EuroHPC SPACE CoE: Redesigning Scalable Parallel Astrophysical Codes for Exascale. Invited Paper

Nitin Shukla
CINECA
Casalecchio di Reno, Italy
n.shukla@cineca.it

Alessandro Romeo
CINECA
Casalecchio di Reno, Italy
a.romeo@cineca.it

Caterina Caravita
CINECA
Casalecchio di Reno, Italy
c.caravita@cineca.it

Lubomir Riha
IT4Innovations
Ostrava, Czech Republic
lubomir.riha@vsb.cz

Ondrej Vysocky
IT4Innovations
Ostrava, Czech Republic
ondrej.vysocky@vsb.cz

Petr Strakos
IT4Innovations
Ostrava, Czech Republic
petr.strakos@vsb.cz

Milan Jaros
IT4Innovations
Ostrava, Czech Republic
milan.jaros@vsb.cz

João Barbosa
IT4Innovations
Ostrava, Czech Republic
joao.barbosa@vsb.cz

Radim Vavrik
IT4Innovations
Ostrava, Czech Republic
radim.vavrik@vsb.cz

Andrea Mignone
University of Turin
Turin, Italy
andrea.mignone@unito.it

Marco Rossazza
University of Turin
Turin, Italy
marco.rossazza@unito.it

Stefano Truzzi
University of Turin
Turin, Italy
stefano.truzzi@unito.it

Vittoria Berta
University of Turin
Turin, Italy
vittoria.berta@unito.it

Iacopo Colonnelli
University of Turin
Turin, Italy
iacopo.colonnelli@unito.it

Doriana Medić
University of Turin
Turin, Italy
doriana.medic@unito.it

Elisabetta Boella
E4 COMPUTER ENGINEERING SpA
Scandiano, Italy
elisabetta.boella@e4company.com

Daniele Gregori
E4 COMPUTER ENGINEERING SpA
Scandiano, Italy
daniele.gregori@e4company.com

Eva Sciacca
INAF
Catania, Italy
eva.sciacca@inaf.it

Luca Tornatore
INAF
Trieste, Italy
luca.tornatore@inaf.it

Giuliano Taffoni
INAF
Trieste, Italy
giuliano.taffoni@inaf.it

Pranab J. Deka
KU Leuven
Leuevn, Belgium
pranab.deka@kuleuven.be

Fabio Bacchini
KU Leuven
Leuven, Belgium
fabio.bacchini@kuleuven.be

Rostislav-Paul Wilhelm
KU Leuven
Leuven, Belgium
paul.wilhelm@kuleuven.be

Georgios Doulis
Goethe-Universität
Frankfurt, Germany
gdoulis@itp.uni-frankfurt.de

Khalil Pierre
Goethe-Universität
Frankfurt, Germany
khalil.pierre@itp.uni-frankfurt.de

Luciano Rezzolla
Goethe-Universität
Frankfurt, Germany
luciano.rezzolla@itp.uni-frankfurt.de

Tine Colman
CNRS
Lyon, France
tine.colman@cnrs.fr

Benoît Commerçon
CNRS
Lyon, France
benoit.commercon@cnrs.fr

Othman Bouizi
Eviden
Paris, France
othman.bouizi@eviden.com

Matthieu Kuhn
Eviden
Paris, France
matthieu.kuhn@eviden.com

Erwan Raffin
Eviden
Paris, France
erwan.raffin@eviden.com

Marc Sergent
Eviden
Paris, France
marc.sergent@eviden.com

Robert Wissing
University of Oslo
Oslo, Norway
robertwi@astro.uio.no

Guillermo Marin
Barcelona Supercomputing Center
Barcelona, Spain
guillermo.marin@bsc.es

Klaus Dolag
Ludwig-Maximilians-Universität
Munich, Germany
dolag@usm.uni-muenchen.de

Geray S. Karademir
Ludwig-Maximilians-Universität
Munich, Germany
karademir@usm.lmu.de

Gino Perna
ENGINSOFT SpA
Trento, Italy
g.perna@enginsoft.com

Marisa Zanotti
ENGINSOFT SpA
Trento, Italy
m.zanotti@enginsoft.com

Sebastian Trujillo-Gomez
Heidelberg Institute for Theoretical Studies
Heidelberg, Germany
sebastian.trujillogomez@h-its.org

Abstract

High Performance Computing (HPC) based simulations are crucial in Astrophysics & Cosmology (A&C), helping scientists investigate and understand complex astrophysical phenomena. Taking advantage of exascale computing capabilities is essential for these efforts. However, the unprecedented architectural complexity of exascale systems impacts legacy codes. The SPACE Centre of Excellence (CoE) aims to re-engineer key astrophysical codes to tackle new computational challenges by adopting innovative programming paradigms and software (SW) solutions. SPACE brings together scientists, code developers, HPC experts, hardware (HW) manufacturers, and SW developers. This collaboration enhances exascale A&C applications, promoting the use of exascale and post-exascale computing capabilities. Additionally, SPACE addresses high-performance data analysis for the massive data outputs from exascale simulations and modern observations, using machine learning (ML) and visualisation tools. The project facilitates application deployment across platforms by focusing on code repositories and data sharing, integrating European astrophysical communities around exascale computing with standardised SW and data protocols.

Keywords

Astrophysics & Cosmology codes, High Performance Computing, exascale computing, Center of Excellence

CF Companion '25, Cagliari, Italy

ACM ISBN 979-8-4007-1393-4/25/05
https://doi.org/10.1145/3706594.3728892

ACM Reference Format:
Nitin Shukla, Alessandro Romeo, Caterina Caravita, Lubomir Riha, Ondrej Vysocky, Petr Strakos, Milan Jaros, João Barbosa, Radim Vavrik, Andrea Mignone, Marco Rossazza, Stefano Truzzi, Vittoria Berta, Iacopo Colonnelli, Doriana Medić, Elisabetta Boella, Daniele Gregori, Eva Sciacca, Luca Tornatore, Giuliano Taffoni, Pranab J. Deka, Fabio Bacchini, Rostislav-Paul Wilhelm, Georgios Doulis, Khalil Pierre, Luciano Rezzolla, Tine Colman, Benoît Commerçon, Othman Bouizi, Matthieu Kuhn, Erwan Raffin, Marc Sergent, Robert Wissing, Guillermo Marin, Klaus Dolag, Geray S. Karademir, Gino Perna, Marisa Zanotti, and Sebastian Trujillo-Gomez. 2025. EuroHPC SPACE CoE: Redesigning Scalable Parallel Astrophysical Codes for Exascale. Invited Paper. In *22nd ACM International Conference on Computing Frontiers (CF Companion '25), May 28–30, 2025, Cagliari, Italy*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3706594.3728892

1 The SPACE CoE and its flagship codes

HPC-based numerical simulations are essential for modelling the physical processes shaping our universe and thus advancing A&C knowledge. HPC is also critical for processing modern simulation outputs, requiring exascale computing to enable high-resolution, reliable analyses. The transition to exascale computing presents challenges due to the complexity of modern HPC architectures, which require significant updates to existing simulation and data analysis codes. In response, European CoEs were established under Horizon Europe to enhance and scale parallel codes for exascale performance, fostering collaboration between academia, industry, and technology providers. One such initiative is the Scalable Parallel Astrophysical Codes for Exascale (SPACE) CoE[1], which aims to adapt and optimise European A&C simulation codes for exascale systems. Supported by various European nations and organisations, SPACE promotes co-design activities, knowledge sharing, and the development of advanced computational techniques to maintain Europe's leadership in scientific research. SPACE focuses on improving scalability, energy efficiency, data processing, and

[1] SPACE: https://www.space-coe.eu/

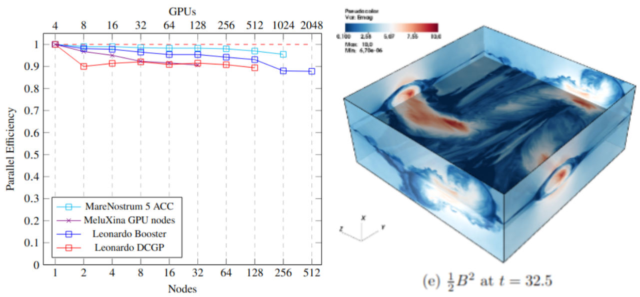

Figure 1: gPluto weak scaling results for the 3D Orszag-Tang on MareNostrum 5 ACC, MeluXina GPU nodes, and Leonardo (Booster and DCGP) (left). Snapshots of the magnetic energy density ($\frac{1}{2}B^2$) at time t = 32.5 (right). Here, time is expressed in units of the light-crossing time of the sheet length.

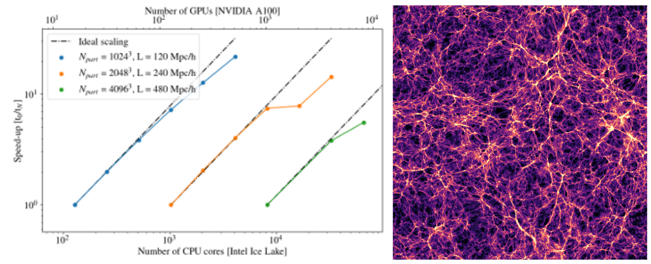

Figure 2: OpenGadget3 strong scaling results of different gravity-only boxes on the Leonardo Booster using three test cases (left). Visualisation of Box3 from the Magneticum Pathfinder simulation set (right). The shown region spans a total size of ~ 180 Mpc and contains ~ $3.5 * 10^8$ dark matter, gas, star, and black hole particles (image credit: B. Seidel).

visualisation capabilities, as well as the collaboration with ML and HPC experts to ensure that upgraded codes meet current and future computational demands in A&C research.

1.1 Pluto

The Pluto code delivers a modular, multiphysics and multi-algorithm framework for simulating astrophysical flows in the presence of high-Mach number flows. It allows independent selection of various hydrodynamic modules and algorithms to accurately model Newtonian hydrodynamics (HD), relativistic HD (RHD), magnetohydrodynamic (MHD), relativistic MHD (RMHD), and resistive relativistic MHD fluids (ResRMHD). The modular design is based on a robust framework for integrating hyperbolic conservation laws by means of modern Godunov-type shock-capturing schemes to ensure high accuracy. Generally speaking, these methods comprise three steps: a reconstruction routine followed by the solution of a Riemann problem at zone edges, and a final update stage.

The new upcoming version of the Pluto code, gPluto[2], has been specifically redesigned to support computations on Graphics Processing Units (GPUs) at the forefront of the exascale era. More specifically, gPluto has been entirely rewritten in `C++`, and relies on the `OpenACC` programming model to provide acceleration up to thousands of GPUs, showing excellent parallel performance. gPluto maintains as much backward compatibility as possible when compared to its predecessor and a very similar user interface. `MPI` communications use non-blocking `MPI` calls and asynchronous data exchange. The development of gPluto involved numerous modifications to optimise parallelisation and memory access, along with a transition from `C` to `C++`. This transition introduced classes for multidimensional arrays and function templates. Implementation details are described in [3] and [14]. The left panel in Fig. 1 shows weak scaling tests of the code conducted on CPU and GPU partitions of three different HPC platforms: MareNostrum 5 Accelerated Partition (GPU), Leonardo Booster (GPU), MeluXina GPU nodes (GPU), and Leonardo DCGP (CPU). The benchmark executed is the 3D version of the well-known Orszag-Tang test. Recently, gPluto has been used to perform 4th-order accurate 3D numerical simulation of magnetic reconnection, triggered by the tearing instability within a resistive relativistic MHD (ResRMHD) framework [4] (Fig. 1 right panel). These computations took advantage of the Leonardo Booster, enabling high-resolution results that would be otherwise unattainable within reasonable timeframes.

1.2 OpenGadget3

OpenGadget3[3] is a collisionless N-Body/Lagrangian cosmological code that uses the smoothed particle HD (SPH) computational method to describe the motion of fluids in addition to gravitational forces, which are calculated using a tree structure. The code allows simulations in a full cosmological context, i.e. accounting for an expanding background and the presence of matter, both "dark" and baryonic (ordinary matter), and dark energy. In addition to the gravitational problem, it also simulates the evolution of baryonic matter, accounting for HD and various physical effects such as radiative cooling, star formation, energy feedback, radiative transfer, magnetic fields, and more. Although the full cosmological context is often the default choice (see right panel of Fig. 2 for an example), having a non-expanding background and a setup with only dark or baryonic matter is equally possible.

OpenGadget3 evolved from the publicly available Gadget-2 code [20]. The code is written in `C/C++` and uses a hybrid parallelisation (`MPI+OpenMP`). OpenGadget3 has been significantly improved compared to its base version, for example, by adding a new state-of-the-art SPH implementation [2], a meshless finite mass solver [8], and `OpenACC` support for running on multiple GPUs [18]. As shown in the left panel of Fig. 2, the code using the `OpenACC` implementation is scaling well up to a few thousand GPUs. Based on this, we are also implementing `OpenMP` offloading, and have experimented with a new strategy of finding neighbours and walking the gravity tree. By coalescing the walk for bunches of particles that belong to the same space region, we consequently synchronise threads and avoid memory divergence. Calculating the gravitational force via direct summation of all the particles within such a bunch of particles avoids branching and conditionals, thus enhancing vectorisation. A proof of concept shows an additional ~ 10× speedup compared to the currently employed approach.

[2] gPluto: https://gitlab.com/PLUTO-code/gPLUTO

[3] OpenGadget3: https://gitlab.lrz.de/MAGNETICUM/Hydro-OpenGadget3

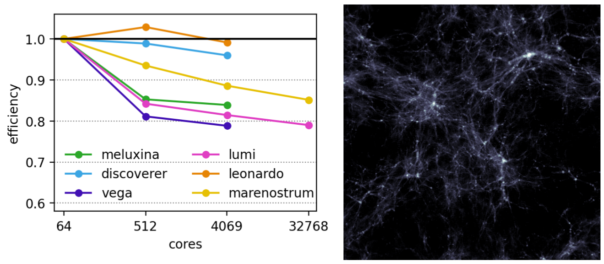

Figure 3: Weak scaling of Ramses (before optimisation) on the CPU partition of several EuroHPC systems for the Sedov test case with 128^3 cells per core (left). Projected image of the cosmological volume test case (right).

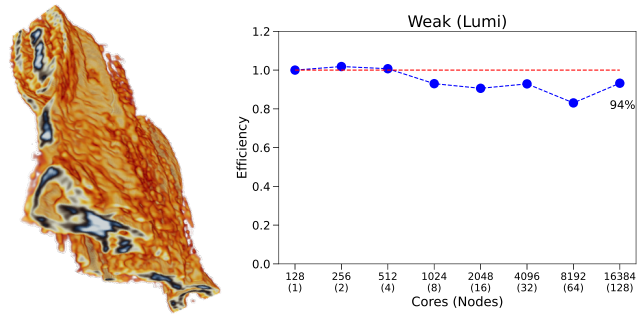

Figure 4: Current-density distribution of a high-resolution 3D simulation of relativistic magnetic reconnection with iPIC3D (left). Weak scaling of iPIC3D on LUMI-C (right).

1.3 Ramses

Ramses[4] is an adaptive mesh refinement (AMR) code used to study astrophysical fluid dynamics and the formation of structures in the Universe [22]. It is based on an oct-tree structure, where parent cells are refined into children cells on a cell-by-cell basis following some user-defined criteria. It couples an Eulerian solver for gas dynamics (hydro) with a particle-mesh (PM) method for collisionless components such as dark matter. The diversity of physics modules included in the code, such as MHD, radiative transfer, self-gravity, and various star formation models, allow for the detailed simulation of complex astrophysical systems ranging from stellar interiors to large cosmological volumes. We focus our efforts on improving performance on the main modules (hydro, gravity, PM, and AMR), as probed by three selected test cases: a Sedov blast wave, a cosmological box (Fig. 3, right panel), and an isolated galaxy. The left panel of Fig. 3 illustrates the weak scaling of the Sedov test.

Ramses is written in `Fortran90` and parallelised with `MPI` using domain decomposition. Internal vectorisation is already excellent. To further improve scaling, we focus on tackling issues related to `MPI` communications. A key limitation when scaling Ramses to a larger number of processes is the increasing surface-to-volume ratio of the `MPI` domains. This results in a significant memory overhead due to boundary duplication between neighbouring domains. To address this, we are integrating `OpenMP` to implement a hybrid parallelisation strategy, reducing the number of `MPI` domains while increasing their volume. This approach improves memory efficiency by a factor of 10 and decreases `MPI` communication overhead, resulting in better execution times and enhanced scalability.

1.4 iPIC3D

The Implicit Particle-in-Cell 3D (iPIC3D[5]) code [12] is a Particle-in-Cell (PIC) simulation tool developed primarily to study plasma dynamics on kinetic scales (see the left panel of Fig. 4 for an example of simulation). The individual (macro)particles used to represent plasma particles are evolved in a Lagrangian framework whereas the moments (plasma density, current, etc) and the self-consistent electric and magnetic fields are tracked on an Eulerian grid. The three main kernels of iPIC3D are (a) Particle Mover, (b) Moment Gatherer, and (c) Field Solver. Due to the implicit nature of the underlying algorithm, as opposed to explicit PIC methods, unresolved scales do not lead to numerical instabilities. This enables the use of time steps and spatial grid sizes that are 10 to 100 times larger than those typically required in traditional explicit PIC codes.

In collaboration with KTH Sweden, we have developed a new version of the code that offers GPU support with `CUDA` (for NVIDIA GPUs) and `HIP` (for AMD GPUs), nonblocking asynchronous communication across `MPI` subdomains, and *in-situ* visualisation using Paraview/Catalyst. The right panel of Fig. 4 illustrates the weak scaling of the code on LUMI-C, demonstrating a parallel efficiency of over 90%.

We have recently implemented the Energy Conserving Semi-Implicit Method (ECSIM) [10, 11] in this new version of the code which enables us to conserve energy of the system exactly up to machine precision. We have also implemented the relativistic semi-implicit method (RelSIM) [1], which is the relativistic counterpart of ECSIM. At present, the moment gatherer module is the most expensive part of the computation, encompassing $\sim$ 80% of the overall runtime. We aim to further optimise this module via vectorisation. After optimisation, we will implement ECSIM and RelSIM algorithms in the GPU code. This will follow ongoing efforts to scale these algorithms, along with iPIC3D, to several thousand GPUs, potentially paving the way for exascale computing.

1.5 ChaNGa

ChaNGa[6] is an N-body and smoothed particle MHD (SPMHD) code designed to simulate a wide range of astrophysical systems [9, 13] (see Fig. 5 left panel for an example). Building upon the gravity and SPMHD algorithms of gasoline [24] and pkdgrav [21], ChaNGa leverages the `Charm++` framework to deliver superior parallel scalability compared to its predecessors. This scalability stems from three key features of `Charm++`. First, overdecomposition, which divides the computational workload into many more units (called chares or tree pieces) than available processors. Second, dynamic load balancing, where the `Charm++` runtime system dynamically manages chares through continuous load-balancing strategies, evaluating and migrating tree pieces between processors during execution to maintain optimal load distribution. Third,

[4]Ramses: https://git-cral.univ-lyon1.fr/hpc/space/ramses
[5]iPIC3D: https://github.com/Pranab-JD/iPIC3D-CPU-SPACE-CoE
[6]ChaNGa: https://github.com/N-BodyShop/changa

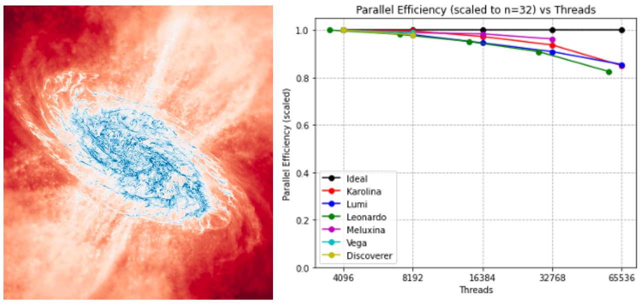

Figure 5: Magnetic field strength distribution from a high-resolution galaxy simulation performed with ChaNGa (left). Weak scaling performance of ChaNGa across multiple EuroHPC supercomputers (right).

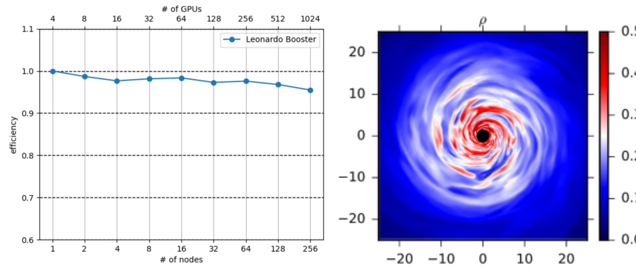

Figure 6: BHAC weak scaling (left) on Leonardo Booster (GPU partition) for the 3D simulation of a magnetised torus around a Kerr black hole–z=0 plane shown (right).

asynchronous task-based execution model, in which computation is message-driven, with tasks triggered by asynchronous messages. This model enables overlapping computation and communication, reducing idle time and enhancing efficiency. The right panel of Fig. 5 demonstrates ChaNGa performance, showing excellent scaling across up to 65,536 cores on the CPU partitions of multiple EuroHPC supercomputers. `Charm++` also provides support to execute `CUDA` kernels on the GPU asynchronously and to manage data transfers between the CPU and GPU. ChaNGa has previously only ported its gravity module to GPU, which has shown excellent scaling up to $\approx 3,000$ nodes (1 GPU per node) on the Piz Daint supercomputer. Recently, we have begun enhancing ChaNGa's GPU module by offloading computationally intensive tasks to the GPU. As a first step, we have implemented a preliminary GPU offload for the radiative cooling module, achieving a speedups varying from 4× to 20× depending on the test case. Additionally, we have improved communication balance within ChaNGa by implementing tree-piece replication, which distributes tree pieces across multiple processors, preventing any single processor from being overwhelmed with messages.

1.6 BHAC

The Black Hole Accretion Code (BHAC[7]) is a multidimensional General RMHD (GRMHD) code that is mainly used to study accretion flows onto compact objects [15, 17]. BHAC has been designed to solve the GRMHD equations in arbitrary stationary spacetimes (Cowling approximation) exploiting AMR techniques with an oct-tree block-based approach provided by the MPI-AMRVAC framework[8]. The code is second-order accurate and uses finite volume and high-resolution shock-capturing (HRSC) methods. Originally designed to study black hole (BH) accretion in ideal GRMHD, BHAC has been extended to incorporate nuclear equations of state, neutrino leakage, charged and purely geodetic test particles, and non-black hole fully numerical metrics. In addition, a non-ideal resistive GRMHD module has been developed and implemented. BHAC's results, after a general-relativistic ray-tracing (GRRT) post-processing, can be used to compute synthetic observable images of BH shadows and the surrounding accretion flows. These calculations are performed with the GRRT Black Hole Observations in Stationary Spacetimes (BHOSS) code [25]. The GRMHD simulation data produced by BHAC are used as input for BHOSS to produce accretion flow and BH shadow images. Currently, two modules have been ported to GPU using `OpenACC` and are fully operational on multiple GPUs: the primitive reconstruction scheme and the Riemann solver. The former is the most resource demanding module of BHAC. The multi-GPU performance of BHAC is presented in Fig. 6 (left panel) where an approximate 95% weak scaling efficiency up to 1,024 GPUs is achieved. As test case, a 3D simulation of a magnetised torus around a Kerr black hole–z=0 plane was used (right panel of Fig. 6). The current GPU-port yields a $\sim 20\times$ speedup over the CPU version. Ongoing optimisations aim to boost performance further by improving parallel efficiency and reducing CPU-GPU data transfers.

1.7 FIL/GRACE

FIL is a GRMHD code capable of simulating relativistic fluids on a curved, dynamically evolving spacetime. This feature, which distinguishes FIL from BHAC, makes it well-suited for modelling BH neutron star (BHNS) and binary neutron star (BNS) collisions (see the right panel of Fig. 7 for an example). FIL leverages the computational infrastructure of the Einstein Toolkit (ET), a codebase designed to support numerical relativity simulations. This includes features such as box-in-box AMR and fourth-order Runge-Kutta integration methods, both of which are employed by FIL. FIL is the successor to Illinois GRMHD, the first open-source GRMHD code for BNS simulations. With fourth-order finite difference methods and a tabulated Equation of State (EOS) interface, FIL offers improved accuracy and enables exploration of the nuclear EOS parameter space. FIL is written in `C++` and uses `MPI` and `OpenMP` for parallelisation. Strong and weak scaling tests of the code are shown in the left panel of Fig. 7.

Work is currently underway to replace the functionalities of the ET framework used in FIL with a new computational infrastructure called GRACE. GRACE is a GRMHD codebase built on `Kokkos` to leverage heterogeneous HW. This will allow FIL to run on a multitude of GPU models. Unlike the current FIL implementation, which is strictly based on Cartesian coordinates, GRACE supports multiple coordinate systems. This flexibility enables simulations

[7]BHAC: https://gitlab.itp.uni-frankfurt.de/BHAC-release/bhac
[8]MPI-AMRVAC: https://amrvac.org

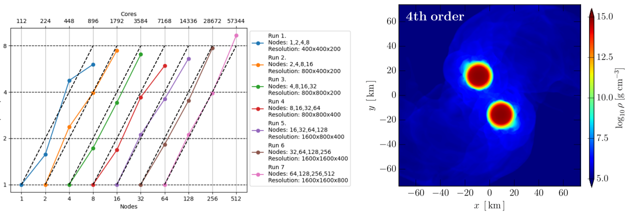

Figure 7: Strong and weak scaling of FIL on Marenostrum (left). Density plot of BNS collision (right).

	X86_64			AARCH64			
Compiler	GNU	IFORT	IFX	GNU NEON	GNU SVE2	ACFL NEON	ACFL SVE2
Time gain (%)	3.7	1	9	-	9.6	2.7	4.1

Table 1: Gains achieved on RAMSES hydro module by optimising vectorisation with different compilers and architectures.

to exploit problem-specific symmetries, improving computational efficiency.

2 Profiling, Co-design & Energy Efficiency

Profiling. Optimisation of all SPACE codes is a key objective of the project. To enable focused optimisation and GPU acceleration of the most critical parts of the codes, a rigorous profiling campaign was done using the POP3 CoE[9] methodology and tools.

Co-design. Another key objective of SPACE is to prepare code for emerging European technologies, as those developed within the EPI-SGA2[10] and EUPEX[11] projects, by targeting the Rhea processor from SiPearl with High Bandwidth Memory (HBM) and ARM Neoverse V1 cores. This involves close collaboration between application developers and HW/SW technology providers to facilitate knowledge exchange and mutual influence. Our investigation focuses on two primary performance tracks: (i) evaluating the impact of HBM versus Double Data Rate (DDR) memory, using Intel Sapphire Rapids HBM processors as a reference; and (ii) assessing the maturity of the ARM ecosystem, including compilers and third-party libraries, with a particular focus on the capability to compile code that efficiently leverages Scalable Vector Extension (SVE) for Single Instruction Multiple Data (SIMD) operations. In the absence of the Rhea processor, we currently use ARM-based platforms such as the NVIDIA Grace CPU for early development and testing. For example, Table 1 showcases the performance gains achieved by optimising the vectorisation of RAMSES using different compilers on both x86_64 and AARCH64 architectures. While performance improvements were observed with all compilers, the optimal set of optimisations is dependent on both the compiler and the architecture. This dependency complicates the task of achieving portable code vectorisation.

Improving Energy Efficiency. As energy awareness is becoming an essential focus for all data centre operators, all SPACE codes have been analysed in terms of energy efficiency on several HPC

[9]POP: Performance Optimisation and Productivity CoE: https://pop-coe.eu/
[10]EPI: European Processor Initiative: https://www.european-processor-initiative.eu/
[11]EUPEX: European Pilot for Exascale: https://eupex.eu/

HW	CPU or GPU frequency [GHz]	Energy efficiency [MFLOPs/W]	Node energy savings	Runtime impact
Nvidia A100 GPU	default	-	–	100 %
	1.29	-	-6 %	103 %
	1.11	-	-9 %	113 %
Intel Xeon 9468 w. HBM	default	376	–	100 %
	CF 2.6; UCF 1.4	398	-4 %	101 %
	CF 2.6; UCF 1.0	411	-6 %	105 %
Nvidia Grace CPU	default	628	–	100 %
	CF 2.9 GHz	805	-22 %	101 %
	CF 2.1 GHz	964	-35 %	122 %

Table 2: Energy efficiency analysis results for PLUTO code.

machines and modern HW platforms. We have investigated how to tune the performance knobs provided by different platforms (CPU core frequency (CF), CPU uncore frequency (UCF), and GPU streaming multiprocessor frequency) to minimise energy consumption when running SPACE codes. Table 2 shows that ARM-based NVIDIA Grace CPU delivers much better energy efficiency. In addition, by proper HW setup we can save up to 22% of energy without runtime impact. For A100 GPUs, energy consumption can be reduced by up to 9% with minimal runtime impact. For more details see public deliverables D2.2 and D2.4[12].

3 Extreme Data Processing and Analysis

One SPACE CoE objective is to integrate data analysis techniques with exascale A&C applications to enhance scientific discoveries from numerical simulations. *Extreme data processing and analysis* addresses this topic by developing prototype frameworks based on workflow engines to design modular HPC applications [5] and on visualisation and ML tools detailed hereafter.

High Performance Visualisation. State-of-the-art large-scale A&C simulations generate petabytes of data, capturing the evolution of millions of objects across 3D space and numerous physical parameters. Developing visualisation systems for pre-exascale architectures is challenging due to HW/SW compatibility, data management, scalability, energy efficiency, and resource constraints, making development and porting processes complex and requiring careful, system-specific optimisation. SPACE is developing three visualisation strategies to address the aforementioned challenges, which are relevant to the A&C community [23] and potentially applicable to simulation results from other scientific domains: 1) A novel solution for *in-situ* visualisation using Hecuba[13], where the analysis and visualisation run concurrently with the simulation, bypassing the need to store the full outputs and permitting the visualisation during runtime. 2) A workflow to produce cinematic visualisations [7] of large volumetric datasets in Blender[14] with high-level of control over the image quality, applied to SPMHD data from CHANGA. 3) The adaptation of VisIVO[15] modular applications [19] for high-performance data visualisation to efficiently exploit (pre-)exascale HPC systems. For this approach, the workflow management system StreamFlow [6] is adopted in order to improve portability and reproducibility of the workflow systems

[12]SPACE deliverables: https://www.space-coe.eu/deliverables.php
[13]Hecuba: https://github.com/bsc-dd/hecuba
[14]Blender: https://www.blender.org/download/
[15]VisIVO: https://visivo.readthedocs.io/

by simplifying the definition, execution, and management of the different visualisation tasks.

ML for Astrophysics. Simulations are sophisticated models used to interpret observations and make testable predictions as an integral part of the scientific method. However, their growing size and complexity is becoming a barrier to their own interpretability. Exascale A&C simulations will produce outputs that will be very challenging for humans to interpret and analyse. SPACE is leveraging powerful scalable ML methods to maximise the scientific discovery potential of these expensive simulations with a focus on exploration, interpretation, and inference. In addition to facilitating the interpretation and analysis of simulation outputs, we have identified the potential to integrate ML models directly into codes at runtime. This approach offers a promising path to efficiently and accurately incorporate physical processes that are currently too computationally expensive to model in large-scale simulations. SPACE is developing three ML applications to enhance the scientific workflows of exascale A&C simulations. The first one consists of a modular framework for interactive explorative access and knowledge discovery in arbitrarily large cosmological simulations using Representation Learning [16]. The second application complements this with a general inference tool that allows cosmological simulations to predict the physical properties and evolution of observed cosmic structures based on observables. The last application aims to develop a surrogate ML model to include costly radiative transfer effects in the largest cosmological simulations by pretraining using detailed calculations.

4 Services For the Community

Transition to exascale HPC demands an up-to-date approach to code writing and integration. It is within this context that the following services have been implemented: services for continuous integration, issue tracking, and data archiving form the foundation of the modern development and research infrastructure. Also, a new vision of continuous deployment is under evaluation in order to be compatible with national computing centres and their security policies and to deal with target machine recipes in a much easier way. The SPACE CoE is defining I/O standards and data models for the A&C community. The goal is to enable data exchange between compatible simulation codes and make outputs accessible to the broader community. A standardised metadata model is also being developed to describe simulation results, supporting discovery and reuse. This model follows the Findable, Accessible, Interoperable, Reusable (FAIR) principles and builds on the work of the International Virtual Observatory Alliance (IVOA).

5 Training activities

A key focus of the SPACE CoE is delivering high-quality training to support the broader scientific and industrial community. To this end, SPACE organises a diverse range of events, including webinars, workshops, tutorials, schools, and hackathons, targeted at early-career researchers and professionals. These activities cover topics such as A&C codes, dataset discovery, data processing, ML, HPC visualisation, and energy efficiency in HPC. Training content also includes updates on exascale development from SPACE code developers and HPC experts. All materials are publicly available via the SPACE CoE website and YouTube channel. SPACE actively collaborates with initiatives like EuroCC and CASTIEL (Training Sprint), alongside other CoEs and host institutions, to expand its reach. Events are promoted through the website, LinkedIn, newsletters, and mailing lists to engage the HPC community and potential users.

6 Summary

The SPACE CoE has recently passed its two-year milestone and is now fully engaged in application optimisation and porting efforts for both current and future EuroHPC systems. Our results demonstrate strong code scalability and active porting to advanced architectures, ensuring long-term sustainability and performance. Extensive work is also underway in energy efficiency, profiling, optimisation, and extreme-scale data processing, analysis, and visualisation. For the latest updates and detailed information, please visit our project website[16].

Acknowledgments

SPACE CoE is funded by the European Union. It has received funding from the European High Performance Computing Joint Undertaking and from Belgium, the Czech Republic, France, Germany, Greece, Italy, Norway, and Spain under grant agreement No. 101093441.

We acknowledge ISCRA for awarding this project access to the Leonardo supercomputer and the EuroHPC Joint Undertaking for granting us access to the Leonardo, MareNostrum and MeluXina supercomputer through EuroHPC Benchmark and Development Access calls.

References

[1] F. Bacchini. 2023. RelSIM: A Relativistic Semi-implicit Method for Particle-in-cell Simulations. *ApJS* 268, 2 (2023). doi:10.3847/1538-4365/acefba

[2] A. M. Beck, G. Murante, A. Arth, R. S. Remus, A. F. Teklu, J. M. F. Donnert, S. Planelles, M. C. Beck, P. Förster, M. Imgrund, K. Dolag, and S. Borgani. 2016. An improved SPH scheme for cosmological simulations. *Monthly Notices of the Royal Astronomical Society* 455, 2 (Jan. 2016), 2110–2130. doi:10.1093/mnras/stv2443 arXiv:1502.07358 [astro-ph.CO]

[3] V. Berta, A. Mignone, M. Bugli, and G. Mattia. 2024. A 4th-order accurate finite volume method for ideal classical and special relativistic MHD based on pointwise reconstructions. *J. Comput. Phys.* 499 (2024), 112701. doi:10.1016/j.jcp.2023.112701

[4] V. Berta, A. Mignone, M. Bugli, G. Mattia, and M. Rossazza. 2024. Towards 4th-order accurate 3D Magnetic Reconnection in Relativistic Plasmas. *submitted to Astronum Proceedings* (2024).

[5] I. Colonnelli, R. Birke, G. Malenza, G. Mittone, A. Mulone, J. Galjaard, L. Y. Chen, S. Bassini, G. Scipione, J. Martinovič, V. Vondrák, and M. Aldinucci. 2024. Cross-Facility Federated Learning. *Procedia Computer Science* 240 (2024), 3–12. doi:10.1016/j.procs.2024.07.003 Proceedings of the First EuroHPC user day.

[6] I. Colonnelli, B. Cantalupo, I. Merelli, and M. Aldinucci. 2021. StreamFlow: cross-breeding cloud with HPC. *IEEE Transactions on Emerging Topics in Computing* 9, 4 (2021), 1723–1737. doi:10.1109/TETC.2020.3019202

[7] M. Faltýnková, O. Meca, T. Brzobohatý, L. Říha, M. Jaroš, and P. Strakoš. 2025. Workflow for high-quality visualisation of large-scale CFD simulations by volume rendering. *Advances in Engineering Software* 200 (Feb. 2025), 103822. doi:10.1016/j.advengsoft.2024.103822

[8] F. Groth, U. P. Steinwandel, M. Valentini, and K. Dolag. 2023. The cosmological simulation code OPENGADGET3 - implementation of meshless finite mass. *Monthly Notices of the Royal Astronomical Society* 526, 1 (Nov. 2023), 616–644. doi:10.1093/mnras/stad2717 arXiv:2301.03612 [astro-ph.IM]

[9] P. Jetley, F. Gioachin, C. Mendes, L. V. Kale, and T. R. Quinn. 2008. Massively parallel cosmological simulations with ChaNGa. In *Proceedings of IEEE International Parallel and Distributed Processing Symposium 2008.*

[10] G. Lapenta. 2017. Exactly energy conserving semi-implicit particle in cell formulation. *J. Comput. Phys.* 334 (2017), 349–366. doi:10.1016/j.jcp.2017.01.002

[16] SPACE: https://www.space-coe.eu/

[11] G. Lapenta, D. Gonzalez-Herrero, and E. Boella. 2017. Multiple-scale kinetic simulations with the energy conserving semi-implicit particle in cell method. *Journal of Plasma Physics* 83, 2 (2017), 705830205. doi:10.1017/S0022377817000137

[12] S. Markidis, G. Lapenta, and Rizwan-uddin. 2010. Multi-scale simulations of plasma with iPIC3D. *Math. Comput. Simul.* 80 (2010), 1509–1519. doi:10.1016/j.matcom.2009.08.038

[13] H. Menon, L. Wesolowski, G. Zheng, P. Jetley, L. Kale, T. Quinn, and F. Governato. 2015. Adaptive techniques for clustered N-body cosmological simulations. *Computational Astrophysics and Cosmology* 2, Article 1 (March 2015). doi:10.1186/s40668-015-0007-9 arXiv:1409.1929 [astro-ph.IM]

[14] A. Mignone, V. Berta, M. Rossazza, M. Bugli, G. Mattia, L. Del Zanna, and L. Pareschi. 2024. A Fourth-Order Finite Volume Scheme for Resistive Relativistic Magnetohydrodynamics. arXiv:2407.08519 [astro-ph.HE] https://arxiv.org/abs/2407.08519

[15] H. Olivares, O. Porth, J. Davelaar, E. R. Most, C. M. Fromm, Y. Mizuno, Z. Younsi, and L. Rezzolla. 2019. Constrained transport and adaptive mesh refinement in the Black Hole Accretion Code. *Astronomy & Astrophysics* 629 (2019), A61. doi:10.1051/0004-6361/201935559

[16] K. L. Polsterer, B. Doser, A. Fehlner, and S. Trujillo-Gomez. 2024. Spherinator and HiPSter: Representation Learning for Unbiased Knowledge Discovery from Simulations. *arXiv e-prints*, Article arXiv:2406.03810 (June 2024), arXiv:2406.03810 pages. doi:10.48550/arXiv.2406.03810 arXiv:2406.03810 [astro-ph.IM]

[17] O. Porth, H. Olivares, Y. Mizuno, Z. Younsi, L. Rezzolla, M. Moscibrodzka, H. Falcke, and M. Kramer. 2017. The Black Hole Accretion Code. *Computational Astrophysics and Cosmology* 4 (2017). doi:10.1186/s40668-017-0020-2

[18] A. Ragagnin, K. Dolag, M. Wagner, C. Gheller, C. Roffler, D. Goz, D. Hubber, and A. Arth. 2020. Gadget3 on GPUs with OpenACC. *arXiv e-prints*, Article arXiv:2003.10850 (March 2020), arXiv:2003.10850 pages. doi:10.48550/arXiv.2003.10850 arXiv:2003.10850 [astro-ph.IM]

[19] E. Sciacca, U. Becciani, A. Costa, F. Vitello, P. Massimino, M. Bandieramonte, M. Krokos, S. Riggi, C. Pistagna, and G. Taffoni. 2015. An integrated visualization environment for the virtual observatory: Current status and future directions. *Astronomy and Computing* 11 (2015), 146–154.

[20] V. Springel. 2005. The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society* 364, 4 (Dec. 2005), 1105–1134. doi:10.1111/j.1365-2966.2005.09655.x arXiv:astro-ph/0505010 [astro-ph]

[21] J. G. Stadel. 2001. *Cosmological N-body simulations and their analysis*. Ph. D. Dissertation. University of Washington, Seattle.

[22] R. Teyssier. 2002. Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES. *Astronomy & Astrophysics* 385 (April 2002), 337–364. doi:10.1051/0004-6361:20011817 arXiv:astro-ph/0111367 [astro-ph]

[23] N. Tuccari, E. Sciacca, F. Vitello, I. Colonnelli, Y. Becerra, E. S. Cintero, G. Marin, M. Jaros, L. Riha, P. Strakos, S. Trujillo-Gomez, E. Tramontana, and R. Wissing. to appear. High Performance Visualization for Astrophysics and Cosmology. *PDP 2025 Conference Proceedings* (to appear).

[24] J. W. Wadsley, B. W. Keller, and T. R. Quinn. 2017. Gasoline2: a modern smoothed particle hydrodynamics code. *Monthly Notices of the Royal Astronomical Society* 471, 2 (Oct. 2017), 2357–2369. doi:10.1093/mnras/stx1643 arXiv:1707.03824 [astro-ph.IM]

[25] Z. Younsi, O. Porth, Y. Mizuno, C. M. Fromm, and H. Olivares. 2020. Modelling the polarised emission from black holes on event horizon-scales. *Proceedings of the International Astronomical Union* 14, S342 (April 2020), 9–12. doi:10.1017/s1743921318007263