Toward a Portable and Reproducible High-Performance Visualization for Astronomy & Cosmology

Eva Sciacca*, Valentina Cesare*, Nicola Tuccari†, Fabio Vitello*, Iacopo Colonnelli‡, Carmelita Carbone§, Emiliano Tramontana†, Ugo Becciani*
*INAF Astrophysical Observatory of Catania, Via Santa Sofia 78, Catania, Italy
†University of Catania, Department of Mathematics and Informatics, Viale Andrea Doria 6, Catania, Italy
‡University of Turin, Computer Science Department, Corso Svizzera 185, Torino, Italy
§INAF Institute of Space Astrophysics and Cosmic Physics, Via Alfonso Corti 12, Milano, Italy

***Abstract*—Modern Astronomy and Cosmology (A&C) generate petabyte-scale data volumes requiring the development of a new generation of software tools to access, store, and process them. The Visualization Interface for the Virtual Observatory (VisIVO) tool performs multi-dimensional data analysis and knowledge discovery in multivariate astrophysical datasets. Thanks to containerization and virtualization technologies, VisIVO has already been exploited on top of several distributed computing infrastructures including the European Open Science Cloud (EOSC). In this work, we outline the execution strategies designed to enhance the portability and reproducibility of the VisIVO modular applications for high performance visualization of data generated on the (pre-)exascale systems by HPC A&C simulations performed with the GADGET (GAlaxies with Dark matter and Gas) code.**

***Keywords*—Visualization, Workflows, Cosmology, High Performance Computing**

I. Introduction

Massively large data volumes (of the order of petabytes) are generated from astrophysical observations or simulation codes executed on high performance supercomputers. Such data volumes pose significant challenges for storage, access, and data analysis for effectively enabling scientific discoveries [1]. Pre-exascale systems offer incredible opportunities for scaling HPC applications in Astrophysics and Cosmology (A&C) requiring high performance visualization of their outcome.

The Visualization Interface for the Virtual Observatory (VisIVO) is a toolkit that performs multi-dimensional data analysis and knowledge discovery of a-priori unknown relationships between multi-variate and complex astrophysical datasets. It has already been deployed (1) using Science Gateways [2] to access DCIs (including clusters, grids, and clouds), and (2) using containerization and virtualization technologies on the European Open Science Cloud (EOSC) infrastructures embedded within interactive notebooks applications [3].

In this work we describe the novel integration of VisIVO with hybrid workflows which allows its execution onto multiple computing infrastructures, including mixed HPC-Cloud resources, aiming at the high performance visualization of large-scale outcomes coming from modern HPC applications in A&C.

The adopted strategy consists of the integration of VisIVO with StreamFlow[1] [4], which allows us to enhance the portability of its modular applications and the related resource requirements, fostering reproducibility and maintainability. Moreover, the proposed solution takes advantage of a flexible resource use over heterogeneous HPC facilities.

Finally, we demonstrate the integration of the workflow abstractions for supporting the studies of the effects of massive neutrinos on the large-scale structure of the Universe, using as input large box-size cosmological N-body simulations performed with the GADGET code [5], [6].

II. VisIVO modular applications

To render the visualization of A&C simulation outcomes, VisIVO Server[2] employs three modules for the following tasks: data importing, filtering, and viewing. The data importing task converts the supplied datasets (originally in different formats) into an internal binary format named VisIVO Binary Table (VBT), which is a highly-efficient data representation used by VisIVO Server internally. A VBT consists of a header file (extension .bin.head) containing all necessary metadata, and a raw data file (extension .bin) storing actual data values. E.g., the header may contain information regarding the overall number of fields and the 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 data filtering task can perform several operations including randomization or decimation to reduce the final resolution, mathematical or statistical operators, or cosmological post-processing. Among the latter, there are three commonly used mass assignment functions [7] that calculate the particles densities, i.e., the nearest grid point (NGP), the cloud-in-cell (CIC), and the triangular-shaped cloud (TSC) methods.

[1] StreamFlow, https://streamflow.di.unito.it/
[2] https://visivo.readthedocs.io/en/latest/

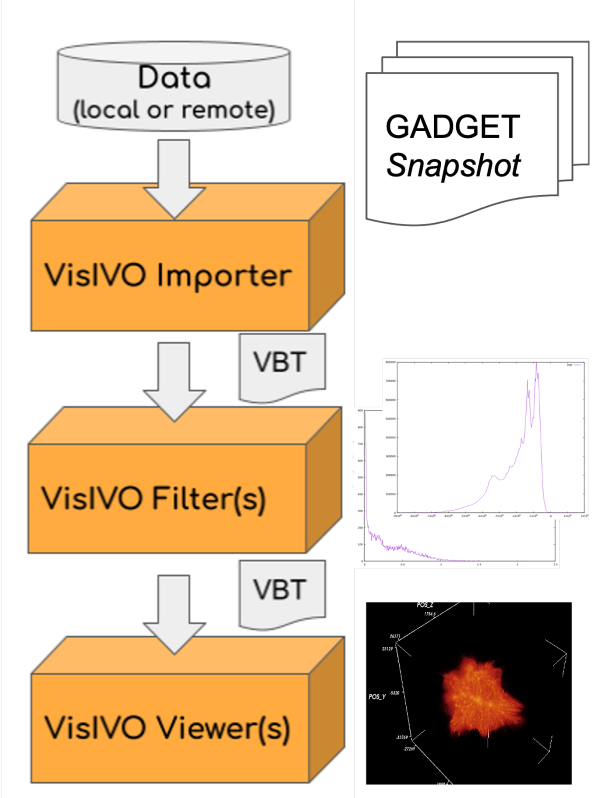


Fig. 1: Typical visualization pipeline of VisIVO consisting on the application of the three main modules on GADGET snapshots.

Finally, the visualization process creates multi-dimensional views from the data that must fit the available RAM. The kinds of visualization include data points, volumes, and vectors and are based on the Visualization Toolkit (VTK)[3].

Figure 1 depicts the typical visualization pipeline of VisIVO for processing GADGET snapshots consisting on the application of the three main modules: `VisIVOImporter` and `VisIVOFilter`, for calculating the particles densities, and `VisIVOViewer`, for final renderings.

III. Methodology

A. *Hybrid workflows*

We employed workflow abstractions to allow a portable representation of the VisIVO modular applications and their resource requirements. This approach fosters reproducibility and maintainability, allowing users to take advantage of heterogeneous HPC facilities (including mixed HPC-Cloud resources) while minimizing data transfer overheads. In particular, we integrated VisIVO with the StreamFlow workflow management system.

StreamFlow complements a standard workflow graph, composed of steps and their inter-dependencies, with a declarative description of potentially complex execution environments.

[3]VTK, https://vtk.org/

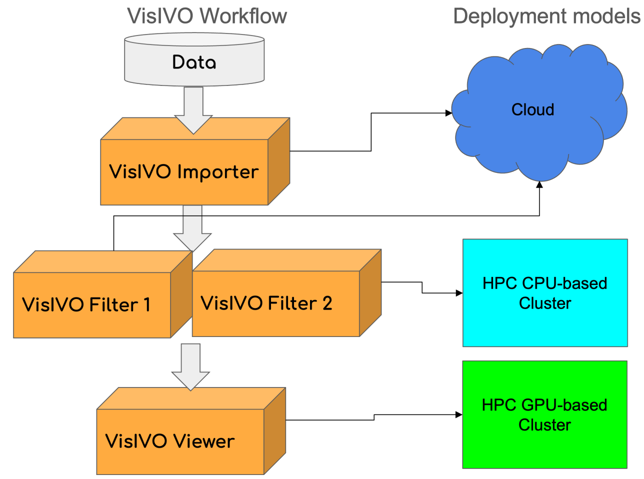


Fig. 2: An example of deployment model for the VisIVO workflow.

This *hybrid workflow model* [8] allows the execution of different steps onto multiple sites without the need to share data spaces, authentication protocols or bidirectional network channels. The StreamFlow control plane takes care of all the orchestration aspects of a workflow execution, including data movements, fault tolerance, and provenance.

With the help of hybrid workflows, different VisIVO modules can run in a multi-container environment, seamlessly switching between different container runtimes (Docker, Singularity, etc.). Moreover, independent workflow steps can be executed in a concurrent fashion on top of multi-agent ecosystems (Cloud/HPC) [9]. Figure 2 shows an example of complex deployment model for the VisIVO workflow.

As a result, StreamFlow provides VisIVO workflows with portability and reproducibility. Indeed, StreamFlow relies on a set of open standards widely adopted in the scientific workflow community, bringing off-the-shelf support for a broad ecosystem of workflow managers (e.g., Galaxy[4] [10] or Airflow[5]) and avoiding technology lock-in. Moreover, it relies on the Common Workflow Language[6] (CWL) [11] for workflow design, the Workflow Hub[7] platform [12] for publishing, and the Workflow Run RO-Crate[8] format [13] for provenance collection.

B. *VisIVO workflows*

We developed two CWL workflows composed of a sequence of three applications: `VisIVOImporter`, `VisIVOFilter`, and `VisIVOViewer` (Figure 1). **Workflow**$_1$ performs the point rendering of the particles visualization through their respective densities, while **Workflow**$_2$ renders a volume visualization of the particle densities. Both workflows are publicly

[4]Galaxy, https://galaxyproject.org/
[5]Airflow, https://airflow.apache.org/
[6]CWL, https://www.commonwl.org/
[7]Workflow Hub, https://workflowhub.eu/
[8]Workflow Run Crate, https://www.researchobject.org/workflow-run-crate

available on the VisIVOCWL GitHub repository[9] and they were assigned the Apache 2.0[10] license. These workflows are described in the README.md file of the VisIVOCWL GitHub repository (workflow n. 3.1 and 3.2).

The `VisIVOImporter` step is common to the two workflows and it executes the following script:

```bash
$ export OMP_NUM_THREADS=AA
$ mpirun --np BB --allow-run-as-root VisIVOImporter
    --fformat gadget
    --out NewTable
    --file snapdir/snap_091.0
```

The script above runs in parallel on a single node over `AA` OpenMP threads and `BB` MPI processes passed as parameters to the workflow itself. This command, takes as input a file of GADGET format (a GADGET snapshot) and it produces a VBT for each species in the GADGET file supplied. In this specific case, it produces two files, NewTableHALO.bin and NewTableGAS.bin, and their corresponding header files, NewTableHALO.bin.head and NewTableGAS.bin.head, which contain information about the dark matter halo and the gas of the GADGET snapshot taken as input. Specifically, this command is executed as:

```bash
$ export OMP_NUM_THREADS=AA
$ mpirun --np BB --allow-run-as-root VisIVOImporter
    paramFile_Imp_Par_MPI.txt
```

The `paramFile_Imp_Par_MPI.txt` parameter points to a file with the list of parameters of the `VisIVOImporter` command. This is true also for all the steps reported below, but in this article we always report the full list of parameters for clarity. In these test cases, the following step of the two workflows takes as input the (NewTableHALO.bin, NewTableHALO.bin.head) couple alone but the procedure is the same if we consider the (NewTableGAS.bin, NewTableGAS.bin.head) couple.

After the importer step, Workflow_1 executes a sequence of two filter steps. The first one executes the following command

```bash
$ VisIVOFilter
    --op pointproperty
    --resolution X_RES Y_RES Z_RES
    --points POS_X POS_Y POS_Z
    --out density.bin
    --outcol density
    --file NewTableHALO.bin
```

The filtering operation modifies the input, in this case the (NewTableHALO.bin, NewTableHALO.bin.head) couple of files. The `pointproperty` operation assigns a property to each data point in the `NewTableHALO.bin` VBT. Specifically, it creates a temporary volume using a field distribution, adopting the CIC algorithm, on a regular mesh. Then, it computes, with the same CIC algorithm, the property for each data point, considering the cells where the point is spread on the volume. The output can be stored in a new table or in a new field of the original input table. In this case, the command creates a column with the density property not present in the original VBT and it saves it in the new file `density.bin`. A correspondent `density.bin.head` is generated as output of the `VisIVOFilter` command. The second `VisIVOFilter` step executes the following command

```bash
$ VisIVOFilter
    --op merge
    --out NewTableHALOMerge.bin
    --filelist tabselection.txt
```

This command generates a new VBT with the (NewTableHALOMerge.bin, NewTableHALOMerge.bin.head) couple of files, merging the original (NewTableHALO.bin, NewTableHALO.bin.head) VBT with the new (density.bin, density.bin.head) VBT, following the specifications reported in the `tabselection.txt` file.

Finally, the viewer step of Workflow_1 executes the command

```bash
$ VisIVOViewer
    --x POS_X_tab_1
    --y POS_Y_tab_1
    --z POS_Z_tab_1
    --color
    --colorscalar density_tab_2
    --colortable volren_glow
    --logscale
    --out VisIVOServerImage
    NewTableHALOMerge.bin
```

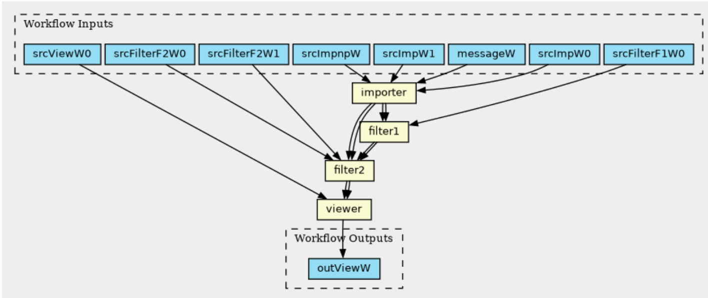
The `VisIVOViewer` command generates four `.png` images, called `VisIVOServerImageXX.png` from the input VBT (NewTableHALOMerge.bin, NewTableHALOMerge.bin.head) and the specifications of the VisIVOViewer command.

In Workflow_2, the filter step executes the command

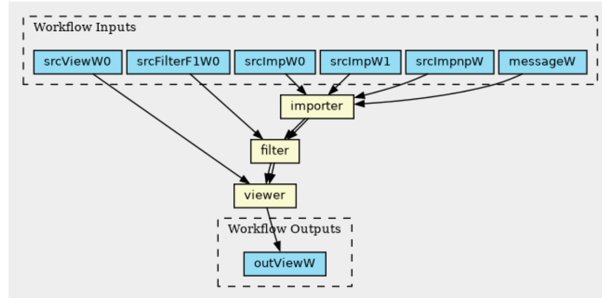
```bash
$ VisIVOFilter
    --op pointdistribute
    --resolution X_RES Y_RES Z_RES
    --points POS_X POS_Y POS_Z
    --out densityvolume.bin
    --file NewTableHALO.bin
```

This command takes as input the (NewTableHALO.bin, NewTableHALO.bin.head) VBT and it transforms it applying the `pointdistribute` operation, saving the new output in the `densityvolume.bin` file. Specifically, the `pointdistribute` operation creates a table which repre-

[9]https://github.com/VisIVOLab/VisIVOCWL
[10]https://spdx.org/licenses/Apache-2.0



(a) **Workflow**$_1$: rendering the particles visualization of the respective densities.



(b) **Workflow**$_2$: rendering the volume visualization of the particle densities.

Fig. 3: Visual representation of the VisIVO CWL workflows developed in this work. Each figure lists all the workflow inputs, outputs, and steps, and depicts the dependency relations between them.

sents a volume from selected fields of the input table that are distributed using NGP, CIC (default) or TSC algorithm, and produces (by default) a density field, which is distributed and divided among the cells in the volume.

Differently from Workflow$_1$, Workflow$_2$ executes a single filter step. After the filter step, the viewer step executes the following command:

```bash
$ VisIVOViewer
    --volume
    --vrendering
    --vrenderingfield Constant
    --color
    --colortable volren_glow
    --showlut
    --out img
    --file densityvolume.bin
```

As in Workflow$_1$, the `VisIVOViewer` command generates four `.png` images from the (densityvolume.bin, densityvolume.bin.head) VBT with the style features specified by the parameters passed to the `VisIVOViewer` command.

The entire Workflow$_1$ and Workflow$_2$ are executed with the following commands[11], where the two `.cwl` files contain the workflows and the two `.yml` files describe the input of the workflows.

```bash
$ cwl-runner
    --streamflow-file streamflow.yml
    workflow_3_1.cwl workflow-job_3_1.yml

$ cwl-runner
    --streamflow_file streamflow.yml
    workflow_3_2.cwl workflow-job_3_2.yml
```

The two `.cwl` files import in turn four and three inner `.cwl` files, corresponding to each step of the two workflows. Workflow$_1$ and Workflow$_2$ are represented pictorially in Figures 3a and 3b, respectively. The `workflow_3_1.cwl` and `workflow_3_2.cwl` scripts declare a CWL `Workflow` class whereas each `.cwl` step called within each of the two workflows contains a CWL `CommandLineTool` class. All VisIVO Docker containers are bound to their related steps by means of the CWL `DockerRequirement` feature.

[11] The names of the two `.cwl` and `.yml` files are reported in the corresponding sections of the `README.md` files in the `VisIVOCWL` repository.

Finally, the `streamflow.yml` file, the entry point of StreamFlow, is in charge of relating each workflow step with the best suitable execution environment, actually plugging the hybrid layer in the workflow execution process. As a first step, StreamFlow translates the CWL dataflow semantics into an internal workflow representation, explicitly modelling a macro dataflow graph [14]. To provide enough flexibility, StreamFlow adopts a three-layered hierarchical representation of execution environments:

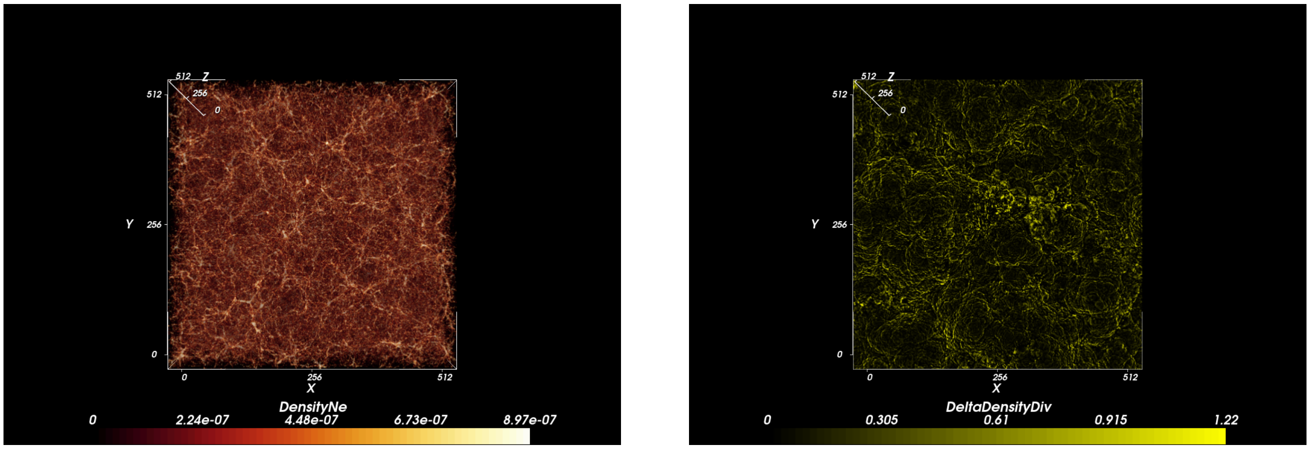
- a `deployment` is the *unit of allocation*, i.e., all its components are always co-allocated when executing a step;
- a `service` constitutes the *unit of binding*, i.e., StreamFlow users can map each workflow step to a single service for execution;
- a `resource` is a single instance of a potentially replicated service and constitutes the *unit of scheduling*, i.e., each step of a workflow is offloaded to a configurable number of resources to be processed.

IV. RESULTS

A. Data description

The presented workflows have been developed to support the studies of the effects of massive neutrinos on the large-scale structure of the Universe through visualization.

Cosmological neutrinos strongly affect the evolution of the largest structures in the Universe, i.e., galaxies, galaxy clusters, cosmic voids, and filaments. In this work we use large box-sized cosmological N-body simulations, the so-called “Dark Energy and Massive Neutrino Universe” (DEMNUni) suite [15], [16]. The DEMNUni simulations have been performed using the tree particle mesh-smoothed particle hydrodynamics (TreePM-SPH) code GADGET-3 [5], specifically modified by [6] to account for the presence of massive neutrinos. This modified version of GADGET-3 follows the evolution of Cold



(a) Volume rendering of cold dark matter particles, in the presence of neutrino particles.

(b) Volume rendering of $\Delta\rho > 0$

Fig. 4: Examples of VisIVO workflow outputs.

Dark Matter (CDM) and Hot Dark Matter (HDM) neutrino particles, treating them as two separated collisionless species.

For the particular case of this work, we have considered the DEMNUni sub-set of simulations with a starting redshift $z_{in} = 99$, and characterised by a comoving volume of $(500\ h^{-1}\ \mathrm{Mpc})^3$ filled with 2048^3 dark matter particles and, when present, 2048^3 neutrino particles [17], [18]. Given the large amount of memory required by the simulations, baryon physics is not included. We consider two different simulations, with a baseline *Planck* cosmology [19], namely a flat ΛCDM model generalized to a $\nu\Lambda$CDM framework by changing only the value of the sum of the three active neutrino masses $M_\nu = \sum_i m_{\nu,i} = 0.16$ eV, and keeping fixed Ω_m and the amplitude of primordial curvature perturbations. This sub-set of simulations is characterised by a softening length $\varepsilon = 5\,h^{-1}$ Kpc, and has been run on the Marconi100 supercomputer at CINECA[12], Italy. For each simulation, we have produced 64 output logarithmically equispaced in the scale factor $a = 1/(1+z)$, in the redshift interval $z = 0-99$, 50 of which lay between $z = 0$ and $z = 10$. For each of the 64 output times, we have dumped on-the-fly a particle snapshot composed by both CDM and neutrino particles.

B. Computing Infrastructure

The VisIVO workflows have been tested on two computing infrastructures, namely the HPC4AI@UNITO[13] cloud infrastructure [20] and the Galileo100 HPC cluster at CINECA[14].

The HPC4AI infrastructure consists of a VM with 8 cores, 32 GB RAM and 500 GB disk deployed on top of OpenStack. It has been used to test the Cloud-HPC continuum execution environment enabled by hybrid workflow models.

On the other hand, 16 nodes of the CINECA Galileo100 HPC facility (48 cores, 384 GB RAM, 2x NVIDIA V100 GPUs each) have been used to render the snapshots of the GADGET data presented in Sec. IV-A. The related data have been generated in the same facility and require 50 TB of storage, making it difficult to move them into another available infrastructure at our premises.

The use of a hybrid workflow system enables us to seamlessly switch between the two infrastructures, performing simpler calculations on the HPC4AI's VMs and more complex ones on Cineca's Galileo 100.

C. Rendering results

The rendering was done using VisIVOViewer volume rendering implementation on the volume generated by the `pointdistribute` filter. A rendering option that was considered was to show the density of the CDM particles described in Section IV-A. Figure 4a shows the results of the volume rendering of the simulation. In addition to this, we are currently working on the rendering of an animation which shows the evolution of the simulation over time.

Then, we considered comparing the densities between the cases where neutrino particles are present and absent. In comparing the two densities, we calculated the difference between the densities in the volumes for the simulation with neutrinos, ρ_1 and the case without, ρ_2, and divided this value by the absolute value of the density without neutrinos, $\Delta\rho = (\rho_1 - \rho_2)/|\rho_2|$. Figure 4b shows the visualization of the particles in the case $\Delta\rho > 0$.

V. Conclusions

Astronomy and Cosmology (A&C) is confronted with the challenge of managing and extracting valuable insights from petabyte-scale datasets generated by observations and simulations. The scale of these data volumes necessitates the development of innovative solutions to address storage, access, and analysis issues. The Visualization Interface for the Virtual Observatory (VisIVO), allows multi-dimensional data analysis

[12]http://www.cineca.it/
[13]https://hpc4ai.unito.it/
[14]https://www.cineca.it/en/data-center/hpc-infrastructure

and knowledge discovery within complex astrophysical and cosmological datasets.

In this paper, we presented the implementation activities to realize high performance visualization of outcomes coming from modern HPC applications in A&C (such as large-scale cosmological simulations). For this, we performed the integration of VisIVO with workflow abstractions and the StreamFlow execution models to enhance the portability of its modular applications and the related resource requirements, fostering reproducibility and maintainability, taking advantage of a flexible resource exploitation over heterogeneous HPC facilities, and including mixed HPC-Cloud resources.

As a next step, we plan to investigate Jupyter Workflow[15] [21], a tool to unleash Jupyter Notebook's productivity and portability features in the HPC area, explicitly targeting Cloud resources and their workload managers (such as Kubernetes), HPC platforms together with their system software (such as SLURM), and their coupled exploitation usage.

In this way, we can exploit Jupyter Notebooks for interactive data analysis with VisIVO workflows for supporting both step-by-step execution on hybrid Cloud-HPC architectures and interactive tuning of VisIVO pipelines, boosting the productivity of the A&C community.

Acknowledgment

The 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 (SPACE CoE). Also, it is supported by the Spoke 1 "FutureHPC & BigData" and the Spoke 3 "Astrophysics and Cosmos Observations" of the ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing – and hosting entity, funded by European Union – NextGenerationEU. The DEMNUni simulations were carried out in the framework of "The Dark Energy and Massive Neutrino Universe" project, using the Tier-0 IBM BG/Q Fermi machine, Intel OmniPath Cluster Marconi-A1 and Marconi 100, at the Centro Interuniversitario del Nord-Est per il Calcolo Elettronico (CINECA). CC acknowledges the CINECA award under the Italian Super-Computing Resource Allocation (ISCRA) initiative, for the availability of high-performance computing resources and storage, as well as the "Accordo Quadro MoU per lo svolgimento di attività congiunta di ricerca Nuove frontiere in Astrofisica: HPC e Data Exploration di nuova generazione", together with storage from INFN-CNAF and INAF-IA2.

References

[1] T. Hey, S. Tansley, K. Tolle, and J. Gray, *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, October 2009. [Online]. Available: https://www.microsoft.com/en-us/research/publication/fourth-paradigm-data-intensive-scientific-discovery/

[2] U. Becciani, E. Sciacca, A. Costa, P. Massimino, C. Pistagna, S. Riggi, F. Vitello, C. Petta, M. Bandieramonte, and M. Krokos, "Science gateway technologies for the astrophysics community," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 2, pp. 306–327, 2015.

[15]https://jupyter-workflow.di.unito.it/

[3] E. Sciacca, M. Krokos, C. Bordiu, C. Brandt, F. Vitello, F. Bufano, U. Becciani, M. Raciti, G. Tudisco, S. Riggi *et al.*, "Scientific visualization on the cloud: the neanias services towards eosc integration," *Journal of Grid Computing*, vol. 20, no. 1, p. 7, 2022.

[4] 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.

[5] V. Springel, "The cosmological simulation code GADGET-2," *Monthly Notices of the Royal Astronomical Society*, vol. 364, no. 4, pp. 1105–1134, Dec. 2005.

[6] M. Viel, M. G. Haehnelt, and V. Springel, "The effect of neutrinos on the matter distribution as probed by the intergalactic medium," *Journal of Cosmology and Astroparticle Physics*, vol. 2010, no. 6, p. 015, Jun. 2010.

[7] W. Cui, L. Liu, X. Yang, Y. Wang, L. Feng, and V. Springel, "An ideal mass assignment scheme for measuring the power spectrum with fast fourier transforms," *The Astrophysical Journal*, vol. 687, no. 2, p. 738, 2008.

[8] I. Colonnelli, "Workflow models for heterogeneous distributed systems," in *Proceedings of the 2nd Italian Conference on Big Data and Data Science (ITADATA 2023), Naples, Italy, September 11-13, 2023*, ser. CEUR Workshop Proceedings, N. Bena, B. D. Martino, A. Maratea, A. Sperduti, E. D. Nardo, A. Ciaramella, R. Montella, and C. A. Ardagna, Eds., vol. 3606. CEUR-WS.org, 2023.

[9] R. F. Da Silva, R. Filgueira, I. Pietri, M. Jiang, R. Sakellariou, and E. Deelman, "A characterization of workflow management systems for extreme-scale applications," *Future Generation Computer Systems*, vol. 75, pp. 228–238, 2017.

[10] The Galaxy Community, "The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update," *Nucleic Acids Research*, vol. 50, no. W1, pp. W345–W351, 2022.

[11] M. R. Crusoe, S. Abeln, A. Iosup, P. Amstutz, J. Chilton, N. Tijanic, H. Ménager, S. Soiland-Reyes, B. Gavrilovic, C. A. Goble, and T. C. Community, "Methods included: Standardizing computational reuse and portability with the Common Workflow Language," *Communications of the ACM*, vol. 65, no. 6, pp. 54–63, 2022.

[12] C. Goble, S. Soiland-Reyes, F. Bacall, S. Owen, A. Williams, I. Eguinoa, B. Droesbeke, S. Leo, L. Pireddu, L. Rodríguez-Navas *et al.*, "Implementing FAIR digital objects in the EOSC-Life workflow collaboratory," *Zenodo*, Mar. 2021.

[13] S. Soiland-Reyes, P. Sefton, M. Crosas, L. J. Castro, F. Coppens, J. M. Fernández, D. Garijo, B. A. Grüning, M. L. Rosa, S. Leo, E. Ó. Carragáin, M. Portier, A. Trisovic, R. Community, P. Groth, and C. A. Goble, "Packaging research artefacts with RO-Crate," *Data Science*, vol. 5, no. 2, pp. 97–138, 2022.

[14] I. Colonnelli, B. Cantalupo, R. Esposito, M. Pennisi, C. Spampinato, and M. Aldinucci, "Hpc application cloudification: The streamflow toolkit," in *12th Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures and 10th Workshop on Design Tools and Architectures for Multicore Embedded Computing Platforms (PARMA-DITAM 2021)*, ser. Open Access Series in Informatics (OASIcs), J. Bispo, S. Cherubin, and J. Flich, Eds., vol. 88. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, p. 5:1–5:13.

[15] C. Carbone, M. Petkova, and K. Dolag, "DEMNUni: ISW, Rees-Sciama, and weak-lensing in the presence of massive neutrinos," *Journal of Cosmology and Astroparticle Physics*, vol. 2016, no. 7, p. 034, Jul. 2016.

[16] G. Parimbelli, C. Carbone, J. Bel, B. Bose, M. Calabrese, E. Carella, and M. Zennaro, "DEMNUni: comparing nonlinear power spectra prescriptions in the presence of massive neutrinos and dynamical dark energy," *Journal of Cosmology and Astroparticle Physics*, vol. 2022, no. 11, p. 041, Nov. 2022.

[17] B. Hernández-Molinero, C. Carbone, R. Jimenez, and C. Peña Garay, "Cosmic Background Neutrinos Deflected by Gravity: DEMNUni Simulation Analysis," *arXiv e-prints*, p. arXiv:2301.12430, Jan. 2023.

[18] G. Verza, C. Carbone, A. Pisani, C. Porciani, and S. Matarrese, "The universal multiplicity function: counting halos and voids," *arXiv e-prints*, p. arXiv:2401.14451, Jan. 2024.

[19] Planck Collaboration, P. A. R. Ade *et al.*, "Planck 2013 results. XVI. Cosmological parameters," *Astronomy & Astrophysics*, vol. 571, p. A16, Nov. 2014.

[20] M. Aldinucci, S. Rabellino, M. Pironti, F. Spiga, P. Viviani, M. Drocco, M. Guerzoni, G. Boella, M. Mellia, P. Margara, I. Drago, R. Marturano, G. Marchetto, E. Piccolo, S. Bagnasco, S. Lusso, S. Vallero, G. Attardi,

A. Barchiesi, A. Colla, and F. Galeazzi, "HPC4AI: an ai-on-demand federated platform endeavour," in *Proceedings of the 15th ACM International Conference on Computing Frontiers, CF 2018, Ischia, Italy, May 08-10, 2018*, D. R. Kaeli and M. Pericàs, Eds. ACM, 2018, pp. 279–286.

[21] I. Colonnelli, M. Aldinucci, B. Cantalupo, L. Padovani, S. Rabellino, C. Spampinato, R. Morelli, R. Di Carlo, N. Magini, and C. Cavazzoni, "Distributed workflows with Jupyter," *Future Generation Computer Systems*, vol. 128, pp. 282–298, 2022.