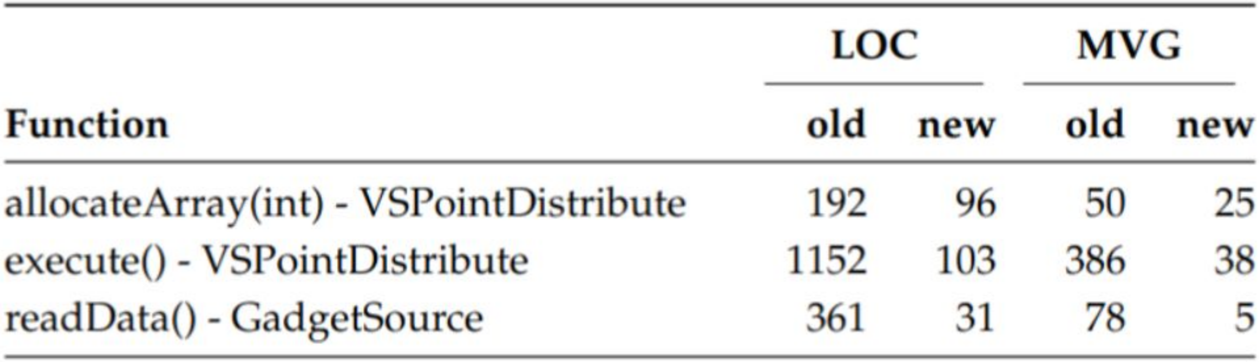
- We identified classes requiring **refactoring** by using the metrics generated by the tools, in particular those with high cyclomatic complexity

- Then applied Refactoring techniques, the main ones are the following:
  - **Extract Method**
  - **Grouping Related Data into Structs**
  - **Remove Dead Code**

Code Refactoring: Results

- **VSPointDistribute**: the main function *execute()* MVG got reduced by 90% (386 ⟶ 38) and *allocateArray()* function was halved (50 ⟶ 25). MVG for all the new extracted methods is lower than 15.

- **GadgetSource**: the main function *readData()* MVG got reduced by 94% (78 ⟶ 5) MVG for all the new extracted methods is lower than 8.

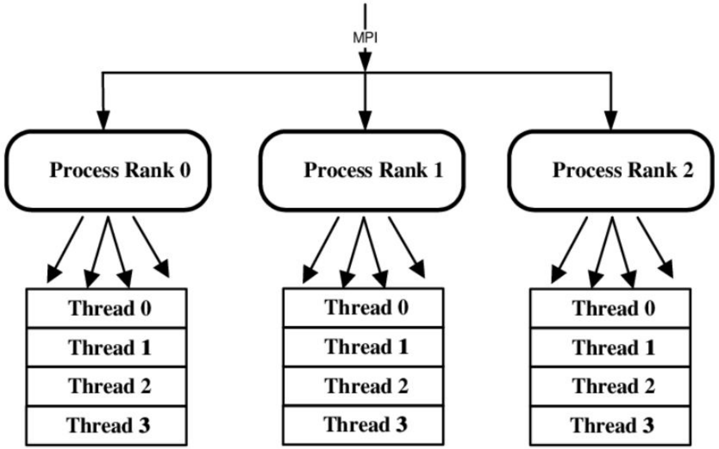


	LOC		MVG	
Function	**old**	**new**	**old**	**new**
allocateArray(int) - VSPointDistribute	192	96	50	25
execute() - VSPointDistribute	1152	103	386	38
readData() - GadgetSource	361	31	78	5

VisIVO Importer Parallelization: Hybrid MPI-OpenMP Parallel Model

We focused on the parallelization of one of the VisIVO Importers and employed an **Hybrid MPI-OpenMP Parallel Model**:

- Work is distributed over multiple MPI processes, using a **file-per-process** approach
- Then the I/O operation are further distributed over multiple threads within each process over the same file
- We used thread-safe I/O functions, such as **pread** and **pwrite**.

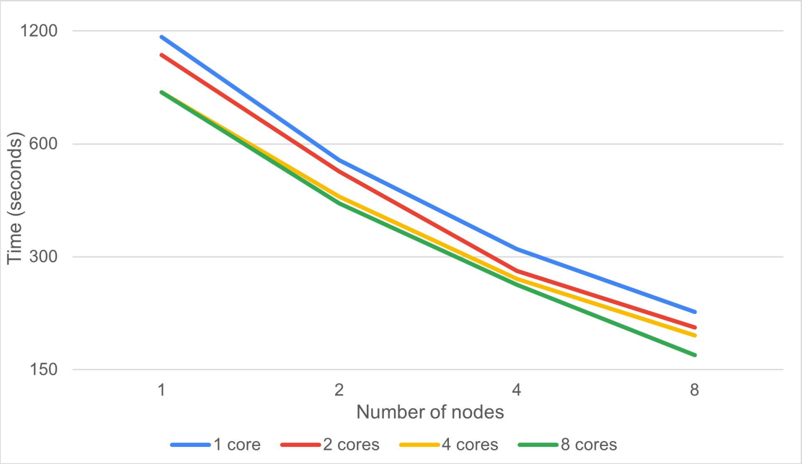


VisIVO Importer Parallelization: Preliminary Results

Preliminary tests on the VisIVO Importer module were performed on the Cineca Galileo100 platform using a 512GB GADGET snapshot:

- With the use of multiple nodes the importer scales linearly using up to 4 nodes, with an **efficiency of 0.89** (**3.56x speed-up**), then the efficiency decreases due to I/O contention.

- Multithreading is limited by the single node bandwidth, we obtain a speedup by using up to 4 or 8 cores because the datasets contain only 4 blocks.

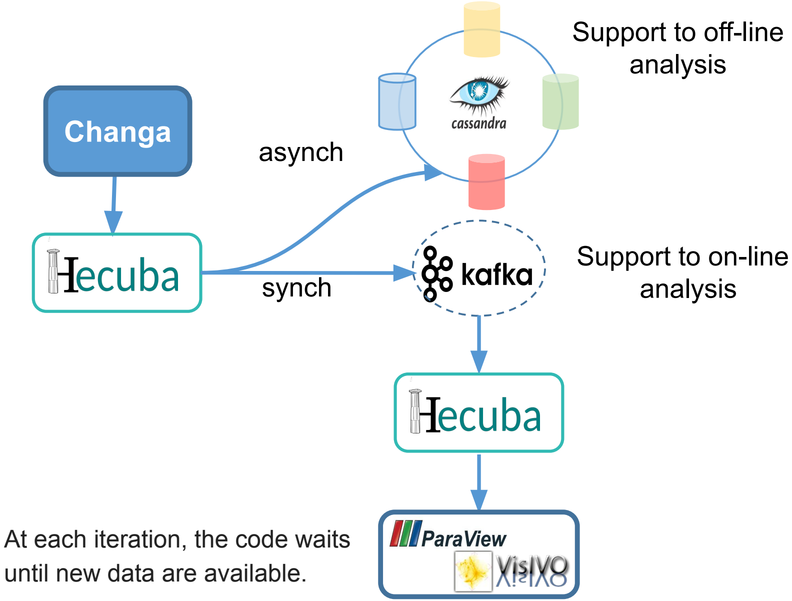


VisIVO In-Situ implementation: Hecuba and Use Case

- **Hecuba** is is a toolset designed to manage persistent data for Big Data applications:
  - It implements an **Object Mapper** for Apache Cassandra and for Apache Kafka
  - Provides an interface for Python and **C++**

- Our objective is to use Hecuba to enable the **in-situ visualization** of data generated by ChaNGa

- The desired pipeline is structured as follows:
  ChaNGa code uses Hecuba enabling streaming of data→ VisIVO Importer retrieves data using Hecuba API → VisIVO Viewer renders the data from the Importer



At each iteration, the code waits until new data are available.

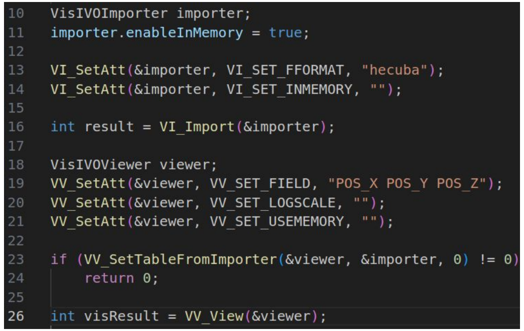
VisIVO In-Situ implementation: Results

- **In-Memory data handling:** new subclass VSTableMem, redefining multiple functions to operate in memory

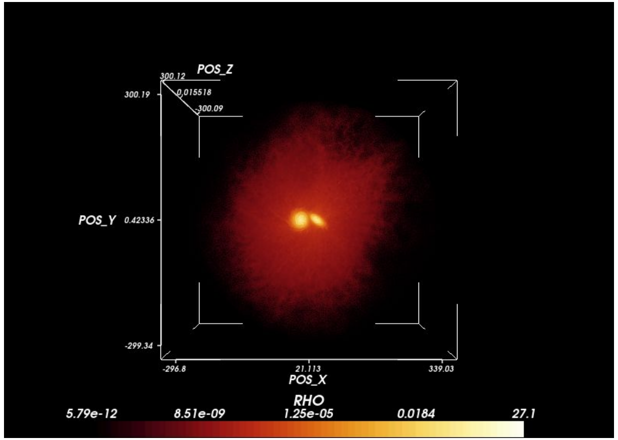
- **Hecuba Importer:** new VisIVO Importer that retrieves data from Hecuba

- **VisIVO Library API Update:** Allow user to use in-memory implementation using a new library function to attach importer output to the Viewer: *VV_SetTableFromImporter(VisIVOViewer* v, VisIVOImporter* i, size_t t_index)*;

- **Test pipeline:** Hecuba ChaNGa Simulator → Importer (in-memory) → Viewer; this demonstrated end-to-end without storage writes, with an expected **speedup of 3.3x**



```
VisIVOImporter importer;
importer.enableInMemory = true;

VI_SetAtt(&importer, VI_SET_FFORMAT, "hecuba");
VI_SetAtt(&importer, VI_SET_INMEMORY, "");

int result = VI_Import(&importer);

VisIVOViewer viewer;
VV_SetAtt(&viewer, VV_SET_FIELD, "POS_X POS_Y POS_Z");
VV_SetAtt(&viewer, VV_SET_LOGSCALE, "");
VV_SetAtt(&viewer, VV_SET_USEMEMORY, "");

if (VV_SetTableFromImporter(&viewer, &importer, 0) != 0)
    return 0;

int visResult = VV_View(&viewer);
```



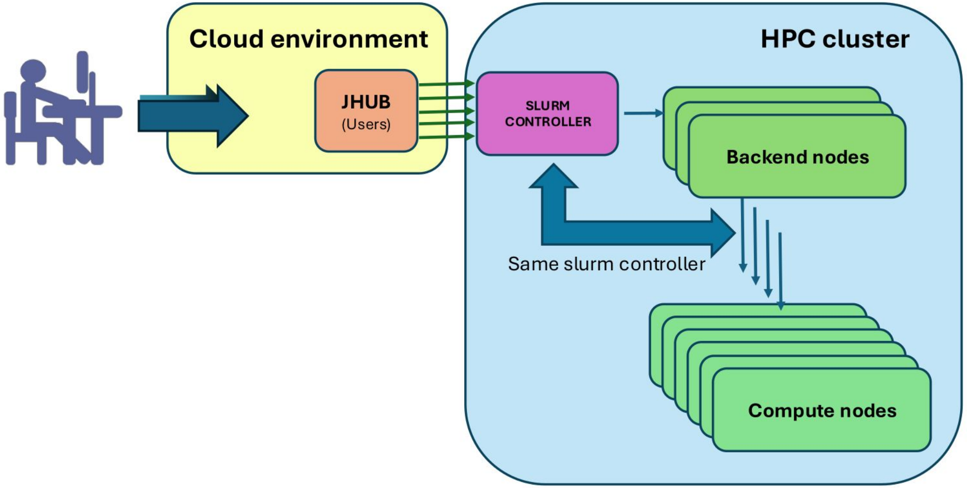
VisIVO and IAC integration: IAC

The **InterActive Computing service** (IAC) has been developed as part of the Fenix research infrastructure by Cineca and E4 Company.

It is currently up and running on the GALILEO100 cluster.

- Users can authenticate and select the desired HPC resources configuration

- Usability is enhanced by providing an interactive **Jupyter Notebook** environment while transparently leveraging HPC resources.

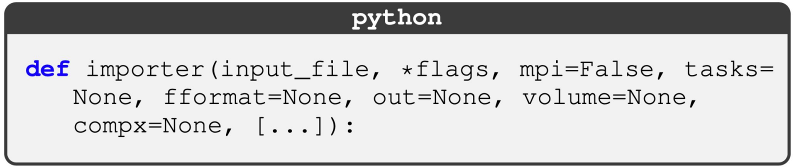


VisIVO and IAC integration: VisIVO Wrappers

We implemented a **Python wrapper** to connect Python and the compiled VisIVO binaries.

Using the **subprocess** package to run the underlying VisIVO commands, mapping the three main VisIVO commands: VisIVOImporter, VisIVOFilter, and VisIVOViewer.

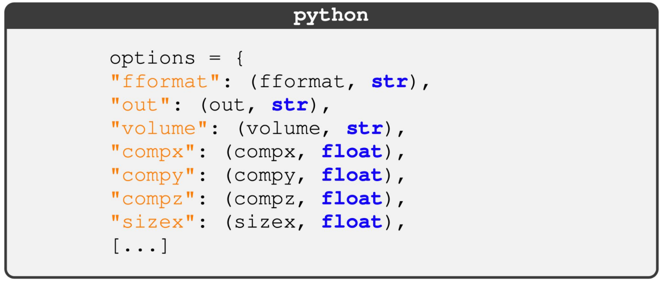
- Options are converted as arguments of the function



python

```python
def importer(input_file, *flags, mpi=False, tasks=
    None, fformat=None, out=None, volume=None,
    compx=None, [...]):
```

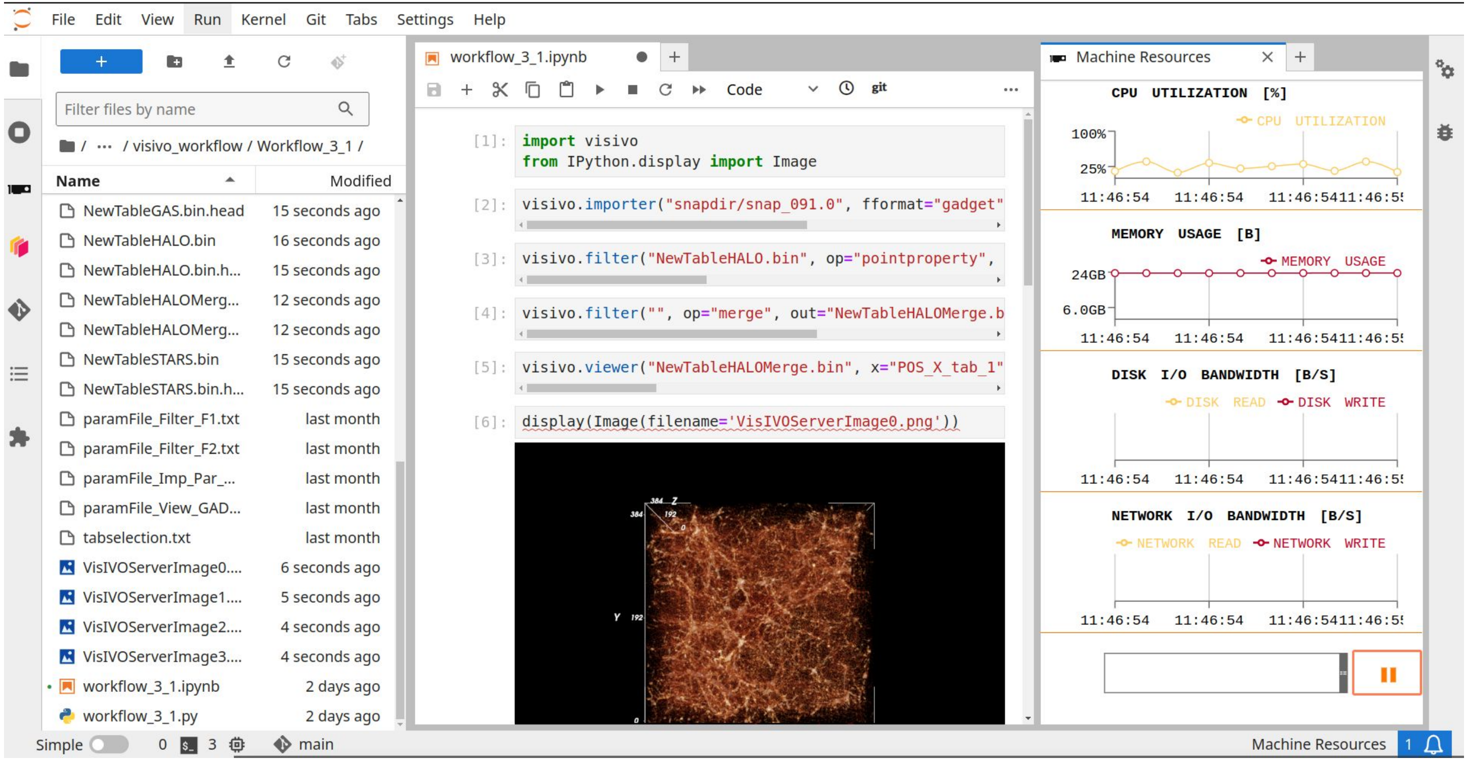
- The variable type is checked using an options dictionary



python

```python
        options = {
        "fformat": (fformat, str),
        "out": (out, str),
        "volume": (volume, str),
        "compx": (compx, float),
        "compy": (compy, float),
        "compz": (compz, float),
        "sizex": (sizex, float),
        [...]
```

VisIVO and IAC integration: Notebook



Name	Modified
NewTableGAS.bin.head	15 seconds ago
NewTableHALO.bin	16 seconds ago
NewTableHALO.bin.h...	15 seconds ago
NewTableHALOMerg...	12 seconds ago
NewTableHALOMerg...	12 seconds ago
NewTableSTARS.bin	15 seconds ago
NewTableSTARS.bin.h...	15 seconds ago
paramFile_Filter_F1.txt	last month
paramFile_Filter_F2.txt	last month
paramFile_Imp_Par_...	last month
paramFile_View_GAD...	last month
tabselection.txt	last month
VisIVOServerImage0....	6 seconds ago
VisIVOServerImage1....	5 seconds ago
VisIVOServerImage2....	4 seconds ago
VisIVOServerImage3....	4 seconds ago
workflow_3_1.ipynb	2 days ago
workflow_3_1.py	2 days ago

```
[1]: import visivo
     from IPython.display import Image

[2]: visivo.importer("snapdir/snap_091.0", fformat="gadget"

[3]: visivo.filter("NewTableHALO.bin", op="pointproperty",

[4]: visivo.filter("", op="merge", out="NewTableHALOMerge.b

[5]: visivo.viewer("NewTableHALOMerge.bin", x="POS_X_tab_1"

[6]: display(Image(filename='VisIVOServerImage0.png'))
```

Conclusions

- The integration of **VisIVO** with **StreamFlow** allowed to improve **portability** and **reproducibility**, enabling efficient visualization across heterogeneous HPC environments.

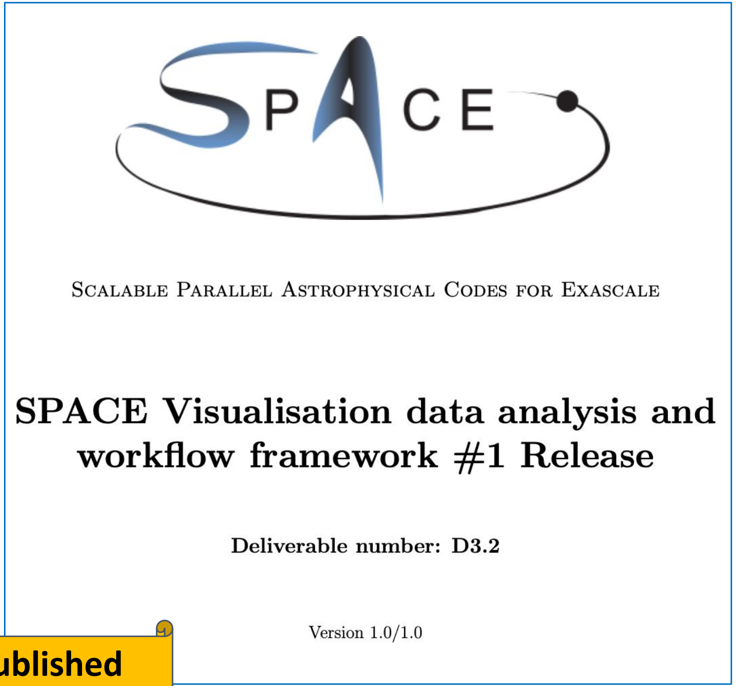
- I/O has been improved for **VisIVO**: 1) parallelization of the importer module reached a **speed-up** of **5.48x** over 8 nodes; 2) **In-situ pipeline** allow to avoid reliance to storage and we expect a **3.3x speedup**.

- The integration of **VisIVO** into the **InterActive Computing** service, through the development **Python wrappers** and a dedicated Jupyter Kernel, allowed interactive analyses and **visualizations** abstracting the underlying large-scale HPC resources and simplified **user interaction**.

SPACE Release #1

https://www.space-coe.eu/visualization.php

- Topology-aware workflows for modular Exascale data analysis
- In-situ parallel visualization through Hecuba
- Interactive, volume-based cinematic visualization
- Interactive, distributed visualization for HPC and cloud environments



Scalable Parallel Astrophysical Codes for Exascale

SPACE Visualisation data analysis and workflow framework #1 Release

Deliverable number: D3.2

Version 1.0/1.0

Published June 2025

Next Release #2 foreseen in December 2026

Related Publications

E. Sciacca, N. Tuccari, U. Arshad, F. Pitari, G. Muscianisi, C. Carbone, F. Vitello, B. Montalto, “**VisIVO interactive high performance visualization**”,
Astronomy and Computing, Volume 55, 2026, 101069, ISSN 2213-1337,
https://doi.org/10.1016/j.ascom.2026.101069

Tuccari, N., Sciacca, E., Becerra, Y., Cintero, E. S., & Tramontana, E. (2025). “**Exascale In-Situ Visualization for Astronomy & Cosmology**”.
https://ceur-ws.org/Vol-4130/paper104.pdf

N. Tuccari et al., "**High Performance Visualization for Astrophysics and Cosmology**," 2025 33rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), Turin, Italy, 2025, pp. 443-450, doi:
https://doi.org/10.1109/PDP66500.2025.00069

Acknowledgement & Disclaimer

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Belgium, Czech Republic, France, Germany, Greece, Italy, Norway, and Spain under grant agreement No 101093441.

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European High Performance Computing Joint Undertaking (JU) and Belgium, Czech Republic, France, Germany, Greece, Italy, Norway, and Spain. Neither the European Union nor the granting authority can be held responsible for them

Co-funded by the European Union

EuroHPC Joint Undertaking