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

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

Project Information

Project Acronym: SPACE
Project Full Title: Scalable Parallel Astrophysical Codes for Exascale
Call: Horizon-EuroHPC-JU-2021-COE-01
Grant Number: 101093441
Project URL: https://space-coe.eu

Document Information

Editor:	Guillermo Marín, BSC
Deliverable nature:	Report (R)
Dissemination level:	Public (PU)
Contractual Delivery Date:	31.12.2023
Actual Delivery Date	20.12.2023
Number of pages:	40
Keywords:	Data analytics, machine learning, artificial intelligence, visualization, workflows
Authors:	Guillermo Marin – BSC Eva Sciacca – INAF Sebastian Trujillo Gomez – HITS Andreas Fehlner – HITS Petr Strakos – IT4I@VSB Iacopo Colonnelli – UNITO Doriana Medić – UNITO Yolanda Becerra – BSC Dario Garcia – BSC Oriol Lehmkuhl – BSC
Peer review:	Luca Tornatore – INAF Tristan Coulange – CNRS

History of Changes

Release	Date	Author, Organization	Description of changes
0.1	06.07.2023	Eva Sciacca, INAF	Provided template
0.2	14.09.2023	Guillermo Marin, BSC; Eva Sciacca, INAF	Provided first draft of the Table of Contents
0.3	19.10.2023	Guillermo Marin, BSC	Added Intro and Methodology
0.4	24.10.2023	Sebastian Trujillo G., HITS	Added ML Use Case 1 description
0.41	25.10.2023	Sebastian Trujillo G., HITS	Added some general information and context for ML use cases 2 and 3
0.5	26.10.2023	Guillermo Marin, BSC; Andreas Fehlner, HITS	Defined Requirements and a bullet point structure for the use cases
0.6	26.10.2023	Eva Sciacca, INAF	Added VisIVO use case
0.61	16.11.2023	Petr Strakos, IT4I@VSB	Added interactive cinematic visualization use case
0.62	20.11.2023	Eva Sciacca, INAF	Contributed to sections 1 and 4
0.63	20.11.2023	Doriana Medić, UNITO	Contributed to Workflows use case
0.7	20.11.2023	Guillermo Marin, BSC	General edit and Conclusions
0.71	24.11.2023	Guillermo Marin, BSC	Added information to use cases 4.2 and 5.3
0.8	27.11.2023	Guillermo Marin, BSC	Homogenize document
0.81	29.11.2023	Eva Sciacca, INAF; Guillermo Marin, BSC	Edit Executive summary, Appendixes, overall final checks and restructuring
0.82	30.11.2023	Guillermo Marin, BSC	Document ready for review
0.9	14.12.2023	Luca Tornatore, INAF; Tristan Coulange, CNRS	Peer review
1.0	20.12.2023	Eva Sciacca, INAF; Guillermo Marin, BSC	Final edits

DISCLAIMER

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.

The space above and below the message intentionally is left blank.

Executive Summary

This deliverable D3.1 "SPACE ML, visualization data analysis and workflow framework use cases and requirements" is a report on requirements and use cases for the SPACE Machine Learning and visualization tools, and topology-aware workflows for modular High Performance Computing (HPC) applications.

The use cases and requirements have been collected in WP3 Task T3.2: "Identification of user cases for visualization and suitable tasks to exploit ML" and in close collaboration with the A&C community in WP1, in order to better identify post-processing analysis for the simulation data products and to integrate run-time modules suitable for coupling the WP1 exascale applications with visualization tools (such as VisIVO, Blender and Paraview) and Machine Learning techniques including representation learning, generative AI and Convolutional Neural Networks.

This deliverable will guide the future development of such tools in WP3 for the duration of the project, as it provides the base layer of technical, non-technical and functional constraints, requirements, and needs of the code developers and users.

Contents

This document is Public (PU)
and was produced under SPACE project EU GA 101093441

List of Figures

List of Tables

List of Acronyms

A&C	Astrophysics and Cosmology
AI	Artificial Intelligence
CFD	Computational Fluid Dynamics
CoE	Centre of Excellence
CWL	Common Workflow Language
GNNs	Graph Neural Networks
HPC	High Performance Computing
ICL	Intra Cluster Light
ML	Machine Learning
SPACE	Scalable Parallel Astrophysical Codes for Exascale
VA	Visual Analytics
VBT	VisIVO Binary Table
VisIVO	Visualization Interface for the Virtual Observatory
WMS	Workflow Management System

This document is Public (PU)
and was produced under SPACE project EU GA 101093441

1 Introduction

One objective of the Scalable Parallel Astrophysical Codes for Exascale (SPACE) Centre of Excellence (CoE) is to integrate data analysis techniques with exascale applications in Astronomy and Cosmology to enhance scientific discoveries from the results of numerical simulations. SPACE WP3 "Extreme data processing and analysis" addresses this topic by developing prototype frameworks based on ML, visualization tools and workflow engines for the exascale applications that will be developed in WP1. These frameworks will be tailored to the Astrophysics and Cosmology community, represented by the developers and end-users of the flagship simulation codes of the SPACE CoE, i.e.: OpenGADGET, PLUTO, Bhac, Changa/Gasoline, Fil, iPic3D, Ramses and WhiskyTHC.

Within the WP3 Task T3.2: "Identification of user cases for visualization and suitable tasks to exploit ML", visualization, Artificial Intelligence (AI) and astrophysics experts have reviewed existing data processing and analysis solutions, and identified components of specific interest and feasibility for their modelling in data analysis frameworks for the Astrophysics and Cosmology (A&C) simulation codes, whether as run-time or post-processing stages, and prioritising those components with reduced dependencies and interactions with other modules. This task was developed in close collaboration with WP1, in order to gather the requirements related to better exploitation of post-processing simulation data products and of run-time modules suitable for coupling the applications with visualization and ML. Furthermore, interactive meetings and hands-on sessions engaged the A&C scientific community to collect the most representative use-case scenarios.

The use cases presented in this document will be developed during the span of the project into functional prototypes within WP3 Tasks T3.1: "Topology-aware workflows for modular HPC applications", T3.3: "High performance visualization for astrophysics" and T3.4: "Machine learning for astrophysics". The key performance indicator KPI.3.1."Use cases for visualization and of tasks suited to ML exploitation" related to the present deliverable states the need to clearly identify the use cases in order to exploit advanced visualization tools and ML techniques in post-processing. While it specifies a minimum of 2 use cases identified and reported in the present document, we have exceeded this number by identifying a total of 7 use cases: 3 in visualization, 3 in ML and 1 in workflows. In this document, we present the identified use cases and the user requirements contributed by the scientific partners, as well as the methodology used to both gather the requirements and select the use cases.

This document is organized as follows:

- Section 2 - Methodology, discusses the processes followed to gather the relevant information from all the partners involved and to structure this information into the user cases and requirements.
- Section 3 - Use Cases and requirements. This section presents the motivation, the main stakeholders and the structure of the use cases presented in the following sections of the document. The use cases sections reflect the classification of the WP3 tasks and activities according to the three main involved technologies, i.e.: High-performance visualization, Machine Learning assisted analysis and workflows.
- Section 4 presents the three identified use cases for high-performance visualization, i.e.: i) In-situ parallel visualization through Hecuba; ii) Interactive, volume-based cinematic visualization; iii) Interactive distributed visualization for HPC and cloud environments.
- Section 5 presents the three identified use cases for Machine Learning, i.e.: i) Representation learning for explorative knowledge discovery in simulations; ii) Physics-informed reconstruction using synthetic observations of cosmological simulations; iii) Predicting the effect of radiation on gas cooling rates on-the-fly in hydrodynamical simulations.
- Section 6 presents the identified use case for workflows: Topology-aware workflows for modular Exascale data analysis.
- Section 7 - Conclusions

2 Methodology

2.1 Resources

As a starting point to identify the use cases and requirements for visualization, machine learning, and workflows, two main resources were used:

- Data characteristics table. Spreadsheet of the technical characteristics of the simulations' input/output data.
- Requirements Questionnaire. Questionnaire specifically tailored for the code users and developers inside the SPACE CoE.

2.1.1 Data characteristics table

A comprehensive spreadsheet of the technical details of each code's input/output data. This spreadsheet was devised in WP1 and WP2, and contains information about file size, data format, data structure, simulation details, input parameters, etc. This resource serves as a valuable complement to section 3 of the Requirements Questionnaire, as it provides more thorough information of the data that WP3 will work with along the project. The table is available in Appendix A.

2.1.2 Requirements Questionnaire

Based on previous experiences, WP3 partners collectively created a questionnaire aimed at the A&C code developers and users. This questionnaire focuses on understanding current and desired uses of visualization and ML, opportunities and challenges of the exascale, and the characteristics of the output data from the simulations. Participants from WP1 filled in the questionnaire once per scientific use case. We gathered a total of 10 responses with at least one answer per code, specifically, 3 for OpenGADGET and one for each of the other 7 codes. The questionnaire is available in Appendix B.

The questionnaire is divided in the following sections:

- Instructions and contact details.
- 1. Scientific case. Name, description, and reference to the scientific use case that was used to fill in the questionnaire.
- 2A. Visualization: This section focuses on understanding how the participant currently uses visualization tools and techniques, and especially, what would they like to achieve in those terms that they can't currently. The questions are:
  - What do you mainly use visualization for?
  - Do you already use any visualization tool? Which one?
  - Are the tools you have tried not sufficient for your needs? why?
  - Are you planning on adopting an existing tool that may suit your needs? Which one and why?
  - Mark below the types of data visualizations that you produce more often or find more useful.
  - As a reference for us, add links to images, papers, documents, etc. of plots that you make and plots you would like to make
  - Mark the most relevant challenges to visualize your data
  - What would you like to use visualizations for, that you cannot now?
- 2B.Machine learning assisted analysis. Similarly to the previous section on visualization, this part focuses on understanding both current and desired uses and applications for Machine Learning.The questions are:
  - Do you currently perform any ML-assisted analysis? Which?
  - Desired (Already identified potential uses of Machine Learning) Which?
  - Mark potential applications of ML of interest to your case

This document is Public (PU)
and was produced under SPACE project EU GA 101093441

  - Why is the ML-assisted analysis in this case necessary/beneficial to address the scientific/research question?
  - Example of envisioned code use

- 2C. Exascale. This section inquires about foreseen challenges that might arise in the porting of the codes to exascale supercomputing. The questions are:
  - Example of envisioned code use
  - Mark potential challenges of porting to exascale, if any

- 3.Output data. Regards to aspects of the simulation results that are relevant in Ml and Vi applications, e.g., file format, average size, data structure, output cadence, etc. The questions are:
  File characteristics
  - Current format? Is it convertible to hdf5?
  - Has Metadata/Description?
  - Approximate size
  - Data type (single, double, quad precision)?
  - Structure (perticles, FVM, FEM, AMR)
  - Are there post-process outputs beyond the raw output?

  Time
  - What is the simulated time span?
  - What is the typical post-process/snapshot frequency?
  - What is the estimated optimal temporal resolution to visualize results and how does this relate to time steps?

  Spatial Resolution
  - What is the Spatial scale and the Spatial Resolution? Is the discretization of the space uniform or multi-resolution?

  Variables/Fields
  - Which are the most relevant or typically used for analysis/visualization?
  - Do you compare several runs with different parameters? or distributions/ evolution within a single run?
  - Is there sample data of this case already available for use in visualization or Analysis?

2.2 Analysis of the results

The output of the first resource, the data characteristics table, provided a thorough understanding of aspects relevant in this present stage of gathering requirements and identifying the use cases. The most important of them are: file format, a measure of file size (average, maximum, etc.), data availability for testing, and presence of metadata. This information allowed us to identify cases which, for example, require more intensive computing resources for visualization, pose difficulties in moving and copying files due to big data sizes, or cases that share similar data structures and can potentially share visualization, ML, or workflows solutions.

Regarding the requirements questionnaire, we identified several opportunities for applying visualization and ML tools and techniques, and workflows. To illustrate the relations between these opportunities and the existing lines of work of WP3 partners, we created the spreadsheet "Proposed use cases". It contains a line for each response submitted by the partners to the questionnaire, and columns for the code name, main interests in visualization or ML, and possible matches among the solutions offered by the WP3 partners as described in the SPACE proposal. The spreadsheet is available in Appendix C.

The main criteria to translate the requirements into Use Cases were:

- There is a match between: a) the needs for a visualization or ML solution in the analysis pipeline for a given code, and b) the lines of work proposed in the description of Work Package 3 of the SPACE CoE proposal (e.g., a code whose output files are expected to use considerable disk space resources matches the proposal of an in-situ visualization pipeline, so the results can be visualized without extra disk usage).
- That the given use case can either be extrapolated to other similar codes in future work with minimum adaptation work, or is based on a code-independent post-processing stage and can be used with more than one of the codes in SPACE.

An important, complementary and transversal part of defining the use cases and requirements for this deliverable were P2P meetings to discuss the interest, requirements, and constraints of all the available possibilities. These meetings could involve two or more of these groups: the simulation code users and developers of WP1 as final users and code providers; the WP3 partners as responsible for the final development of the ML, visualization and workflows applications; and the Principal Investigator of the project or the Management Board as agents with a broader understanding of the project and the consortium. As an example, the identification of the codes that are best suited for modularization and for the insertion of run-time ML solutions in their exascale-optimized versions, was more efficiently performed through direct meetings with the code developers and scientists than through the requirements questionnaire because the adaptation of the codes to exascale architectures is an ongoing process and the core of the SPACE CoE.

This document is Public (PU)
and was produced under SPACE project EU GA 101093441

3 Use Cases and Requirements

3.1 Motivation.

Over the years, the A&C domain has developed a set of ad-hoc tools and software modules to tackle the challenging particularities of the field. With the emergence of ML, high-performance visualization, and Visual Analytics (VA) as enabling technologies, some of these components become candidates to be replaced by either faster, more accurate, or more efficient data-driven technologies. In SPACE WP3, experts in visualization and AI techniques applied to astrophysics have performed a review of the existing software of interest, and identified components of specific interest in the SPACE exascale codes to be integrated with visualization and ML techniques, addressing run-time and post-processing stages, and prioritising those components with reduced dependencies and interactions with other modules while maximising data usage and availability. In this task, WP3 partners worked in close collaboration with WP1 scientific partners to gather the use case requirements and the software specifications related to better exploitation of post-processing analysis products and of run-time modules suitable for coupling the applications with visualization and ML.

3.2 Stakeholders and Users

The selected ML and visualization use cases will be developed for and in close collaboration with the scientific partners of the SPACE project. However, the general goal is to address the whole A&C community by developing tools that are accessible and usable by a larger number of users. In this regard, some of the ML and visualization applications will be offline post-processes that use the results of the simulation codes and/or observational datasets as inputs, which will make them more easily ported to different simulation codes, and adopted by other members of the community outside SPACE. Other tools, however, will be devised as run-time solutions and hence embedded within the selected simulation codes. This will make portability and adaptability to other codes more difficult but still feasible. We will try to make this type of ML and visualization tools as modular and independent from the codes as possible, in order to facilitate its future adaptation to other codes. Based on the resources mentioned in the previous section, we have selected three visualization Use Cases, three ML use cases, and one workflow use case.

3.3 Structure of the use cases

In order to maintain coherence among the document, all the use cases follow the following structure:

- Title
- Use Case leading partner in WP3
- Statement of the problem
- Proposed solution
- Functional requirements
- Future applicability to other codes

4 Visualization Use Cases

The three identified use cases for high-performance visualization are described below.

4.1 In-situ Parallel visualization through Hecuba

Lead WP3 partner: BSC

Statement of the problem Traditional workflow prescribes storing the simulation results to disk and later retrieving them for analysis and visualization. However, at petascale this storage of the full results is prohibitive. A solution to this problem is to run the analysis and visualization concurrently (in-situ) with the simulation and bypass the storage of the full results. One mechanism for doing so is in transit visualization in which analysis and visualization is run on I/O nodes that receive the full simulation results but write information from analysis or provide run-time visualization.

Proposed solution The BSC will use Hecuba to adapt both the simulator and the visualization tool to interact with a highly distributed database. Hecuba is a set of tools and interfaces that aims to facilitate the interaction with key-value datastores. Hecuba implements an Object Mapper for Cassandra, a recognized noSQL distributed database, that allows programmers to use a common interface to access data as regular in-memory objects, regardless if they are persistent (stored in disk) or they are actual in-memory data.

The starting point will be previous work in which we extended the ParaView software package to use Hecuba as a data source. While this approach implemented a pipeline to provide off-line and on-line visualizations, it lacked a mechanism to synchronize on-line visualizations with the arrival of new data, requiring the user to manually refresh the visualization. Hecuba will be extended to implement a lambda architecture. Lambda architecture (3) is a data-processing architecture defined to speed up online analysis for Big Data applications, without losing the ability to persist the output data of the applications. With this approach, Hecuba will allow at the same time, to persist the generated data in a key-value datastore and to produce a stream of data for on-line visualization. Thus, Hecuba will be able to support both off-line and on-line data visualizations (see figure 1). With the streaming capability, the visualization tool will detect if new data is available and automatically refresh the visualization.

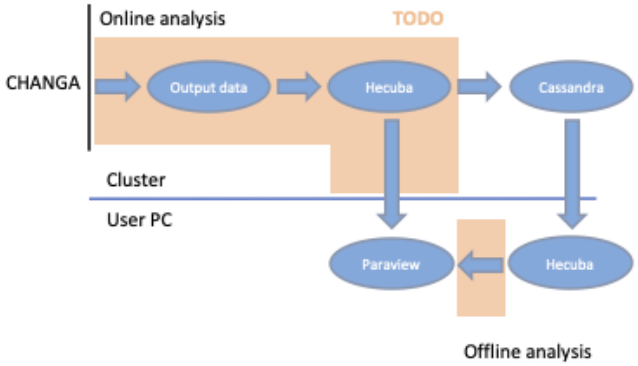

Figure 1: Workflow to support online and offline visualization using Hecuba and Paraview

Functional requirements The goal is to enable online visualization of the data output of the simulator. This online analysis allows to check the partial results in advance and to select which data is worth keeping in persistent storage to perform additional offline analysis once the simulator ends the execution. With this approach, on the one hand, scientists do not need to wait until the simulation finishes to start analyzing the data; on the other hand, it is possible to save storage time and storage capacity with a conscious selection of the data persisted.

This document is Public (PU)
and was produced under SPACE project EU GA 101093441

The development of this framework will require the implementation of a specific writer in Changa, the selected simulation code for this use case, that outputs the data files directly to Hecuba in a format also readable by the end visualization software: Paraview.

Regarding the execution platform, Hecuba can run on any general-purpose machine. The ideal way to extract the greatest possible performance is to use distributed systems with local disks on each node. It is compatible with Slurm-based queuing systems (although it is feasible to adapt the deployment scripts to other queuing systems). This lack of specific hardware requirements makes the solution easy to port from one HPC cluster to another.

Future applicability to other codes The Paraview plugin will use Hecuba to retrieve the data to visualize. The interface to retrieve the data will be mostly independent of the data source (online data, streamed during the execution of the application, or offline data, stored in the database), the only difference will be the specification of the base classes in the data class definition. This plugin could be used, potentially, with any spatial data that is readable by Paraview. The main complexity of the developments focuses on the new Hecuba code (to deal with the online scenario) and the deployment of the software stack (that should allow remote visualization). The modifications to the simulation software will involve adapting the data class definition to use Hecuba classes, which depends on the type of data generated by the simulator. A priori, this should translate in a flexible adaptability of the pipeline with reasonable effort to other SPACE A&C codes that, a) permit the implementation of Hecuba classes in the code, and b) the output data can be read with Paraview, such as OpenGadget, RAMSES, or PLUTO.

4.2 Interactive, volume-based cinematic visualization

Lead WP3 partner: IT4I

Statement of the problem Astrophysical and cosmological simulation codes demand intensive computational resources, generating datasets ranging from terabytes to petabytes. Efficient post-processing is integral to the simulation process for successful analysis. Among various post-processing techniques, volume rendering stands out as both illustrative and computationally intensive. This technique enables the simultaneous display of the entire volume of resulting data, offering a clear visualization of complex physical problems. Volume rendering has a distinct advantage in visualizing trillions of calculated values concurrently and efficiently manipulating this data across the analysis domain. Unlike other techniques, such as isosurfaces or streamlines, volume rendering does not require calculating additional geometric structures. Users can adjust transfer functions to modify the volume transparency, define color scales, and visualize intricate physical processes. Achieving high-fidelity outputs demands features like complex lighting, shading, detailed material effects, and advanced camera control. Classical visualization and analysis tools like ParaView and VisIt lack or have limitations in these advanced features, necessitating the exploration of alternative data visualization tools.

SPACE WP3 partner IT4I has developed a scalable approach to convert the results of a large-scale Computational Fluid Dynamics (CFD) simulation into a volumetric representation and provide high-fidelity volume rendering-based visualizations. The workflow for CFD data visualization consists of the following:

- Parallel loading of CFD data into distributed memory of a cluster.
- Data redistribution to improve load balance for voxelisation.
- Voxelisation and data conversion into a volumetric database, namely OpenVDB (1).
- Import of OpenVDB (1) data in a visual effects software. Our pipeline is currently based on Blender(4) using volume rendering as implemented in Blender Cycles path-tracing renderer (5) to produce cinematic-style visualizations (6), but other similar software packages could be used in this stage.

Using this workflow, it is possible to interactively visualize the data in a reduced path-tracing setup but still in high enough quality to observe and understand the captured phenomena. In this mode, it is possible to tweak shader parameters, scene lighting, camera setup, etc. and provide the final visualization setup of the

scene. Then the path-tracing renderer can be used as an offline tool to provide high-fidelity outputs. An example of a large-scale CFD simulation can be seen in Figure 2. It is an idealized supercell thunderstorm simulation.

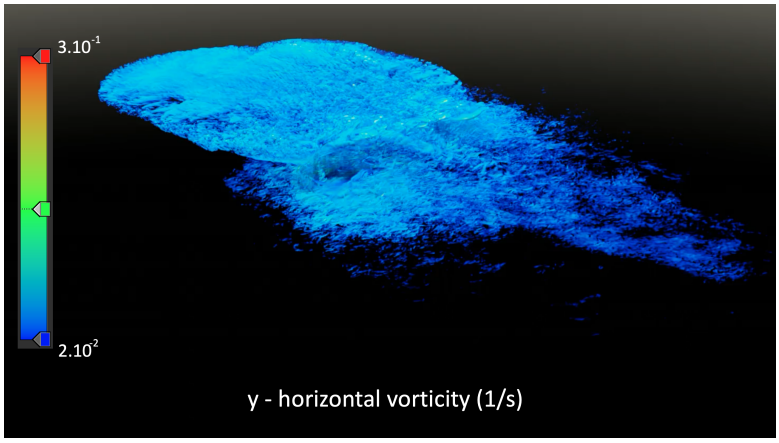

Figure 2: Volume rendering of data from CM1 (Cloud Model 1) converted to OpenVDB (1). CM1 is a three-dimensional, time-dependent, non-hydrostatic numerical model for atmospheric research made by George Bryan at The Pennsylvania State University. Parameters of the simulation: Domain size x = 127 km, y = 127 km, z = 9 km; Grid of 1700 x 1700 x 121 points; timax = 6000.0s (end time); tapfrq = 30.0s (timesteps); Output 1.6TB of data, 200 files in netCDF4 format; Simulated on 64 nodes, each 128 MPI processes, through 24hours.

The visualization Team of the BSC has also developed previous work consisting in a translator from simulation formats compatible with the VTK (7) format to multi-level OpenVDB (1) files. The aim is to generate sparse volumes that are more efficiently stored and loaded, and faster on render time.

Proposed solution Based on previous experience on CFD data, this use case will produce a visualization workflow around three main components: (i) a sparse volumetric format, (ii) Blender software (4) for visualization and user interaction, and (iii) path-tracing rendering via CyclesPhi (5) or another HPC compatible renderer. See the Figure 3.

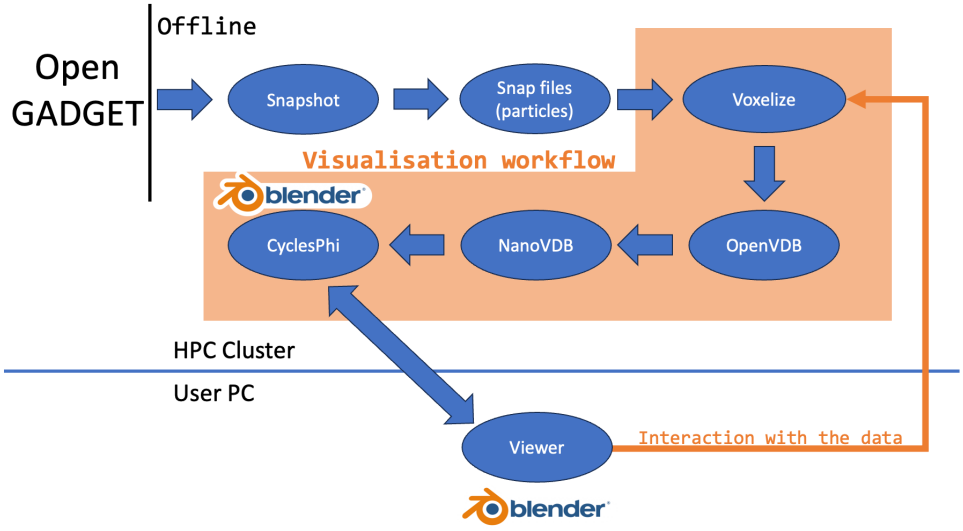

Figure 3: Proposed concept of interactive data visualization applied to data from OpenGADGET

The development of this use case will focus on OpenGADGET and its particle-based output data type. To exploit the volumetric format and consequent volumetric visualization, particles will be converted to voxels by

performing parallel voxelisation, similar to previous work for CFD data. There will be no further dependencies with OpenGADGET after the voxelisation stage, and the visualization approach will be general. Volumetric data will then be transferred to allocated GPUs running the path-tracing renderer, e.g. CyclesPhi. All of this processing will happen on an HPC cluster. From there, only the image data will be sent to a viewer on the user's PC. For the viewer, the Blender software will be leveraged by adapting its environment, given the possibility to extend its functionality via Python addons. This will result in a user-friendly, data-minimalistic and interactive solution for data visualization.

IT4I plans to use the open-source C++ library OpenVDB (1) with NanoVDB (8) for the sparse volumetric format. The OpenVDB (1) format is already an industry standard. It will allow universal use among large collections of rendering and visualization software other than Blender. The NanoVDB format is optimised for GPU usage, and therefore, it is an ideal solution for interactive volume rendering of cosmological data.

As a complementary line of work, the visualization Group of the BSC plans to investigate the use of OpenVDB (1) for the visualization of A&C simulation codes in offline pipelines aimed towards the production of high-impact visualizations for dissemination purposes. Effective pipelines of data conversion from scientific formats to OpenVDB (1) can be applied to a variety of movie-industry design softwares (e.g., Blender, SideFX Houdini, Autodesk Maya and 3DS Max, etc) and render engines (e.g., Cycles, Redshift, Arnold, Karma, etc.) to produce high-quality cinematic imagery. As opposed to an interactive visualization solution, an offline render pipeline sacrifices the interactive data exploration capabilities, but provides the highest possible render quality of offline rendering. Along with the main purpose of this use case, the development of an interactive cinematic visualization tool for cosmological simulations, we will investigate the application of the OpenVDB (1) format as an exchange format, to efficient, high-quality production of visualizations of the A&C codes in the SPACE project.

Functional requirements The main purpose of data visualization is to help users understand, interpret, and extract insights from the information presented. Under specific conditions, such as multidimensional datasets, these goals are more effectively achieved through interactive visualizations where users can perform transformations to the visual representations, such as faceting and zoom/pan, or even change the data source through filtering, annotation or aggregation. With such aspects in mind, it will be required to provide the following:

- High-fidelity visualization of different cosmological problems
- Interactive exploration of the data through basic camera operations (zoom, pan, roll)
- User-friendly interface for advanced interaction with the data (slice, filter, control color and transparency, etc.)

Future applicability to other codes As stated above and similarly to the previous use case, 4.3.1, the proposed workflow poses a flexible and modular implementation that should permit its adaptation to other similar SPACE A&C codes with a reasonable effort. In this case, it will require to adapt the data reader to the output of the simulation code.

4.3 Interactive, distributed visualization for HPC and cloud environments

Lead WP3 partner: INAF

Statement of the problem SPACE codes are expected to produce massively large data volumes (in the order of petabytes) executed on high performance computers. Such data volumes pose significant challenges for storage, access and data analysis. The visual exploration of big datasets, as one aspect of data analysis, pose some critical challenges as well, specifically: (i) Interactivity. The ability to deal with datasets exceeding the local machine's memory capacity; for complex visualizations the relevant computations should be performed close to the data to avoid time consuming streaming of large data volumes; (ii) Integration with the scientists' toolkit for seamless usage, abstracting from technical details related to the underlying HPC resources and freeing scientists to concentrate in scientific research; (iii) Facilitate visualization, processing and data analysis in a collaborative manner within, e.g., science gateway technologies to allow collaborative activity between users and

provide customization and scalability of data analysis/processing workflows, hiding underlying technicalities.

INAF Astrophysical Observatory of Catania has been developing and maintaining the Visualization Interface for the Virtual Observatory (VisIVO)[1] since 2005. VisIVO has recently been extended with the ViaLactea Visual Analytic modules. VisIVO is developed adopting the Virtual Observatory standards[2] and its main objective is to perform 3D and multi-dimensional data analysis and knowledge discovery of a priori unknown relationships between multivariate and complex astrophysical datasets.

To produce a visualization, VisIVO typically requires three steps: data importing, filtering, and visualizing. The importing process converts the supplied datasets (originally in heterogeneous formats) into an internal binary format. A VisIVO Binary Table (VBT) is a highly-efficient data representation used by VisIVO Server internally. A VBT is composed by a header file (extension .bin.head) containing all necessary metadata, and a raw data file (extension .bin) storing actual data values. The header may contain information about the overall number of fields and number of points for each field for point datasets, or the number of cells and relevant mesh sizes for volume datasets. The raw data file is typically a sequence of values, e.g., all X followed by all Y values. The filtering process, performed with the VisIVO Filter tool, allows to perform several operations on the data, this may include randomization or decimation to reduce the final resolution, mathematical or statistical operators or commonly adopted cosmological post-processing analysis such as the three commonly used mass assignment functions, i.e., the nearest grid point (NGP), the cloud-in-cell (CIC), and the triangular-shaped cloud (TSC) methods. Finally, the visualization process creates multi-dimensional views from the data that must fit the available RAM. The types of visualizations available include points, volumes and vectors, and are based on the visualization Toolkit (VTK (7)). Figure 4 depicts the typical visualization pipeline of VisIVO Server consisting on the application of the three main modules: VisIVO Importer, one or more VisIVO Filter(s), and one or more VisIVO Viewers.

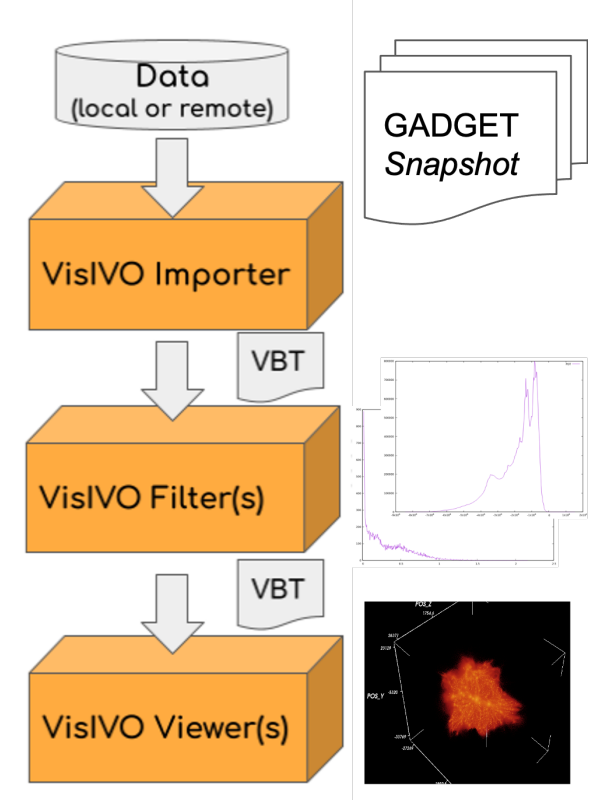

Figure 4: Typical visualization pipeline of VisIVO Server consisting of the application of the three main modules on GADGET snapshots.

[1] VisIVO, https://visivo.readthedocs.io/
[2] https://www.ivoa.net/documents/

Proposed solution VisIVO has already been deployed using Science Gateways to access distributed computing infrastructures (including clusters, grids and clouds) using containerization and virtualization technologies. In this regard, it has been selected as one of the pilot applications deployed on the EOSCpilot infrastructure to demonstrate that the tools can be accessed using gateways and cloud platforms and it has been deployed on the European Open Science Cloud (EOSC), efficiently exploiting Cloud infrastructures and interactive notebooks applications. Figure 5 presents one aspect of the evolution of the current Cloud deployment prototype (see the left diagram) towards allowing workflow abstractions and integration on HPC clusters. This will exploit the parallelization of VisIVO modules to handle large volumes of data more efficiently.

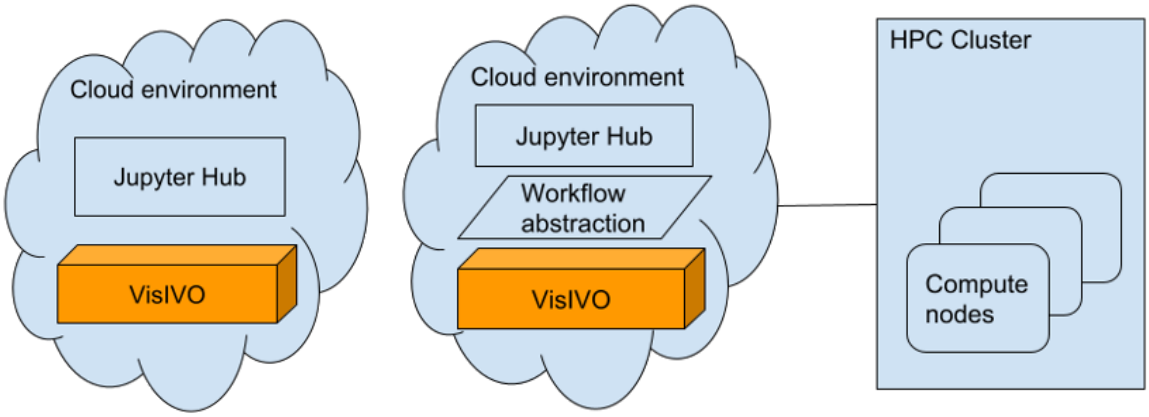

Figure 5: Evolution of the current Cloud deployment prototype (left diagram) toward allowing workflow abstractions and integrating HPC clusters (right diagram) to better exploit the parallelization of VisIVO modules thanks to the workflow abstraction offered by SPACE WP3 Task 3.1.

Workflow abstractions will allow a portable representation of the VisIVO modular applications and their resource requirements, fostering reproducibility and maintainability to take advantage of heterogeneous HPC facilities (including mixed HPC-Cloud resources) while minimizing data-movement overheads. In particular, we would like to integrate VisIVO with StreamFlow and Jupyter, a workflow provided by SPACE WP3 Task 3.1 partners.

Functional requirements The use case evolution and the related implementation activities will align with the following functional requirements:

- Enhance the portability of the VisIVO modular applications and their resource requirements. VisIVO modules, i.e., importer, filter(s) and viewer, are being parallelized to further exploit HPC and Exascale infrastructures. Improving its portability and potential of integration with other astrophysical pipelines and computing resources will make VisIVO fully integrated within the scientists' toolkit for its seamless usage, abstracting from technical details and freeing astronomers to concentrate in scientific research.
- Foster reproducibility and maintainability. Visualization-aided data analysis often requires several parameter settings for pre-processing, plus the actual rendering of complex multidimensional datasets. Furthermore, in this era of Open Science, offering novel mechanisms and techniques to make scientific discoveries reproducible and maintainable is a must, especially for enhancing scientific and technical collaborations.
- Take advantage of a more flexible resource exploitation over heterogeneous HPC facilities, including also mixed HPC-Cloud resources. Because of the increasing size and complexity of astrophysical datasets, there is a need for increasing the computing performances as well as the storage capacities for processing and analysis tasks while maintaining the cloud software-as-a-service opportunities.
- Minimize data-movement overheads and improve I/O performances. The importer modules of VisIVO rely on heavy I/O tasks for translating the astrophysical datasets to the internal VisIVO binary format which is used by the VisIVO filtering and visualization modules. Therefore, minimizing the computing costs of these I/O tasks could potentially improve the overall performance of the VisIVO pipelines.

Future applicability to other codes This use case will be developed primarily for OpenGADGET. Still, we expect to apply a similar methodology to other simulation codes of the SPACE CoE such as RAMSES (to compare and analyse different cosmological hydrodynamic simulations) and PLUTO (to exploit its output data format, which is fully compatible with VisIVO's underlying technology, i.e. VTK (7)).

This document is Public (PU)
and was produced under SPACE project EU GA 101093441

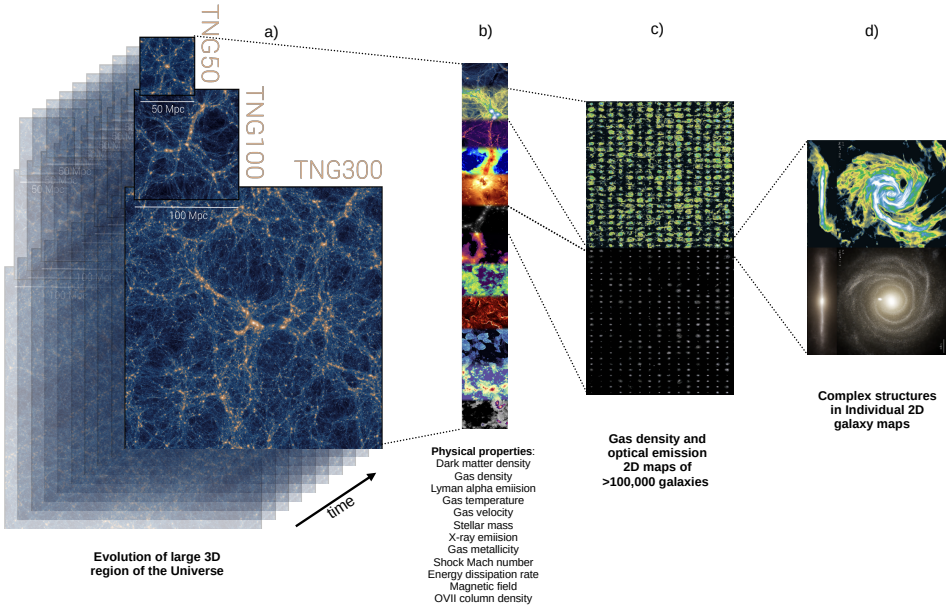

Figure 6: Schematic view of a small slice of the data generated by a typical large cosmological simulation from 2019 (in this case from the IllustrisTNG project), highlighting how the structures of interest inhabit a high-dimensional space that is difficult to explore, visualize, and analyze. From left to right, the images show the hierarchy of views of a single simulation run, from the entire time-evolving content of the full box (a), to slices through several of the dozens of dimensions corresponding to physical variables of the particles (b), to 2D gas density and optical emission maps of a small subsample of galaxies ($\sim 0.2\%$ of the entire dataset; c), to the detailed structure of a single 3D object (d). For this type of standard representation, the 6D phase space structure of the point clouds representing each galaxy must be collapsed to 2D maps along specific directions (individual images courtesy of the TNG Collaboration).

5 Machine Learning Use Cases

We describe the three main identified use cases for Machine Learning methods below.

5.1 Representation Learning for Explorative Knowledge Discovery in Simulations

Lead WP3 partner: H-ITS

Statement of the Problem Cosmological hydrodynamical simulations are excellent numerical laboratories for understanding the formation of galaxies and large-scale structure. They provide a highly detailed realization of structures in the Universe across a vast range of spatial and temporal scales, from shortly after the Big Bang until the present day. The simulation outputs typically consist of more than one hundred snapshots of the evolution of a large cubical representative region of the Universe containing all its main matter components: dark matter, gas, stars, and black holes. These components are discretized in mass (using particles), volume (using grid cells), or a combination of both. Each of these simulation snapshots is information-rich, including the 6D phase-space positions and velocities, as well as dozens of physical properties (i.e. fields or channels) for all components (e.g. gas density, temperature, metallicity, individual element abundances, etc.; see Fig. 6).

State-of-the-art large-volume cosmological simulations currently model the evolution of the Universe using more than 10^{11} particles (see Fig. 7), allowing the formation of structures to be followed in detail across a dynamic range of ~ 7 orders of magnitude in both space and time. Their computational cost is extremely high, currently on the order of 100M CPU hours, and their output datasets are measured in petabytes. This level of data size and complexity already far exceeds the exploration, synthesis, and interpretation capacity of humans. Moreover, architectures and I/O formats vary greatly across cosmological codes, imposing a barrier to applying code-specific tools more generally to all cosmological datasets. In the Exascale era, simulations are expected to

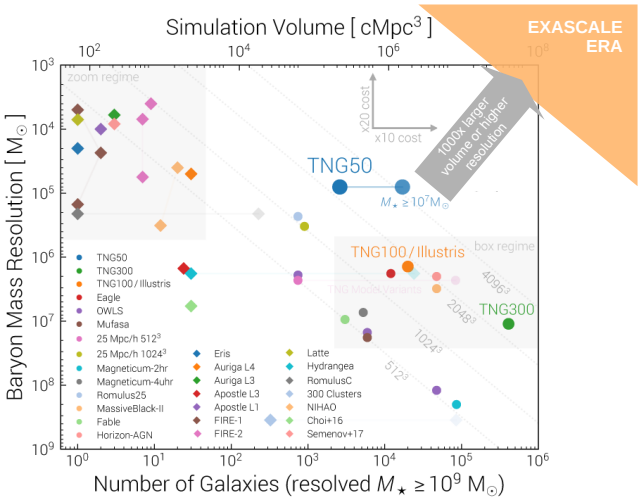

Figure 7: The extreme data challenge posed by the exploitation of cosmological simulations in the Exascale era. The largest simulations today model the detailed evolution of galaxies and the cosmic web in a representative volume of the Universe over cosmic timescales. This requires following more than 10^5 objects represented by $\sim 10^{11}$ resolution elements (particles and grid cells) across hundreds of individual snapshots in time, generating petabytes of data. Exascale simulations will increase these numbers by a factor of one thousand. Their high computational cost will require efficient extraction of the large amount of information that they contain. Our exploration framework will be designed to work with very large datasets containing millions of objects, well beyond what is currently possible (indicated by the orange shaded area). Figure adapted from (2).

increase in size (or resolution) by a factor of at least $\sim 10^3$, rendering the traditional human- and code-centered simulation exploration and analysis techniques obsolete.

The approach to the problem of increasing simulation data size and complexity has changed very little over the past three decades since cosmological simulations became widespread tools. Simulation data is typically represented in catalogs by first collapsing the rich multidimensional data onto a simplified zero-dimensional representation (i.e. scalar values) of galaxy and Dark Matter halo properties. Examples of these representations include stellar mass, morphology, half-light radius, mean surface brightness, bulge-to-disk ratio, hydrogen mass, gas metallicity, halo concentration, and maximum circular velocity, among many others. This approach was inspired by the data scarcity of observational astronomy, where in the case of large galaxy surveys it is much more efficient to measure and analyze relationships between global galaxy properties (e.g. the Hubble diagram, the galaxy main sequence, the mass-metallicity relation, the Tully-Fisher relation, the fundamental plane, etc.). The same approach was used for the 'extrinsic' causes of these properties that are only available in simulations, including halo mass and shape, environment, or assembly history. These properties may be expressed in 1D (e.g. as radial profiles of the mass components, etc.) for a more fine-grained analysis of these structures. These choices are based on simplifications assuming various symmetries necessary for analytical and semi-analytical models of galaxy populations. All the simplifications above are no longer relevant in hydrodynamical simulations, where the evolution of the 3D structure of galaxies and the cosmic web can be followed in detail from their formation until the present day. The collapse of structures from $>$ 3D to a single number or a 1D vector wastes most of the detailed morphological data and potentially removes valuable information on the physics behind the formation of galaxies and large-scale structure.

In addition to addressing the challenge of interpretability, if cosmological simulations are to adopt and follow the 'FAIR' data principles, they should make their output data easily findable, accessible, and interoperable. Our approach aims to address these additional requirements by enabling users to find existing datasets, interactively explore them, compare them, and directly access their raw and post-processed data products.

Proposed Solution Representation Learning is a well-developed area of unsupervised machine learning that aims to learn compact representations of complex high-dimensional data that efficiently compress the informa-

tion with the goal of allowing easier visualization, exploration, and interpretation. This class of models offers a powerful way to understand the intrinsic distribution of large datasets and to sample from it to generate new data points. They can be trained using raw data, avoiding the costly need to produce labels.

Our proposed solution for this use case consists of a self-contained tool that takes the raw simulation data as input, performs all the preprocessing under the hood, and trains a model to learn a compact interpretable representation of user-selected structures, such as galaxies, in arbitrary physical components. For this, instead of collapsing the galaxies to single scalars or 1D representations along arbitrary projections guided only by human intuition, the concept will use a Generative Deep Learning algorithm to learn the most efficient representation of simulated cosmological structures (and in particular galaxies) in a low-dimensional latent space. The latent space dimensionality can be selected by the user and is guided by the requirement that it can be easily visualized and explored by a human. This guarantees that the data can be inspected easily and interactively, even for the largest cosmological simulations of the Exascale era. The galaxies in this reduced representation space can then be further painted with the traditional global (scalar) properties to aid interpretation and enable knowledge discovery. They may also be colored and annotated using latent representations of the extrinsic variables (like formation history or environment) to find and investigate the causal drivers of observed galaxy properties. The visualization is lightweight such that it can run on any laptop via a web server, and it provides the functionality to interactively inspect and select subsamples of data for local analysis.

Understanding the intrinsic structure of galaxies and the physical processes that determine it is an active area of current research within both observational and theoretical Astrophysics, highlighting the potential applications of Generative Deep Learning. In addition to learning to represent the traditional 2D maps of simulated galaxies (which originated from the need to produce synthetic observations to compare with real data), we plan to extend our concept to learn galaxies' full 3D (or even 6D phase-space) structure. Geometric Deep Learning is a broad class of methods designed to incorporate knowledge of geometrical constraints in the learning process. Graph Neural Networks (GNNs) are an example of this type of algorithm that can be used to process irregular and sparse data that is represented by graphs. They efficiently capture the graph structure of data, which is often very rich (and computationally prohibitive for standard Deep Learning), and are particularly suitable for learning global permutation-invariant quantities. Because of their defining characteristics, GNNs can be applied to any astrophysical data characterized by point clouds. These properties make GNNs ideal tools for describing the $>$ 3D structure of galaxies, dark matter halos, and large-scale structures in cosmological simulations. Their potential has been demonstrated for cosmological analyses of the large-scale matter distribution (i.e. the cosmic web), which traditionally focus on two-point (or higher-order) statistics that can only exploit a subset of the available information content, and are computationally expensive. Using GNNs, the properties of the cosmic web can be extracted without the information loss associated with summary statistics. They can also be used to directly compare the clustering properties of simulations with existing observational galaxy catalogs, allowing likelihood-free inference of cosmological parameters. We aim to employ Geometric Deep Learning to describe galaxies and dark matter halos in their native dimensionality, using them as inputs to learn compact, explorable, and interpretable representations of their structure as predicted by cosmological simulations.

Functional Requirements With few exceptions, the largest state-of-the-art simulation projects struggle to provide easy access to their data, and even if these are available to download from a cloud database, there is a very high barrier to explorability and interpretability for anyone not directly associated with the projects. This barrier results from the adoption of very heterogeneous data formats and standards by the major codes, and by the development of code-specific postprocessing and analysis tools that are often also not findable and accessible to outsiders. Overcoming these two major obstacles will require a solution that:

1. exploits the statistical power of Unsupervised Machine Learning, and specifically generative models, to learn and create interpretable data representations that provide simple interactive exploration of entire simulation datasets, and
2. does so in a code-agnostic way by operating on a universal representation of the simulation data that is user-friendly (i.e. as matter components with physical attributes in standard unit systems), independent of the I/O requirements, and requires no expertise in the simulation codes.

Achieving these goals will require the development of a workflow supported by a software toolkit with the following general requirements and characteristics:

- It must include utilities to generate a compact, low-dimensional representation of galaxy structure for *all* the galaxies in a given simulation dataset using arbitrary user-selected physical properties. The representation should be easy to visualize and explore on a computer screen with minimal local computational resources.
- The tool should automatically handle the selection and preprocessing of the simulation and training data, with minimum user input. It should be flexible enough to accommodate the user's needs and must include options for using 2D (i.e. projected maps or images), or 3D or even higher dimensional (e.g. 6D) point cloud representations of the simulated galaxies in an arbitrary number of channels (i.e. physical properties like gas or stellar density).
- Using the trained model, the tool should efficiently render an interactive visualization of the entire content of the dataset on the bounded latent space. This representation must enable the visualization of very large galaxy catalogs by grouping nearby objects into tiles that display only the representative objects at each level of the hierarchy, with the highest level displaying the entire dataset.
- The tool must enable interactive visualization of the hierarchical representation for both the input data and the model reconstruction at every point on the sphere including panning and hierarchical zoom on specific regions to examine different classes of objects.
- The functionality of the tool should be first demonstrated using the largest simulation datasets currently available.
- For its applicability to Exascale cosmological simulations, it is essential that the performance of the software scales well with dataset size, up to 10^8 galaxies and even beyond.

Future applicability to other codes We aim to develop the representation learning and analysis frameworks to be able to agnostically ingest and represent the data from any large cosmological simulation of the Exascale era, regardless of the simulation code used to produce the output and the technical details of the simulation. This effectively removes the requirement for the user to have in-depth technical knowledge about specific codes and data formats. The initial prototype will be developed to work with a single code for testing purposes, but the long-term goal is to include wrappers to translate the output of all codes involved in the project to a common abstract and user-interpretable input data format. The tool will have applicability to all codes that are currently used for cosmological hydrodynamic simulations, namely RAMSES, OpenGADGET, and Changa/Gasoline.

Even beyond the broad applicability to most current cosmological simulation codes, we expect that the tool will be useful more generally to analyze data from simulations that model the time evolution of a fluid in a fixed volume of space. The tool could be applied to many of these codes with the aim to identify and characterize specific emergent phenomena, such as turbulent and magnetic field structures and shocks in astrophysical fluid dynamics simulations with the PLUTO code, or electron flows in plasma simulations with the iPic3D code.

5.2 Physics-informed reconstruction using synthetic observations of cosmological simulations

Lead WP3 partner: BSC

Statement of the problem One of the main goals of cosmology is to understand how the diversity of complex structures that we observe today, from the smallest to the largest scales, emerged over cosmic timescales from tiny initial fluctuations in the early Universe. Observations of the early Universe have provided detailed knowledge of the initial conditions for structure formation, while observations of nearby galaxies show the end result of this process. However, observations provide only a very limited 2D view of these structures, severely limiting our ability to reconstruct their physical properties and formation processes. Simulations have provided an unparalleled statistical overview of the complex physical processes that shape galaxies, but it is not yet possible for these models to make predictions for specific objects. This is because we do not know how to connect the initial conditions to the galaxies they give rise to.

This document is Public (PU)
and was produced under SPACE project EU GA 101093441

This type of inverse problem, where the goal is to reconstruct the physical properties (e.g. structure and history) of an object based on limited information, is highly degenerate due to the loss of information in the observation (forward) process. This makes its solution formally intractable. This has been traditionally addressed by approximating galaxies using extremely simplified models with only a few components, and then performing parameter inference by fitting the simple models to the observational data. This method is limited by the numerous simplifying assumptions and by the ambiguity in the choice of number of components (or parameters) that determine its capacity to fit the data. A hybrid approach that combines the predictive power of large-volume hydrodynamical cosmological simulations and the increasing amount of observational data may offer a promising solution to these limitations. This requires the flexibility of state-of-the-art deep learning methods to learn complex high-dimensional mappings between physical properties and synthetic observations.

This use case will start working on a very specific scientific question which aims to close this gap between observational and simulation methods. It focuses on the measurement of the Intra Cluster Light (ICL), which is the faint and diffuse light observed in the vast spaces between galaxies within galaxy clusters. In principle it is possible to determine the ICL distribution in a cluster from observation data, given that several assumptions on the light profile are made. However, results dramatically differ for slight variations of the assumptions taken, which highlights the importance of identifying ICL by dynamical means both in simulation and observational data. To achieve this, and as a first step, the methodology will utilize a line-of-sight velocity dispersion map (available in observational data) to compute the ICL distribution using deep learning. This approach offers a dynamic and robust means of characterizing the ICL, providing a more consistent representation of its distribution within galaxy clusters. Once this is validated, we will explore the exportation of the same methodology to other domains, always attempting to reconstruct galaxy/cluster physical properties such as mass, accreted fraction, merger histories and trees from observable variables.

Proposed solution The increase in computational power expected in the Exascale computing era will enable cosmological simulations to include more important physical processes at higher resolution, as well as modelling larger volumes and more diverse galaxy populations. The task of using the simulations to reconstruct galaxy properties requires a universal machine learning tool that can read the raw simulation data and produce synthetic maps of observables such as multi-band images, velocity moment maps, metallicity maps, and even position-position-velocity datacubes for the stellar and cold gas components, among many others. The tool should offer the flexibility to select among different deep learning methods (specialized for handling image data) depending on the specific reconstruction task. It would allow the user to select the input maps (i.e. the features) and the desired output reconstruction, including physical properties such as mass, 3D structure, progenitor properties, origin and mass fraction of each component, merger tree, etc. Once trained on the millions of galaxies in the simulation, the tool can be applied to observational data to obtain the target properties of real galaxies.

To advance towards the development of such a general tool, we will follow a modular approach, starting by focusing on the reconstruction of a specific set of galaxy/cluster physical properties (i.e: ICL) from different maps of observables. This will constitute a first prototype, which will function as a standard deep learning model. As input we will feed an observable and as output we expect the reconstructed cluster property. Then, following the same procedure we will extend it to the reconstruction of other galaxy/cluster physical properties such as mass, accreted fraction, merger histories or trees from maps of observables. Thus, this second prototype will allow users to select the simulation or physics model, the objects (galaxies or clusters), the observables (maps and vectors/scalars), and the regression variables they want to infer (maps of ICL fraction, merger trees, etc).

Functional requirements The methodology, based on previous work, consists in the following pipeline: First, a sample of clusters among the most massive are selected from a zoom-in cosmological hydrodynamical simulation. Then, mock images of the galaxy clusters are created, which mimic the observational conditions of a hypothetical observation by a state-of-art integral field spectrograph (see Figure 8). Finally, data augmentation is performed on the original dataset, plus some treatment of contaminants (i.e: interlopers) by masking substructures in the images. The resulting dataset is composed of line-of-sight velocity dispersion and ICL mass fraction maps, which are then divided into training, validation and test sets.

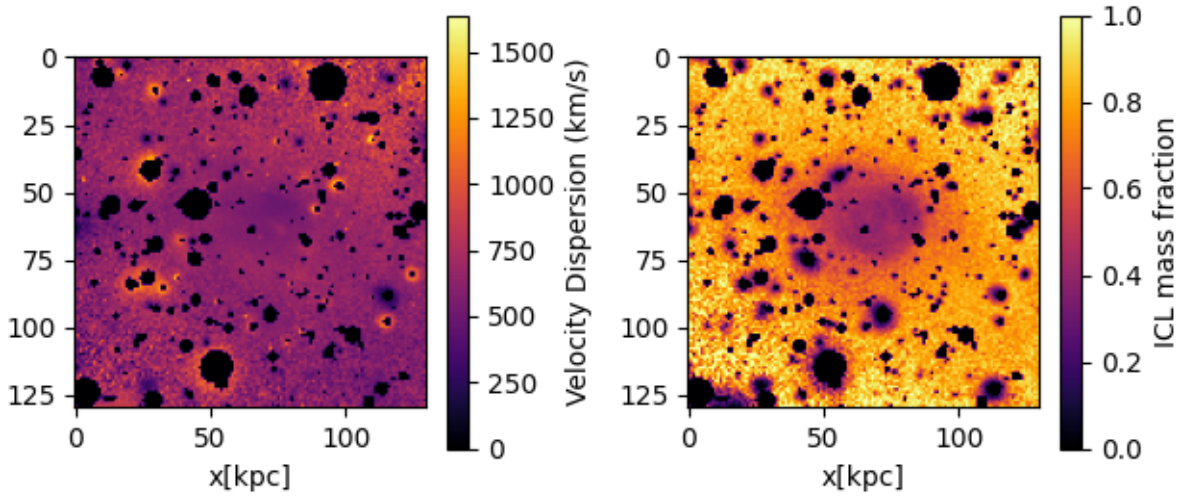

Figure 8: Examples of the mock images. In the left panel, there is the velocity dispersion map, and in the right panel the ICL mass fraction from a randomly extracted halo.

The next step is the training of deep learning models on this data. Following the use-case's previous work, we begin with the U-Net and Attention U-Net architectures, which are a common choice for pixel-wise regression and which we will use as a baseline. Then we will test against this baseline other architectures and more modern techniques such as Generative Adversarial Networks (GANs), autoencoders, and diffusion models. The goal for the first prototype is to improve on the results of the use-case's previous work to obtain a model capable of reconstructing ICL maps with high accuracy from the velocity dispersion maps. The limitation is the amount of data that we have at out disposal, for only a handful of the structures we are interested in (massive clusters) appear on each simulation run. To deal with this we will produce mock images from different positions and use other data augmentation techniques. We will also put a focus on making the predictions of the regression models robust to the possibly large uncertainties in the simulations.

Once this is validated, we will generalize this methodology to reconstructing other galaxy/cluster physical properties such as mass, accreted fraction, merger histories and trees from maps of observables.

Future applicability to other codes The initial prototype will be developed to work with simulation data coming from OpenGADGET. However, we expect most of the methods developed to have applicability to all codes that are currently used for cosmological hydrodynamic simulations, namely RAMSES and Changa/Gasoline, given that the training will be performed mainly with mock images and that standard codes to produce mock images can be adapted to other codes with small modifications.

All models we develop will be made publicly available. By working with containers we will ensure that the results are reusable and reproducible, and that they can be integrated with other tools.

5.3 Predicting the effect of radiation on gas cooling rates on-the-fly in hydrodynamical simulations

Lead WP3 partner: BSC

Statement of the problem Numerical methods based on High Performance Computing (HPC) have now established themselves as central tools for modelling complex systems. This is particularly the case in the

fields of astrophysics and cosmology. These methods are now the essential tool for theoretical laboratories that enable scientists to investigate, interpret and ultimately understand the physical processes in our observed universe, laying the foundation for new, spectacular scientific discoveries. For these numerical experiments and data evaluations, efficient and effective use of extreme computing power, in the Exascale range, is essential. In particular, sub-grid models for cooling and star formation are among the most complex sub-areas needed to correctly describe astrophysical systems. The cooling function can be very complex as it depends on various local properties of the gas (such as temperature, density, chemical composition, shape and strength of the local and global radiation fields). Since the calculation of the energy loss rate involves very complex physical processes, it is necessary to use pre-calculated values. However, this calculation needs to be further simplified because interpolation in the large parameter space within the simulation is too data intensive. Here, new interpolation methods as well as ML methods open the door to make highly precise cooling functions available for cosmological simulations, where the complete local description of the gas is used to evaluate the energy loss processes.

Machine learning methods have recently been used in the literature to estimate cooling rates with promising results. However, these studies used actual hydrodynamic simulation data as training data sets to investigate how cooling rates depend on various parameters such as gas temperature, density, metallicity and radiation fields. While this method significantly reduces the amount of training data, it can be problematic when the gas cells/particles in the simulation do not sample the parameter space well. For example, gas at the cooling peak (e.g., shock/feedback heated gas at temperature $\sim 10^5$K and relatively high density $n > 1$ cm^{-3}) often lives extremely shortly in the density-temperature parameter space before it gets cooled to a different temperature, thus sampled very poorly in actual simulation data. As a result, the neural network trained from one snapshot can be highly specific and may not be accurate enough for a different time step, which implies that one needs to always run the simulation first for the training, then run it again to incorporate the more accurate cooling function. This is numerically expensive and becomes infeasible when applied to Exascale cosmological simulations.

A more straightforward way is to train neural networks with pre-calculated tables from radiative transfer codes such as CLOUDY. Such tables usually cover the parameter space reasonably well. For cosmological simulations without radiative transfer, the cooling table is often only a function of gas temperature, density and redshift (to capture the uniform extragalactic radiation field), and thus relatively small in size. The standard practice is that the metal cooling and heating rate of the gas is obtained by simply interpolating CLOUDY tables during runtime. However, the inclusion of local radiation field in modern radiative transfer simulations increases the dimentionality tremendously in order to capture the intensity and shape of the radiation field which varies significantly both spatially and temporally. A uniform grid CLOUDY table is not only memory intensive but also too computing intensive to be interpolated on the fly. Even with ML methods, such training data set may still be too large to have feasible training time. One way to mitigate this problem is to create unstructured data points for sampling the cooling function, instead of using a uniform grid table. This can reduce the size of the training set significantly. One of the science partners within the SPACE CoE has recently developed such tool (CHIPS: Cloudy based Heuristic and Interactive Parameterspace Sampler,https://github.com/Vetinar1/CHIPS), which can be applied to create an optimal unstructured training data set for ML methods. The lead group of this use case, the Large-scale Computational Fluid Dynamics group of the BSC will investigate the use of training sets generated by CHIPS in implementing a run-time, ML-based pipeline for gas cooling prediction.

Proposed solution Previous experience of the BSC team is in the integration of CFD and ML to accelerate the prediction of urban air flows. In that sense, ML methodologies exploiting high-fidelity CFD simulation data have been explored to help satisfy the need for fast flow predictions in various applications, including urban flow prediction whose physics are similar to those in the present use case. In previous projects, we have used deep neural networks to provide real-time and reliable predictions for urban emission dispersion of contaminants. The proposed strategy was based on two sequentially coupled DNNs which covered the necessary steps to provide the pollutant concentration and dispersion evolution prediction.

Following this previous experience, the activities in SPACE will cover several research areas, ranging from the numerical algorithms to the software interface between the A&C codes and ML libraries. Indeed, it is non-trivial to efficiently couple A&C solvers developed using low-level compiled languages, e.g., Fortran, C/C++, with ML libraries based on high-level interpreted languages such as Python. While this can be accomplished through Unix sockets or message-passing interface (MPI), in the current setup we will employ the SmartSim

library (https://github.com/CrayLabs/SmartSim). SmartSim allows communication between processes through an in-memory Redis database with minimal overhead and, compared to the previously mentioned approaches, it lowers the software complexity of the framework. The novel algorithms will be assessed by computing basic test cases, but representative of challenging A&C problems, and the performance of the ML-enhanced methods will be compared to classical and well-established approaches from the literature. The capabilities of the new and highly scalable A&C-ML coupling strategies will be tested on several platforms, including massively parallel hybrid CPU/GPU architectures. With the use of SmartSim we can exploit the power of accelerated HPC architectures even when the simulation code runs in a CPU-based cluster. Hence, we can use graphics-processing units (GPUs) to efficiently exploit ML algorithms, while the A&C can run traditionally using MPI and the central-processing unit (CPU) part of a cluster node.

As a first step, we will implement this solution to run on run-time with the A&C code Changa. This will involve:

- identify suitable parts of the code to be accelerated by the run-time ML modules.
- Generate suitable training databases. As a starting point, we will use CHIPS.
- Select the best suited CNN architecture based on the requirements of the code.

Functional requirements

- Accuracy: must be significantly more accurate than existing tabulated cooling estimators or analytical functions, and should approximate reasonable accurately the CLOUDY results (used as validation data).
- The method must satisfy the following computational constraints: 1) it must be extremely fast to execute due to the need to call it at every time step of the simulation, for millions of calls. 2) it must use limited space in memory to allow execution on current HPC distributed memory architectures. 3) it must be competitive with traditional approaches, including tabulated estimators (9; 10; 11) or analytical functions (e.g. 12) which sacrifice accuracy in favor of computational efficiency.

Future applicability to other codes This use case will be developed based on the Changa simulation code. Since it is devised to be a run-time implementation, given the characteristics of the problem, the adaptability of the resulting tool to other simulation codes might not be as straight-forward as with the other use cases mentioned in this document. Porting the framework to other simulation codes will involve changes to the codes, plus identifying the parts suitable for replacement or adaptation. However, given the similarities between Changa and OpenGADGET, a future implementation of the tool in the latter involving reasonable effort might be possible and will be assessed during the development of the framework.

This document is Public (PU)
and was produced under SPACE project EU GA 101093441

6 Workflows Use Cases

6.1 Topology-aware workflows for modular Exascale data analysis

Lead WP3 partner: UNITO

Statement of the problem This use case will develop coordination semantics to design modular HPC applications for the next generation Exascale machines. This approach will allow a topology-aware co-scheduling of loosely-coupled components of the complex Exascale data analysis applications, allocating each component to the execution architecture that best fits its resource requirements, e.g. burst-buffers for I/O-bound operations, or hardware accelerators for high-performance matrix operations. In this setting, workflow abstractions allow a portable representation of modular applications in terms of atomic components, their resource requirements, and their mutual (data/control) dependencies, fostering reproducibility and maintainability. Still, conventional workflow models fail to capture the details of the underlying execution environment, leaving the mapping of steps onto processing elements entirely to the runtime scheduling layer.

Existing work or prototypes UNITO is working on topology-aware workflows, i.e. workflow graphs augmented with a high-level representation of the resource topology, to support more advanced orchestration techniques that take advantage of heterogeneous HPC facilities while minimising data-movement overheads. In particular, the StreamFlow framework (13) (https://streamflow.di.unito.it/) is a container-native Workflow Management System (WMS) based on the Common Workflow Language (CWL). It is founded on two main principles:

- it supports the concurrent execution of multiple communicating tasks in a multi-agent ecosystem by enabling the execution of tasks in multi-container environments;
- it empowers hybrid workflow executions on top of multi-cloud or hybrid Cloud-HPC infrastructures by relaxing the requirement of a single shared data space.

StreamFlow declaratively describes cross-application workflows with data dependencies, complex execution environments composed of heterogeneous and independent infrastructures, and mapping steps onto execution locations. The hybrid workflow approach enables the deployment of different, potentially distributed workflow steps (e.g., MPI, TensorFlow) onto different modules (e.g., HPC facilities, Cloud VMs, Kubernetes). StreamFlow allows the seamless integration of new modules and deployment methods through self-contained plugins. The topology awareness emerging from these workflow models allows StreamFlow to implement locality-based scheduling strategies, automated data transfers, and fault tolerance.

On Fig. 9, representing the StreamFlow logical stack, can be seen that it needs three different types of inputs: workflow descriptions, location topology descriptions and a StreamFlow file with the step-location mapping. Before the actual execution of a step, it is necessary to complete three different tasks:

- deploy the locations when needed, done by Deployment manager;
- select the best locations to execute each step while guaranteeing that all requirements are satisfied, performed by Scheduler; and
- assure that each location can access all the data dependencies required to complete the assigned job, performing the data transfer only when necessary, carried out by Data manager.

Exploiting the features of StreamFlow, this use-case will develop suitable coordination semantics to express (and optimise) parallel workflow patterns typically used to design HPC applications. Moreover, the StreamFlow (https://streamflow.di.unito.it/) framework, currently used in the EuroHPC ACROSS and EUPEX projects, will be extended to orchestrate modular HPC applications on Exascale architectures.

The other tool UNITO is working on, relying on StreamFlow workflow management system, is Jupyter workflow (14) (https://jupyter-workflow.di.unito.it). It is an extension of the IPython kernel designed to support distributed literate workflows. The Jupyter workflow kernel enables Jupyter Notebooks to describe complex workflows and execute them in a distributed fashion on Hybrid Cloud HPC infrastructures. In particular, code cells are regarded as the nodes of a distributed workflow graph. In contrast, cell metadata are used to express data and control dependencies, parallel execution patterns (e.g., Scatter/Gather), and target execution infrastructures (e.g., HPC facilities, Cloud VMs, Kubernetes). Relying on cell metadata to describe workflows has several significant advantages:

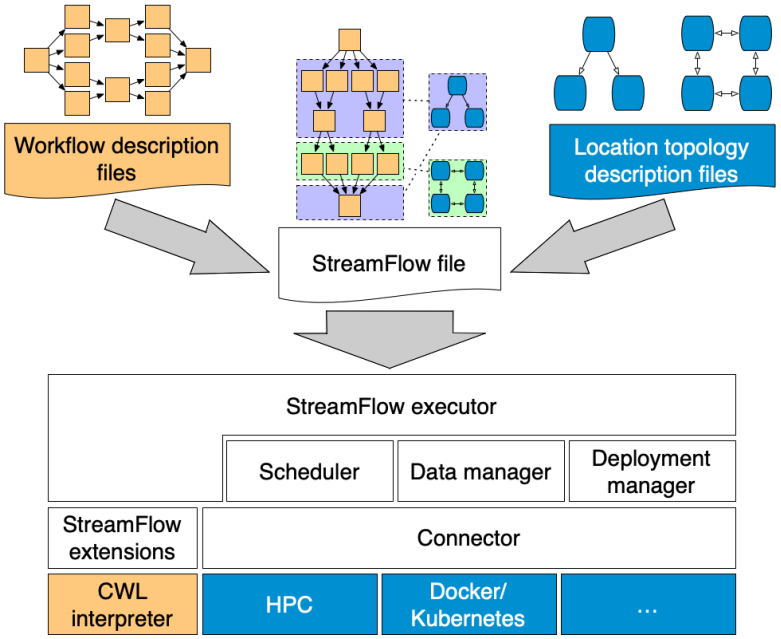

Figure 9: The StreamFlow logical stack.

- it maintains a clear separation between host and coordination semantics, improving the readability and maintainability of complex applications;
- it avoids technology lock-in: the same metadata format can be interpreted by different Jupyter kernels to support more languages (other than Python), specific execution architectures, or commercial software stacks; and
- it smooths the learning curve of stand-alone workflow systems. Users familiar with Jupyter Notebooks do not have to learn a new framework to scale their experiments.

Functional requirements The heterogeneity and complexity of modern applications force monolithic approaches to give way to modular architectures and patterns for designing and developing software, of which workflow models are first-class representatives. In the same way, heterogeneity in contemporary hardware resources (e.g., highly parallel hardware accelerators, low energy-consuming FPGAs, or application-specific quantum solvers) fosters modular approaches in designing execution environments for such applications. Hybrid workflows, depicted by StreamFlow framework, represent an essential methodological step in this direction and provide the following features:

- generalization of the concepts of portability and reproducibility from the application plane to the entire execution process.
- better cooperation between domain experts, who write the application logic, and computer scientists, who find the best execution environment for each workflow step according to specific requirements (e.g., in terms of cost, time-to-solution or energy consumption), obtained by the separation of concerns brought by hybrid workflows. Additionally, both of them are free from the burdens of managing applications deployment and life-cycle and writing explicit data transfer logic, enhancing productivity.
- automatic cross-stack executions of independent steps, providing a trivial way to offload tasks in urgent computing scenarios thanks to loose mapping relation between steps and locations.

Proposed solution In SPACE we plan to start with the integration between VisIVO and StreamFlow (see also Section 4.3). First, we intend to support the concurrent execution of multiple communicating tasks in a multi-agent ecosystem by executing the different VisIVO modules in multi-container environments. Second, we will be able to obtain hybrid workflow executions on top of multi-cloud or Hybrid Cloud HPC infrastructures.

The hybrid workflow approach will allow for the deployment of the different distributed VisIVO workflow steps into different modules. Additionally, it will exploit the topology awareness emerging from the VisIVO workflow models allowing StreamFlow to implement locality-based scheduling strategies, automated data transfers, and fault tolerance. Furthermore, we will increase the reproducibility and provenance of our VisIVO workflows thanks to the Common Workflow Language (CWL) used by StreamFlow and all its related platforms e.g. the workflow Hub and the RO-CRATE and supports also other workflow managers like Galaxy, Airflow.

Additionally, we plan to investigate the Jupyter workflow kernel to describe the VisIVO workflows and execute them in a distributed fashion on Hybrid Cloud HPC infrastructures aiming to improve the usability, readability and maintainability of VisIVO applications. Moreover, we expect this integration to improve the application scalability to better exploit the heterogeneity of the underlying computing resources.

Future applicability to other codes or other workflows for different use cases As the visualization and ML use cases explained in this document begin to be developed, we will interact with the development and research teams to identify opportunities for building workflows on parts or the whole processes.

7 Conclusions

The SPACE ML, visualization and workflow use cases and requirements reports on the process, methods, tools and results of the work performed in the identification of use cases and requirements for visualization, ML-assisted analysis and workflows. This work will be central to the future development of such tools for the duration of the project, as it provides the base layer of technical, non-technical and functional constraints, requirements, and needs of the code developers and users. While the core definitions of the use cases should remain the same, the requirements and technical characteristics of the data analysis solutions to be developed in WP3 might change and adapt in the development phase. This document will also be the basis of the future deliverables D.3.2."SPACE Visualization data analysis and workflow framework #1 Release" and D3.3."SPACE ML framework #1 Release", which will report about the work developed in each area until M30, as well as of the final reports D3.4."SPACE Visualization data analysis and workflow framework Final Release" and D3.5."SPACE ML framework Final Release", due in M48.

References

[1] K. Museth, "Vdb: High-resolution sparse volumes with dynamic topology," *ACM Trans. Graph.*, vol. 32, no. 3, jul 2013. [Online]. Available: https://doi.org/10.1145/2487228.2487235

[2] D. Nelson, A. Pillepich, V. Springel, R. Pakmor, R. Weinberger, S. Genel, P. Torrey, M. Vogelsberger, F. Marinacci, and L. Hernquist, "First results from the TNG50 simulation: galactic outflows driven by supernovae and black hole feedback," , vol. 490, no. 3, pp. 3234–3261, Dec. 2019.

[3] N. Marz and J. Warren, *Big Data: Principles and best practices of scalable real time data systems*. Manning Publications Co., 2015.

[4] "Blender," 2023. [Online]. Available: https://www.blender.org/

[5] "CyclesPhi," 2022. [Online]. Available: https://code.it4i.cz/raas/cyclesphi

[6] K. Borkiewicz, A. J. Christensen, R. Wyatt, and E. T. Wright, "Introduction to cinematic scientific visualization," in *ACM SIGGRAPH 2020 Courses*, ser. SIGGRAPH '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3388769.3407502

[7] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit (4th ed.)*. Kitware, 2006.

[8] K. Museth, "Nanovdb: A gpu-friendly and portable vdb data structure for real-time rendering and simulation," in *ACM SIGGRAPH 2021 Talks*, ser. SIGGRAPH '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3450623.3464653

[9] R. P. C. Wiersma, J. Schaye, and B. D. Smith, "The effect of photoionization on the cooling rates of enriched, astrophysical plasmas," , vol. 393, no. 1, pp. 99–107, Feb. 2009.

[10] B. D. Smith, G. L. Bryan, S. C. O. Glover, N. J. Goldbaum, M. J. Turk, J. Regan, J. H. Wise, H.-Y. Schive, T. Abel, A. Emerick, B. W. O'Shea, P. Anninos, C. B. Hummels, and S. Khochfar, "GRACKLE: a chemistry and cooling library for astrophysics," , vol. 466, no. 2, pp. 2217–2234, Apr. 2017.

[11] N. Y. Gnedin and N. Hollon, "Cooling and Heating Functions of Photoionized Gas," , vol. 202, no. 2, p. 13, Oct. 2012.

[12] A. Rosen and J. N. Bregman, "Global Models of the Interstellar Medium in Disk Galaxies," , vol. 440, p. 634, Feb. 1995.

[13] I. Colonnelli, B. Cantalupo, I. Merelli, and M. Aldinucci, "StreamFlow: cross-breeding cloud with HPC," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 4, pp. 1723–1737, 2021. [Online]. Available: https://doi.org/10.1109/TETC.2020.3019202

[14] I. Colonnelli, M. Aldinucci, B. Cantalupo, L. Padovani, S. Rabellino, C. Spampinato, R. Morelli, R. D. Carlo, N. Magini, and C. Cavazzoni, "Distributed workflows with jupyter," *Future Gener. Comput. Syst.*, vol. 128, pp. 282–298, 2022. [Online]. Available: https://doi.org/10.1016/j.future.2021.10.007

A Appendix I

The template document to collect Input/Output information of each WP1 exascale application is shown in the following page.

OUTPUT #1								
NOTES:								
Effective bytes per element								
FORMAT								
CONFIGURABLE								
I/O INTERFACE								
CASE 1	***lower bound***	***upper bound***						
Nr. of elements								
Nr. of files per set								
Nr. of sets								
Size per set (GB)		**Total size (GB)**		**Toal Nr of files**		**Avg File size (GB)**		**I/O Rate (GB / hour)**
lower bound	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	
0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	#DIV/0!	#DIV/0!	
CASE 2	***lower bound***	***upper bound***						
Nr. of elements								
Nr. of files per set								
Nr. of sets								
Size per set (GB)		**Total size (GB)**		**Toal Nr of files**		**Avg File size (GB)**		**I/O Rate (GB / hour)**
lower bound	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	
0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	#DIV/0!	#DIV/0!	
CASE 3	***lower bound***	***upper bound***						
Nr. of elements								
Nr. of files per set								
Nr. of sets								
Size per set (GB)		**Total size (GB)**		**Toal Nr of files**		**Avg File size (GB)**		**I/O Rate (GB / hour)**
lower bound	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	
0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	#DIV/0!	#DIV/0!	

OUTPUT #2								
NOTES:								
Effective bytes per element								
FORMAT								
CONFIGURABLE								
I/O INTERFACE								
CASE 1	***lower bound***	***upper bound***						
Nr. of elements								
Nr. of files per set								
Nr. of sets								
Size per set (GB)		**Total size (GB)**		**Total Nr of files**		**Avg File size (GB)**		**I/O Rate (GB / hour)**
lower bound	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	
0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	#DIV/0!	#DIV/0!	
CASE 2	***lower bound***	***upper bound***						
Nr. of elements								
Nr. of files per set								
Nr. of sets								
Size per set (GB)		**Total size (GB)**		**Total Nr of files**		**Avg File size (GB)**		**I/O Rate (GB / hour)**
lower bound	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	
0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	#DIV/0!	#DIV/0!	
CASE 3	***lower bound***	***upper bound***						
Nr. of elements								
Nr. of files per set								
Nr. of sets								
Size per set (GB)		**Total size (GB)**		**Total Nr of files**		**Avg File size (GB)**		**I/O Rate (GB / hour)**
lower bound	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	***lower bound***	***upper bound***	
0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	#DIV/0!	#DIV/0!	

All the collected tables of Input/Output information for each code of the SPACE project, provided by WP1 and WP2, are available on Google Drive at the following link: https://docs.google.com/spreadsheets/d/1a-JHvbmPBK2q_V3EcLMySEqXIEXz_YVXu8NSmUvisAQ/edit?usp=sharing

B Appendix II

The Google Form questionnaire used to gather the user needs and requirements in ML, visualization and workflows from WP1 partners is available at the following link:
https://docs.google.com/forms/d/e/1FAIpQLSf7eY90nnY5IpLlVH13kwzCbk1kmEe8NfrvSu6DhIs7dzi9hg/viewform.
The template questionnaire is also shown in PDF format in the following pages.

All the responses to the questionnaire are available as a spreadsheet in the following link:
https://docs.google.com/spreadsheets/d/1BLpq0lkYcQxd4jslta2HLusMiGyqH1Wl/edit?usp=sharing&ouid=113154209904236217999&rtpof=true&sd=true. Personal information, such as the name and email, has been deleted from the spreadsheet and substituted by an ID code.

WP3 - Visualization and ML Use cases. User requirements

Instructions

Please fill out this form once per each use case (for different codes, or different applications of the same code) that are of interest to you in one or more of these aspects:

- Use visualization to explore/analyze data
- Use visualization to present/disseminate/communicate results
- Opportunities for Machine Learning-assisted analysis
- Envisioned challenges when ported to exascale

---

1. Email *

1. Scientific use case

General description of your case

2. Contact name and institution

3. What is the name of the simulation code used? *

4. Add a feference to the publication (if published), or provide brief context. *

5. What is the scientific or research question? And why is the simulation essential to answer it?

6.

2 A- Visualization

Fill in this section only if relevant for your case

7. Main use.

   What do you mainly use visualization for? (Mark option below)

   Seleziona tutte le voci applicabili.

   - [ ] Exploratory Visual Analysis (Navigate through the data to understand it)
   - [ ] Produce support figures for presenting results (static or videos)

8. Visualization Tools
   Do you already use any visualization tool? (Yt, pynbody, Paraview, OpenSpace, VisIVO, proprietary, none, ..) Which one?

   _____

9. Are the tools you've tried not sufficient for your needs? why?

   _____

   _____

   _____

   _____

   _____

10. Are you planning on adopting an existing tool that may suit your needs? Which one and why?

    _____

    _____

    _____

    _____

    _____

11. Mark below the types of data visualizations that you produce more often or find more useful

    Seleziona tutte le voci applicabili.

    - [ ] Charts (scatterplots, boxplots, barcharts, histograms, etc.)
    - [ ] 2D cuts
    - [ ] Particles
    - [ ] 3D volumes, Isosurfaces, Streamlines
    - [ ] Altro: _____

12. As a reference for us, add links to images, papers, documents, etc. of
    - Plots that you make
    - Plots you'd like to make (made by others from similar simulations/results

13. Mark the most relevant challenges (current or potential) to visualize your data

    Seleziona tutte le voci applicabili.

    - [ ] File size or computational mesh are too big to store in disk
    - [ ] in-situ visualization is required
    - [ ] Dimensionality is too high
    - [ ] Spatial resolution is too sparse
    - [ ] Altro: _____

14. What would you like to use visualizations for, that you can't now?

2 B- Machine Learning assisted analysis

Fill in this section only if relevant for your case

15. Current use of ML
    Do you currently perform any ML-assisted analysis? Which?

16. Desired (Already identified potential uses of Machine Lerning) Which?

17. Mark potential applications of ML of interest to your case

Seleziona tutte le voci applicabili.

- [ ] feature exploration (i.e. finding most relevant variables/properties)
- [ ] object detection/segmentation (e.g. identifying galaxies or merger events)
- [ ] dimensionality reduction (feature selection/extraction, clusters, dominant trends/correlations, PCA, Reduced Order Models)
- [ ] finding interesting predicted phenomena/objects or outliers
- [ ] optimize postprocessing
- [ ] generate mock catalogs for observers
- [ ] Runtime computing (e.g. subgrid physics modeling, gas cooling function)
- [ ] Altro: _____

18. Why is the ML-assisted analysis in this case necessary/beneficial to address the scientific/research question?

_____

_____

_____

_____

_____

2 C- Exascale

19. Example of envisioned code use

_____

20. Mark potential challenges of porting to exascale, if any

Seleziona tutte le voci applicabili.

- [ ] data size
- [ ] output cadence
- [ ] high dimensionality
- [ ] runtime
- [ ] Altro: _____

3. Output data

Characteristics of the output data

21. **File characteristics**

Current format? Is it convertible to hdf5?

_____

22. Has Metadata/Description?

Contrassegna solo un ovale.

- ◯ yes
- ◯ no

23. Approximate size (average per file or approximate total per run)

_____

24. Data type (single, double, quad precision)?

_____

25. Structure

Seleziona tutte le voci applicabili.

- ☐ Particles
- ☐ FVM
- ☐ FEM
- ☐ AMR
- ☐ OpenVDB
- ☐ Altro: _____

26. Are there post-process outputs beyond the raw output (e.g., halo catalogs)?

_____

27. **Time**
What is the simulated time span?

_____

28. What is the typical post-process/snapshot frequency?

_____

29. What is the estimated optimal temporal resolution to visualize results (e.g. effects are visible in a span of x decades with a frequency of x centuries) and how does this relate to time steps?

_____

30. **Spatial Resolution**

What is the Spatial scale and the Spatial Resolution? Is the discretization of the space uniform or multi-resolution?

_____

31. **Variables/Fields**

Which are the most relevant or typically used for analysis/visualization?

32. Do you compare several runs with different parameters? or distributions/ evolution within a single run?

Seleziona tutte le voci applicabili.

- [ ] Several runs with different parameters
- [ ] Distributions/ evolution within a single run
- [ ] Altro: _____

33. Is there sample data of this case already available for use in Visualization or Analysis?

THANK YOU!

C Appendix III

The table 1 illustrates the relations between desired solutions and existing lines of work. It contains a line for each response submitted by the partners to the questionnaire presented in Section 2.1.2. The columns of the table contain: the code name, main interests in visualization or ML, and possible matches among the solutions offered by the WP3 partners, as described in this deliverable.

SPACE Code	Visualization Requirements	WP3 Use Case	ML Requirements	WP3 Use Case
RAMSES	3D visualization/Paraview	See Section 4.3	Identify structures (haloes, galaxies, disks, mergers), dimensionality reduction, cooling rates	See Section 5.1
OpenGADGET	NONE	–	Accurate cooling rate values	See Section 5.3
OpenGADGET	Cinematic visualization/Blender	See Section 4.2	Identify galaxies more precisely (or faster)	–
OpenGADGET	NONE	–	Mock images of galaxy clusters	See Section 5.2
Changa/Gasoline2	Higher quality images / streamlines	See Section 4.1	Gas cooling rates	See Section 5.3
iPic3D	NONE	–	Advanced shape recognition tools (unsupervised classification of shapes and meaning of particle distributions)	See Section 5.1
PLUTO	3D rendereing and CLI	See Section 4.3	Identification of critical regions for adaptive (2nd - 4th) order integration	See Section 5.3
BHAC	NONE	–	–	pending
FIL	NONE	–	–	pending

Table 1: Summary of questionnaire requirements in visualization or ML and possible matches among the solutions offered by the WP3 partners.

All complete information can be accessed at the following link: https://docs.google.com/spreadsheets/d/1pjY_ET98BxqmUyrHshrGqAyLqwA3UvtBLZ55nwcBPX0/edit?usp=sharing

This document is Public (PU)
and was produced under SPACE project EU GA 101093441