A Systematic Mapping Study of Italian Research on Workflows

Marco Aldinucci
University of Turin, Italy

Elena Baralis
Polytechnic University of Turin, Italy

Valeria Cardellini
University of Rome Tor Vergata, Italy

Iacopo Colonnelli
University of Turin, Italy

Marco Danelutto
University of Pisa, Italy

Sergio Decherchi
Fondazione IIT, Genoa, Italy

Giuseppe Di Modica
University of Bologna, Italy

Luca Ferrucci
University of Pisa, Italy

Marco Gribaudo
Polytechnic University of Milan, Italy

Francesco Iannone
ENEA HPC laboratory, Rome, Italy

Marco Lapegna
University of Naples Federico II, Italy

Doriana Medić
University of Turin, Italy

Giuseppa Muscianisi
CINECA, Bologna, Italy

Francesca Righetti
University of Pisa, Italy

Eva Sciacca
INAF, Catania, Italy

Nicola Tonellotto
University of Pisa, Italy

Mauro Tortonesi
University of Ferrara, Italy

Paolo Trunfio
University of Calabria, Italy

Tullio Vardanega
University of Padua, Italy

ABSTRACT

An entire ecosystem of methodologies and tools revolves around scientific workflow management. They cover crucial non-functional requirements that standard workflow models fail to target, such as interactive execution, energy efficiency, performance portability, Big Data management, and intelligent orchestration in the Computing Continuum. Characterizing and monitoring this ecosystem is crucial to developing an informed view of current and future research directions. This work conducts a systematic mapping study of the Italian workflow research community, analyzing 25 tools and 10 applications from several scientific domains in the context of the "National Research Centre for HPC, Big Data, and Quantum Computing" (ICSC). The study aims to outline the main current research directions and determine how they address the critical needs of modern scientific applications. The findings highlight a variegated research ecosystem of tools, with a prominent interest in advanced workflow orchestration and still immature but promising efforts toward energy efficiency.

CCS CONCEPTS

• Computing methodologies → Distributed computing methodologies; **• Computer systems organization → Heterogeneous (hybrid) systems**.

SC-W 2023, November 12–17, 2023, Denver, CO, USA

ACM ISBN 979-8-4007-0785-8/23/11.
https://doi.org/10.1145/3624062.3624285

KEYWORDS

Scientific Workflows, Computing Continuum, HPC

ACM Reference Format:
Marco Aldinucci, Elena Baralis, Valeria Cardellini, Iacopo Colonnelli, Marco Danelutto, Sergio Decherchi, Giuseppe Di Modica, Luca Ferrucci, Marco Gribaudo, Francesco Iannone, Marco Lapegna, Doriana Medić, Giuseppa Muscianisi, Francesca Righetti, Eva Sciacca, Nicola Tonellotto, Mauro Tortonesi, Paolo Trunfio, and Tullio Vardanega. 2023. A Systematic Mapping Study of Italian Research on Workflows. In *Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis (SC-W 2023), November 12–17, 2023, Denver, CO, USA*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3624062.3624285

1 INTRODUCTION

In the Big Data (and AI) ramp-up period, driven by the rush of specialization, the tools and cultures revolving around traditional HPC and Cloud platforms serving data analysis tasks started differentiating [60]. However, pushed by their complementarity and interdependence for the broad spectrum of major research domains, their path of integration (and perhaps unification) has not taken too long to gain strength [61]. The cross-breeding of HPC and Cloud resource provisioning in the same computing ecosystem is becoming a consolidated approach to ease access to HPC resources and accelerate Cloud services [3, 6]. Also, according to the Edge Computing paradigm [68], Cloud services are progressively moving closer to data sources to comply with crucial extra-functional requirements, such as latency, privacy, and security.

Despite this trend, tghe orchestration of large-scale scientific applications in modular supercomputers [71, 73], and in the large, over the full spectrum of the so-called *Computing Continuum* [9] (i.e., hybrid HPC+Cloud+Edge execution environments) is still an ambitious goal. Execution locations can be *heterogeneous*, exposing different hardware architectures, communication layers, authentication methods, and resource allocation paradigms. They are typically *geographically distributed*, meaning placement choices significantly impact communication overhead, especially in the Big Data domain. Eventually, different execution locations might exhibit different a significantly different *energy efficiency* to complete the same task.

Workflows are a powerful abstraction to model large-scale scientific applications. They can explicitly model many relevant aspects of a modular workload, like the requirements of each module, the data dependencies between different modules, and a global view of the whole application as a graph. They are an effective *intermediate representation* for distributed applications. Nevertheless, workflows alone fail to cover the whole process of workload orchestration in the Compute Continuum, and they do not target crucial non-functional requirements such as interactive execution, energy efficiency, and performance portability.

For this reason, an entire ecosystem of methodologies and tools is being developed that revolves around the field of scientific workflow orchestration. Some aim to lower the barriers between prototypical workflows and production-ready implementations, allowing domain experts to interactively access heterogeneous computing resources through a common, high-level interface. Others start from a graph-based representation of a complex application and optimize one or more aspects of its enactment, e.g., scheduling, data movements, Quality-of-Service (QoS), or energy consumption. Further approaches expose programming paradigms that serve as abstraction layers between the application level and the details of the underlying execution environment (e.g., network, storage, or hardware architecture), fostering performance portability.

This report stems from a collaboration among many partners with diverse competencies under the framework of the novel Italian *"National Research Centre for High-Performance Computing, Big Data and Quantum Computing"* (ICSC), which is described in Sec. 1.1. The adopted approach is derived from the *Systematic Mapping Study* (SMS) methodology, which aims at structuring a research area [7, 58]. Differently from *Systematic Literature Reviews*, SMSs are built on general questions to discover research trends. The quality assessment of primary studies is optional; for instance, primary studies without empirical evidence can be included. Being the analysis limited to the Italian scientific community, the present work deals with the national trends in the HPC and Cloud community. The 25 universities and research institutes participating in ICSC have been selected among the most active bodies in the international HPC arena. There are few *EuroHPC JU*[1] research projects that do not include one or more ICSC partners.

This report aims to document the ICSC ecosystem analytically, cataloguing current research efforts and investigating how they meet the real needs of application developers. In particular, it aims to answer three questions regarding current and future directions for research on Workflow Management Systems (WMSs):

- Q1 Which are the main research directions for WMSs in the Computing Continuum?
- Q2 Which research directions are widespread in the scientific community?
- Q3 Which research directions address a critical need for modern scientific applications?

We collected 25tools and 10scientific applications from several domains, and we (manually) classified them into five categories that segment quite well the primary emerging research directions (Sec. 2): 1) Interactive Computing, 2) Orchestration, 3) Energy efficiency, 4) Performance portability, and 5) Big Data management.

Then, application developers were asked to select the tools they deemed helpful in improving execution in a Computing Continuum environment (Sec. 3). The results of this survey, i.e., the answers to the previous research questions, are reported in Sec. 4, while Sec. 5 concludes the report by sketching future collaborations.

1.1 The National Research Centre for HPC, Big Data, and Quantum Computing

The National Research Centre for High-Performance Computing, Big Data, and Quantum Computing (ICSC) is one of the five Italian research champions, recently funded by NextGenerationEU with 320M€ (2022-26). It is organized according to a Hub&Spoke model, where the Hub has coordination functions and the Spokes are scientific departments participated by universities and companies throughout the national territory. ICSC has 11 Spokes representing different areas of HPC. Each Spoke has two main research topics and two scientific leaders.

Spoke 0 – Supercomputing Cloud infrastructure aims to implement supercomputing infrastructures capable of competing globally and providing computational resources to the European scientific community. The extension of the 238 PFLOPS Leonardo supercomputer with an additional 100 PFLOPS modules ("Mona Lisa," co-funded by EuroHPC JU) is one of the ongoing action items of Spoke 0.

Spoke 1 – FutureHPC & Big Data and *Spoke 10 – Quantum Computing* are focused on the platforms, methods, and tools of future computing. Spoke 1 has an evolutionary vision of next-generation systems, which will be based on classic architectures and methods consistent with the Turing Machine model. Spoke 10 looks to a more distant and necessarily more uncertain future. Quantum Computing promises a possible revolution in computation that goes beyond the limits of classical physics to which the Turing machine and its computations are bound.

The remaining Spokes are focused on scientific areas that most benefit from HPC: *Spoke 2 – Fundamental research & space economy*; *Spoke 3 – Astrophysics & cosmos observation*; *Spoke 4 – Earth & climate*; *Spoke 5 – Environment & natural disasters*; *Spoke 6 – Multiscale modelling & engineering applications*; *Spoke 7 – Material & molecular sciences*; *Spoke 8 – In-silico medicine & omics data*; *Spoke 9 – Digital society & smart cities*. All Spokes are described in detail on the ICSC website[2].

As sketched in Fig. 1, Spoke 1 is organized into 5 scientific Flagships (FLs) and two living labs. This work stems from *FL3) Workflows & I/O, Cloud-HPC convergence, digital twins* of Spoke 1, but gathers information on applications from all other Spokes. The objective of Spoke *1–FutureHPC & Big Data* is the creation of new labs as an integral part of a national federated centre on a global level with skills aimed at hardware and software co-planning and

[1]The European High-Performance Computing Joint Undertaking, https://eurohpc-ju.europa.eu/

[2]https://www.supercomputing-icsc.it

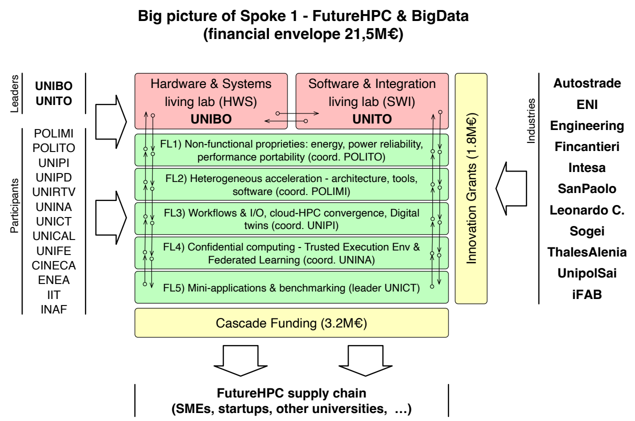

Figure 1: The Big picture of Spoke 1–FutureHPC & Big Data (total financial envelope 21,5M€).

enhancing Italian leadership in the EuroHPC JU, as well as in the ecosystem of data infrastructure for science and industry. [4].

1.2 The Italian ecosystem and beyond

This study only considers the Italian ICSC ecosystem and cannot be considered a survey of the state-of-the-art workflows at the international level. Instead, the study aims to discuss research directions in scientific workflows using the ICSC ecosystem as a statistical sample of international research on workflows.

Scientific workflows play an essential role in the whole EuroHPC JU funding program. For example, the ACROSS project[3] leverages pre-exascale infrastructures and effective mechanisms to easily describe and manage complex workflows in a diverse set of scientific domains, achieving high performance and energy efficiency, and the eFlows4HPC[4] project aims to create a European workflow platform to enable the design of complex applications that integrate HPC, Big Data and AI.

In the USA, a similar survey involving 15 partners of the *Exascale Computing Project* (ECP) [50] led to the definition of *ExaWorks* [2], a workflow Software Development Toolkit (SDK) consisting of a wide range of workflow management tools that can be composed and interoperated through standard interfaces. Despite similar in the scientific aim, e.g., in the definition of a common software stack (called *European Open Stack*) [28], the EuroHPC JU is structurally different from the American ECP project [50]: being a *Joint Undertaking*, each funded project requires the co-investment of the European Union and the countries that participate that project (or tender). For this, leading EU countries in the HPC arena need to mobilize national funding, which the EuroHPC JU can amplify. All European precursor-to-Exascale (Leonardo, LUMI, MareNostrum5), forthcoming Exascale platforms, and many funded research projects have been realized thanks to 50% EU and 50% national co-funding.

In addition to institutional research activities, community efforts are crucial in shaping future research directions. For example, the Workflow Community Initiative[5] is a community-centred effort for gathering and promoting long-standing and recent community-focused efforts in the field of scientific workflow research.

[3]https://www.acrossproject.eu/
[4]https://eflows4hpc.eu/
[5]https://workflows.community/

2 RESEARCH DIRECTIONS AND TOOLS

This report analyzes 25different tools from 9 Italian research institutions. Tools were collected among ICSC Spoke 1 partners in the context of FL3, which targets large-scale scientific workflows and their execution in the Computing Continuum (see Fig. 1).

We clustered the collected tools according to their principal research direction. All tools exhibit a primary direction, even if some cover multiple research topics. As reported in Table 1, five classes have been identified: interactive computing (Sec. 2.1), orchestration (Sec. 2.2), energy efficiency (Sec. 2.3), performance portability (Sec. 2.4), and Big Data management (Sec. 2.5). To answer questions 1 and 1, we discuss the main challenges related to workflow management in the Computing Continuum, and we briefly introduce the related tools for each class.

2.1 Interactive computing

The advent of the Cloud-based *-as-a-Service model significantly lowered the technical barriers to Cloud-based infrastructures, replacing Command Line Interfaces (CLIs) with user-friendly web-based dashboards, promoting an on-demand resource provisioning paradigm and adopting declarative public web APIs as the primary communication medium. Conversely, most HPC facilities expose only SSH-based remote shells, queue management systems, and air-gapped worker nodes. One of the challenges in scientific workflow management is to fill this gap, developing user-friendly interactive computing interfaces for HPC systems without compromising the critical features of these environments, namely, performance, security, and system-wide usage.

Given their widespread diffusion in diverse scientific domains, Jupyter Notebooks [70] are a promising technology enabling interactive workflows in HPC infrastructures. Jupyter Notebooks have been designed initially to support scientific computing, from interactive prototyping to publication [38]. Nowadays, Jupyter supports interactive computing in several languages through dedicated kernels and can be offered as a service on Cloud platforms, e.g., through JupyterHub[6] or Google Colaboratory[7]. Still, enabling Jupyter-based workflows as a service on HPC facilities poses three main challenges addressed by the collected tools.

First, interactive computing requires on-demand resource provisioning, while HPC facilities offer batched executions through queue managers. However, several batch systems (e.g., SLURM [74]) provide ways to access resources on-demand through *advanced reservations*. The BookedSlurm plugin introduces a methodology to easily create resource reservations through a web calendar and account for them under a pay-per-use mode using a digital currency.

Second, the standard transport layer (i.e., ZeroMQ) requires a bidirectional TCP connection between the publicly exposed front-end web server and the air-gapped worker nodes, commonly not allowed in HPC centres. The Interactive Computing Service (ICS) [19] integrates the Jupyter stack with the SLURM controller to interactively provide near-instantaneous access to HPC resources.

[6]https://jupyterhub.readthedocs.io/en/stable/
[7]https://colab.research.google.com

Table 1: Collected tools classified in five research directions.

Interactive computing	Orchestration	Energy efficiency	Performance portability	Big Data management
BookedSlurm	TORCH [72]	PESOS [16]	FastFlow [5]	ParSoDA [10]
ICS [19]	INDIGO [23]	Lapegna et al. [41]	Nethuns [12]	MALAGA
Jupyter Workflow [20]	Liqo [37]	De Lucia et al. [46]	INSANE [64]	aMLLibrary [33]
	StreamFlow [21]		CAPIO [47]	WindFlow [49]
	SPF [53]		BLEST-ML [14]	CHD [18]
	BDMaaS+[17]		MLIR [43]	Mingotti et al. [51]
	MoveQUIC [59]			

Third, the standard execution flow of Notebook cells is purely sequential, preventing users from modelling applications as workflow graphs. The Jupyter Workflow [20] kernel enables Jupyter Notebooks to describe and orchestrate complex distributed workflows, where each cell is seen as a step and inter-cell dependencies are extracted semi-automatically by inspecting the Abstract Syntax Tree (AST) of each code cell.

2.2 Orchestration

The micro-service and Function-as-a-Service (FaaS) paradigms foster modularity of applications and infrastructure-agnostic deployments, simplifying maintainability and enhancing portability. The FaaS paradigm recently percolated to Edge and Fog environments, pushing towards a Cloud-Edge Continuum. Hence, the serverless computing model attracts attention in scientific workflow management as a primary execution infrastructure [8, 62] or combined with HPC facilities in hybrid settings [65].

If this approach simplifies operations from the application developers' point of view, it moves all the deployment and life-cycle management aspects to the provider side. Hence, there is a need for advanced orchestration algorithms and tools capable of guaranteeing near real-time responses for function invocations independent of the underlying deployment infrastructure. When moving to the Continuum, additional aspects emerge. Function placement decisions become vital as invocation performance rises from the combination of available computing power and near-data processing. Plus, efficient migration strategies are crucial whenever data sources expose high dynamicity in their generation rate.

Seven of the collected tools target orchestration in the Computing Continuum. TORCH [72] and the INDIGO orchestrator [23] are TOSCA-based frameworks for deploying and orchestrating applications targeting multi-Cloud environments. Liqo [37] enables dynamic and seamless Kubernetes multi-cluster topologies. StreamFlow [21] orchestrates hybrid workflows on top of heterogeneous Cloud/HPC environments. SPF [53] is a Fog-as-a-Service platform targeting Smart City environments. BDMaaS+ [17] is a decision support tool for service providers who want to distribute an IT service on a global scale relying on private and public Cloud platforms. Finally, MoveQUIC [59] is a toolbox for the live migration of micro-services at the Edge.

2.3 Energy efficiency

Energy consumption is a key indicator in the whole spectrum of the Computing Continuum. On the HPC side, there is growing attention on measuring and reducing the carbon footprint of computational research [36, 39, 40], even if initiatives promoting sustainable HPC have existed for several years. For example, the Green500 list [29], which ranks supercomputers based on the amount of power needed to complete a fixed amount of work, dates back to 2007.

With the entire Computing Continuum available, a viable solution is to adopt energy-aware placement algorithms [44], which try to minimize the carbon footprint of workload executions without violating QoS requirements. Another possibility is to move computations on Edge sensors whenever possible. Besides relying on low-power hardware, this strategy also removes data transfers, saving additional energy. However, efficiently exploiting this class of devices requires resource-constrained algorithms and implementations.

Three of the collected tools address energy efficiency and low-power devices. PESOS [16] is an energy-efficient resource management algorithm for the placement of VMs in a Cloud environment, aiming to minimize the energy footprint of the overall platform while considering the QoS requirements of each VM. Lapegna et al. [41] investigate how to implement clustering algorithms on parallel and low-energy devices for Edge computing. De Lucia et al. [46] propose a technique to make hyperspectral image classification through convolutional neural networks affordable on low-power and high-performance sensor devices.

2.4 Performance portability

Performance portability can be defined as a measurement of an application's performance efficiency for a given problem that executes correctly on a set of platforms [57]. In practice, it derives from the composition of two opposing forces. Gaining portability across a set of diverse execution environments requires high-level abstractions, agnostic of the specific hardware stack that the application will target at runtime. On the other hand, maximizing performance requires a deep knowledge of the target execution architecture, such as the network topology and speed, the cache sizes, the amount of memory, and the presence or absence of high-end storage devices.

The heterogeneity of hardware accelerators that followed the end of Dennard scaling and the increasing modularity of modern scientific applications made performance portability libraries crucial for any large-scale scientific application that targets production usage. Commonly, a performance portability library is composed of two key elements. A *programming model* provides developers with abstractions between the application and one or more low-level resources, e.g., network, memory, storage, or data structures. Each abstraction is then translated into an *efficient implementation* optimized for a specific target execution environment, e.g., a high-end network with smart NICs, a high-bandwidth burst buffer, or a distributed data structure. Depending on the library, this translation can happen at compile time [27] or runtime [32].

Six of the collected tools target different aspects of performance portability. FastFlow [5] leverages the structured parallel programming methodology to define a single streaming dataflow programming model for shared-memory and distributed-memory systems. Nethuns [12] and INSANE [64] abstract the network layer, exposing a minimal set of communication primitives. CAPIO [48] provides a programmable file system in user space that intercepts the POSIX I/O system calls of an application, allowing users to target different devices and inject data streaming capabilities without modifying the existing codebase. BLEST-ML [14] leverages a machine learning algorithm to estimate a suitable block size for data partitioning in large-scale HPC infrastructures, optimizing data-parallel applications. Finally, MLIR [43] extends the LLVM toolchain with domain-specific middle-end representations to make compiler-level code optimizations more flexible.

2.5 Big Data management

With the advent of Big Data and the rise of Deep Learning, novel algorithms based on neural networks began to co-exist with standard simulation approaches in large-scale scientific workflows. If the training of huge neural models is still anchored to HPC facilities, data pre-processing and inference steps are becoming first-class citizens in geographically distributed Big Data pipelines, preferring near-data processing approaches for better performance and privacy. On the other hand, workflows are proving their worth in modelling and orchestrating Deep Learning pipelines [22, 45].

As in the case of traditional scientific workflows, most Big Data tools expose a dataflow paradigm [52]. However, the Big Data domain requires a higher expressive power of dataflow operators to support batch, micro-batch, and streaming execution models [1]. Also, advanced data structures [75] are fundamental to seamlessly enable distributed in-memory computations, ensuring near-data processing and avoiding I/O and data transfer overhead.

Moving Big Data analytics and Deep Learning pipelines to the Continuum poses new challenges. Data are commonly collected, filtered, and pre-processed at the Edge and moved to large data warehouses for parallel data mining and large models' training. Domain experts need tools and algorithms to support pluggable data processing operators for diverse data types, from images to graphs to geospatial information. At runtime, they should coordinate the execution of such operators from Edge to Cloud transparently, providing efficient implementations targeting multi-core and distributed architectures and exploiting heterogeneous hardware devices when available.

Six of the collected tools belong to the Big Data ecosystem. ParSoDA [10] is a Java programming library supporting parallel data mining applications executed on HPC systems. MALAGA is a Hadoop-compliant Java-based framework for multi-dimensional Big Data analytics over graph data. The aMLLibrary [33] is a high-level Python package that trains and optimizes multiple performance models using autoML, supporting feature selection and hyperparameter tuning. WindFlow [49] is a high-level library for continuous data stream processing on multi-core and hybrid CPU+GPU architectures. CHD [18] implements a parallel multi-density clustering approach to discover urban hotspots in a city. Finally, Mingotti et al. [51] implement a Real-Time Simulator (RTS) of a Phasor Measurement Unit (PMU) supporting hardware-in-the-loop (HIL) simulation techniques.

3 APPLICATIONS

This Section describes 10scientific applications from 11 ICSC partners. Application providers were asked to identify, among the tools listed in Sec. 2, those that they deemed valuable to improve the current status of their workload, with a specific focus on workflow execution in a Computing Continuum environment. The results of this selection are summarized in Table 2. Aiming to answer question 1, this Section provides, for each application, a brief description of the workload, the list of tools chosen for integration, and the reasons behind these choices.

3.1 Compression of petascale collections of textual and source-code files

The case study for this application is the Software Heritage initiative[8]. It aims to collect the complete history/heritage of human coding publicly available, replicate it massively to ensure its preservation and share it with everyone who needs it, from science to industry. The Software Heritage archive is reported to contain over 800 TB of data. Since the archive is steadily growing, the consequent impact on the scalability and storage cost of the archive and its mirrors is becoming a serious concern, not only in economic terms but also in terms of energy demands and the environmental impact of operating storage devices and replacing them when worn out. Compressing large collections of files is very challenging, and this problem was addressed in the past with various techniques.

The application is based on the so-called PPC paradigm [30]: Permuting + Partition + Compress, whose main algorithmic idea is to permute the files, bringing the "similar" files close to each other, partition them into blocks (of proper size); and eventually, compress each block with a suitable compressor (whose compression window is at least larger than the block size). This application aims to address the current limits of the PPC framework: it is based on a single-threaded implementation in Python, and it can manage GBs of data, but it cannot scale to TBs/PTs.

In order to address the mentioned challenges, the intention is to extend the compression libraries to a parallel and distributed scenario by adopting tools for parallel and distributed batch processing across clusters. *ParSoDA* can be used to define a data compression pipeline composed of three main phases: parallel sorting of files based on their filenames or other content- or context-based features; serialization and grouping of files in blocks of predefined size; parallel compression of those blocks of files by commodity or ad-hoc compressors. *FastFlow* can implement the entire data compression workflow expressing stream parallelism between the different phases, thus enabling their overlap. *WindFlow* can accelerate intra-node parallelization phases requiring complex streaming semantics and use HW accelerators.

[8]https://softwareheritage.org

Table 2: The list of collected scientific applications and the tools identified for integration.

		3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	3.10
Interactive computing	BookedSlurm										
	ICS		✓							✓	
	Jupyter Workflow		✓					✓			
Orchestration	TORCH										
	INDIGO				✓				✓		
	Liqo				✓				✓		
	StreamFlow		✓	✓							✓
	SPF										
	BDMaaS+							✓	✓		
	MoveQUIC				✓	✓					
Energy efficiency	PESOS					✓					
	Lapegna et al.										
	De Lucia et al.										
Performance portability	FastFlow	✓									
	Nethuns		✓				✓				
	INSANE										
	CAPIO		✓				✓				
	BLEST-ML										
	MLIR										✓
Big Data management	ParSoDA	✓								✓	
	MALAGA										
	aMLLibrary							✓		✓	
	WindFlow	✓									
	CHD										
	Mingotti et al.							✓			

3.2 Astrophysics data analysis and visualization

Over the years, the astrophysics domain has developed a set of ad-hoc tools and software modules to tackle the challenging particularities of the field. With the emergence of high-performance visualization and Visual Analytics as enabling technologies, some of these components become candidates to be replaced by either faster, more accurate, or more efficient data-driven technologies modelling pre-processing, runtime, and post-processing stages.

The tool considered for this application is VisIVO [35, 69], developed by adopting the Virtual Observatory standards. Its main objective is to perform 3D and multi-dimensional data analysis and knowledge discovery of unknown relationships between multivariate and complex astrophysical datasets. VisIVO requires three steps to render the visualization: data importing, filtering, and viewing. The importing process converts the supplied datasets (originally in different formats) into an internal binary format. The filtering process allows several operations on the data, e.g., randomization or decimation to reduce the final resolution, mathematical or statistical operators, or commonly adopted cosmological post-processing. Finally, the visualization process creates multidimensional views from the data that must fit the available RAM.

The evolution direction of VisIVO is tailored to pursue the following objectives: enhancing the portability of the VisIVO modular applications and their resource requirements; fostering reproducibility and maintainability; taking advantage of more flexible resource exploitation over heterogeneous HPC facilities (including mixed HPC-Cloud resources); minimizing data-movement overheads and improving I/O performances. For this, the integration of VisIVO with two groups of tools is considered. *StreamFlow*, *Jupyter Workflow*, and *ICS* allow a portable representation of the VisIVO modular applications and their resource requirements, foster reproducibility and maintainability, and allow taking advantage of heterogeneous HPC facilities while minimizing data-movement overheads. *CAPIO* and *Nethuns* can boost VisIVO I/O performances without modifying the original codebase and allow it to coordinate the I/O within its modules.

3.3 Genomic variant calling pipeline

This application aims to increase the flexibility of the prototype defined above by looking at its execution model. The objective is to remotely run the process on an HPC system by exploiting the *StreamFlow* WMS. In particular, the intention is to adapt the current implementation of the pipeline to the StreamFlow environment. StreamFlow allows the remote execution of the pipeline, making the whole execution more agile. The increased flexibility will allow testing the pipeline in several other (possibly heterogeneous) execution environments. Therefore, the integration outcome will provide fast provisioning and make it possible to evaluate the effects of system hardware/software aspects, such as the availability of GPUs or different storage and file systems in the host machines.

3.4 Edge-Cloud Continuum federation infrastructure

The observed evolution of the computing space warranted by networking suggests the emergence of a decentralized, federated, yet seamless organization. This prediction evokes the concept of the Continuum as a platform infrastructure where data processing may take place dynamically and where it is deemed most convenient under any of the criteria of interest to the end user (e.g., latency, privacy, and energy). This concept enables the traditional Internet and the Internet of Things to integrate into a seamless Continuum, where many *-as-a-service applications may be developed, deployed, and employed regardless of the location.

The architectural concept of this application assumes workflow executions that traverse service components deployed dynamically across compute-capable nodes of the Edge-to-Cloud Continuum. This vision gives rise to the notion of "dynamic orchestration," in which a workflow is specified in terms of required services, and the matching to provided services may be resolved dynamically based on user preferences, service levels, privacy, energy, and latency requirements. Therefore, deploying service components selected for the workflow may be dynamic and opportunistic, and may also contemplate mobility and migration.

Allowing compute bundles to migrate requires understanding that they may have ongoing communications with client endpoints, which may also share a connection state. This scenario requires server-side connection migration, which *MoveQUIC* may support. The next level up in the application system concept is the orchestrator control plane across federations of Compute nodes. In that direction, the intention is to incorporate a single cluster zone in larger federations using *Liqo*. Describing and deploying the user application as a dynamic orchestration of a workflow execution will need a flexible orchestration platform for which the plan is to explore the use of the *INDIGO* orchestrator.

3.5 Serverledge: QoS-Aware FaaS in the Edge-Cloud Continuum

Serverledge [66] has been designed to fill the gap between Edge and Cloud and provides a flexible and extensible framework for the FaaS paradigm in geographically distributed environments. Serverledge provides a suitable framework for low-latency FaaS execution in the Edge-Cloud Continuum. However, several challenges must be addressed to fully support QoS-aware execution and scheduling in such a dynamic environment. The two main directions in which Serverledge can evolve are improving the runtime management layer by providing live function migration and supporting energy-efficient orchestration.

While serverless functions usually have a short duration, long-running functions are gaining popularity as approaches for serverless data analytics and machine learning. The existence of such workloads at the edge calls for live migration mechanisms to migrate running instances, free up resources as needed, or, in general, respond to adaptation needs. Indeed, function migration to a different node can allow the system to revise initial scheduling decisions that become far from optimal over time, reschedule a resource-consuming and long-running function on a different node with more powerful resources, or support the smooth movement of mobile users during function execution. In this direction, the considered tool is *MOVEQuic*.

The other intention is to integrate Serverledge with the energy-efficient orchestration provided by *PESOS*. The goal is to provide holistic energy-efficient management from FaaS execution at the Edge to FaaS frameworks deployed in Cloud data centres. This goal is achieved by managing how FaaS resources are redirected to different Cloud nodes and by taking into account the current load of each node and trying to consolidate the allocation of resources to power off some of the nodes whenever possible. PESOS will be extended and integrated with the Serverledge toolkit to manage heterogeneous Cloud and Edge nodes.

3.6 Improving I/O phases in computational modelling of Galaxy Formation

The formation and evolution of galaxies and Supermassive Black Holes at their centres is a central theme of contemporary Astrophysics and Cosmology. Numerical modelling of this problem has proven to be challenging due to the long-range nature of the gravitational interaction, which cannot be shielded [34]. For these reasons, the computational complexity of algorithms devised to model galaxy formation and evolution poses challenging problems when implemented in parallel codes. Astrophysical codes are often adopted as testbeds of new hardware architectures, as they can challenge their scaling capabilities.

The considered application is designed to cope with a well-known state-of-the-art parallel code, FLASH [31]. This code implements a spatial and temporal partition based on an Adaptive Mesh Refinement decomposition and a rather sophisticated hierarchical tree scheme to deal with the long-range gravitational interactions. Despite its modular architecture, allowing considerable flexibility in designing target-specific numerical experiments and physical simulations is a very complex task. Plus, some related tasks are not supported by the FLASH library. Instead of producing new modules in FLASH to undertake these tasks, it is more convenient to glue FLASH together with other packages specifically designed to resolve the desired computation, e.g., SYGMA [63].

This application implements a workflow in which FLASH and SYGMA run concurrently and asynchronously to perform different tasks, periodically synchronizing their outputs for physical consistency. Inside FLASH, the paths to the libraries (including those specific to I/O) are stored in a site-specific configuration file with compilation and link-specific options. The main intention of this application is to improve the I/O of both the large datasets produced by FLASH and during data exchange between FLASH and SYGMA. More specifically, *CAPIO* middleware will be used to boost the I/O performance of the FLASH-SYGMA workflow without modifying the original codes, and *Nethuns* will be used to improve the I/O of the large data outputs produced by FLASH, i.e., of both checkpoints and data files. This aspect is particularly relevant, as parallel I/O is seen as one of the major bottlenecks in exploiting FLASH capabilities on future exascale architectures.

3.7 WorldDynamics.jl

WorldDynamics.jl is an open-source Julia package that provides a modern framework to investigate Integrated Assessment Models

(IAMs) of sustainable development benefiting from Julia's ecosystem for scientific computing. An IAM aims to integrate the critical aspects of society and economy with the biosphere and atmosphere within a unified modelling framework, providing informed policymaking in different contexts such as climate change, human development, and social development. The goal of WorldDynamics.jl is to allow users to easily use and adapt different IAMs, from World3 to recent proposals.

This application, available at [24], is a modern framework based on current software engineering and Machine Learning techniques to investigate models of global dynamics focused on sustainable development. Among its features, it can recreate all the figures found in books that detail the World1, World2, and World3 models, perform sensitivity analysis by adjusting the initial values of variables, and analyze alternative scenarios by modifying either the model's parameters or the interpolation tables, which are utilized to approximate non-linear functions through linear segments.

WorldDynamics.jl would benefit from more readable access to its models and workflows. Integration with *Jupyter Workflow* would make WolrdDynamics.jl more accessible and allow model executions in a distributed fashion, improving the performance and readability while exploiting HPC architectures. On the other hand, integration with *BDMaaS+* would speed up the process and run different models (and simulations) in a parallel fashion.

In addition, the intention is to investigate Machine Learning techniques and the data furnished by real-time simulators. The *aMLLibrary* already implements regression, which is the base case of WorlDynamics.jl model discovery algorithm. Furthermore, aMLLibrary can make a better exploitation of the available data. Also, the tool *Mingotti et al.* proposed can be plugged into the application model as a source of new data and, hence equations. Indeed, the models proposed until now are generally based on coarse-grain data. In contrast, the tool proposed by Mingotti et al. allows for building a more precise subsystem alongside extrapolating valuable data for global consumption.

3.8 Optimized deployment of Cloud-native applications in the Cloud Continuum

Multi-Cloud and Cloud Continuum scenarios refer to many interconnected computing resources consisting of Cloud, Edge, and on-premises resources, usually located in different locations with possibly different ownership, renting prices, and sizes. The high heterogeneity of the Cloud Continuum introduces many challenges from the service management perspective. One of them is the identification of optimized deployments for HPC applications, which require high computational resources. On the one hand, to minimize the overall provisioning costs, a service provider would like to deploy its application by analyzing the pricing perspective, thus looking at the renting prices of the chosen execution environment, such as Kubernetes clusters or vanilla Virtual Machines (VMs). On the other hand, communication latencies between different computing locations might play a crucial role in assessing the performance of complex workflows, such as the ones of HPC applications. The state-of-the-art orchestration components are not designed to consider all these requirements and to fully benefit from the capabilities of multi-Cloud and Cloud continuum scenarios.

This application is designed to solve these challenges by presenting a novel approach integrating three different tools: *BDMaaS+*, *INDIGO*, and *Liqo*. This integration aims at enabling an optimized deployment of complex Cloud-native applications over multi-Cloud and cloud continuum scenarios by exploiting the capabilities of multiple and distributed computing clusters. To illustrate the application design, the considered use case describes a provider interested in deploying an HPC application. First, the provider needs to describe the application case and its workflow using the standardized TOSCA notation. Then, the INDIGO orchestrator interacts with BDMaaS+ to find the most appropriate computing resources considering the application requirements, provider-defined policies (pricing, latency), and the current availability of resources among the multi-Cloud. After that, BDMaaS+ returns to INDIGO the information on the computing resources that can sustain the QoS demanded by the composite Cloud application described in the TOSCA blueprint.

Given such fresh data, the INDIGO orchestrator produces an application deployment plan that includes Kubernetes "intents." The orchestrator will then enforce the application provisioning by invoking the Liqo API and providing it with the Kubernetes intents defined above. Guided by the deployment requests issued by the INDIGO orchestrator, Liqo dynamically creates a federation of networked computing resources. It will then instantiate, configure, and run the application's distributed components in the federation.

3.9 Anomalous subgroup characterization with DivExplorer

DivExplorer [55, 56] is an automatic approach for exploring datasets and finding subgroups of data for which a model behaves anomalously. The notion of divergence is introduced to estimate the different classification behaviours in data subgroups concerning the overall behaviour. Subgroups are characterized by attribute values, making the subgroups directly interpretable. DivExplorer effectively integrates performance and divergence into the exploration process, leveraging frequent pattern mining algorithms and enabling efficient exploration of all subgroups with adequate representation in the dataset. The approach also handles hierarchical subgroup exploration over data hierarchies [54] and proved its effectiveness also on complex models such as speech ones.

This application represents a comprehensive framework to analyze the behaviour of Machine Learning models, focusing on peculiarities at the subgroup level. Having as its core component the DivExplorer approach, it covers multiple directions: handling Big Data and Big Data models, generalizing to multiple tasks and models, proposing novel methodologies for model comparison and selection, leveraging subgroup analysis for model improvement, and integrating subgroup analysis into interactive frameworks enabling access to computational data on an HPC system.

The intention is to investigate the integration of DivExplorer with three different tools. In the *aMLLibrary*, the analysis of performance at the subgroup could be a relevant component for validating and choosing the best regression model. Therefore, integrating DivExplorer and aMLLibrary would enable a comprehensive model comparison and selection for the regression task. The *Interactive Computing Service* would enable access to computational data on an

HPC system, enabling a seamless workflow capable of streamlining the analysis process. The subgroup analysis functionality would be directly accessible in the Jupyter launcher of the IAC interface. By accessing subgroup analysis via IAC, users can easily access and analyze relevant data, build models, and explore subgroup-specific behaviour in a unified and cohesive manner. Finally, integrating the Big Data implementation of DivExplorer with the *ParSoDA* library would simplify the development of parallel data mining applications executed on HPC systems by using the set of functions of ParSoDA for processing and analyzing data.

3.10 Compilation flow and deployment strategy targeting HPC RISC-V accelerators

Generating efficient executable code for HPC and Cloud computing architectures is complex. Compiler toolchains play a crucial role in providing techniques and methodologies to achieve optimal workload mapping. This scenario poses a severe challenge for efficient compiler design. The high-level structure of a standard compiler toolchain includes three main stages: front-end, middle-end, and back-end. The front-end stage recognizes legal programs and produces an intermediate representation (IR) with an abstraction level suitable for later transformations. The middle-end and back-end optimization passes transform an input program representation into an equivalent one optimized for a target metric (e.g., speed, size, or safety), and the design of the IR language must simplify this goal, adopting machine-independent or machine-specific knowledge, respectively.

In recent years, compiler researchers and companies have explored an approach based on Multi-Level Intermediate Representation (MLIR) [43]. MLIR introduces a set of domain-specific middle-end representations geared toward domain-specific optimizations, allowing different levels of abstraction to co-exist freely using a uniform IR grammar.

This application aims to demonstrate the MLIR flow into an HPC environment, supporting high-level workloads targeting experimental RISC-V accelerators. The specification and initial design of MLIR abstractions for a RISC-V accelerator for HPC computing are completed. A research prototype is available as an open-source project [76, 77]. The application prototype will implement the low-level representations required to target a RISC-V accelerator and the related transformations from high-level representations down to LLVM [42] IR. It will also manage the orchestration of the optimization flow. *StreamFlow* is chosen as a workflow management tool to orchestrate the MLIR transformation steps.

4 DISCUSSION

This Section quantitatively analyzes the collected data to answer the three research questions introduced in Sec. 1. Despite being specifically related to the Italian research community, most of what is discussed below aligns with the future directions of scientific workflows outlined in recent literature for the whole research ecosystem.

Q1. Which are the main research directions for WMSs in the Computing Continuum?

This report identifies five main research directions: interactive computing, orchestration, energy efficiency, performance portability, and Big Data management. These findings overlap with recent literature's beliefs about the future directions of scientific workflows. Ben-Nun et al. [11] identify performance portability as a crucial aspect of Exascale applications. They propose workflows as a programming model to achieve it, aided by advanced orchestration tools and high-level interfaces (which fall under the broader scope of "productivity"). Dube et al. [26] identify performance portability across heterogeneous devices and High-Performance Data Analytics in the Continuum as crucial ingredients to realize an "Internet of Workflows," i.e., to dynamically compose workflows and deploy them across multiple organizational and geographical boundaries. The 2022 Workflow Community Summit report [25] identifies "Workflows for Continuum and Cross-Facility Computing" as a critical research direction for the workflow community. It underlines how Big Data management and intelligent orchestration represent crucial challenges in this topic.

The presence of energy efficiency as a key research direction is interesting. Despite several works in the literature about energy-efficient scheduling of scientific workflows [13, 15, 67], this topic is rarely seen as a first-class citizen in workflow research's roadmap. However, orchestrating workflows in the full spectrum of Computing Continuum means including low-power Edge devices and sensors as potential execution locations. This requires including energy consumption in workflow requirements and developing a new class of energy-aware WMSs

On the other hand, the absence of tools covering crucial aspects of distributed workflow orchestration, such as performance monitoring, provenance collection, fault tolerance, and security, is unexpected. Properly handling all these aspects will be a relevant goal for the project's subsequent phases.

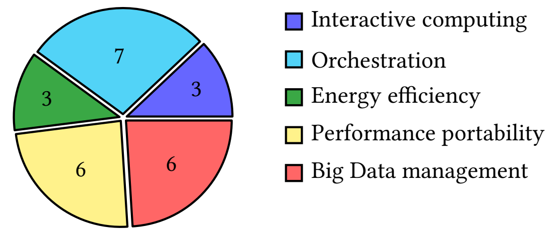

Figure 2: Pie chart reporting the tool distribution over the five identified research domains.

Q2. Which research directions are widespread in the scientific community?

Fig. 2 shows the distribution of collected tools over the five identified research domains. The effort is quite balanced among the different research directions, with 3 tools covering interactive computing and energy efficiency (12%) and 7 addressing diverse orchestration aspects (28%). These data imply that there is no single, predominant research line in distributed workflow management but several independent efforts that contribute to scientific advancement.

Fig. 3 shows how many research directions are covered by the tools provided by a single institution. It is worth noting that more than half of the involved institutions cover a single research topic, and no institutions span the whole set of identified directions. These

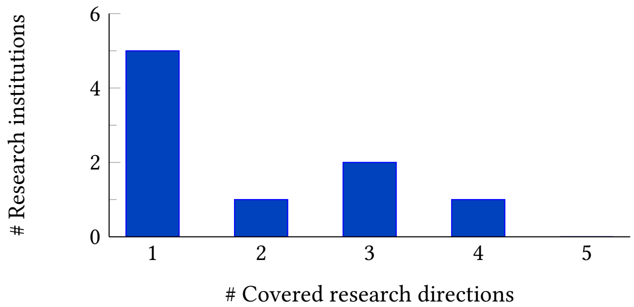

Figure 3: Histogram reporting how many research directions are covered by the tools provided by a single institution.

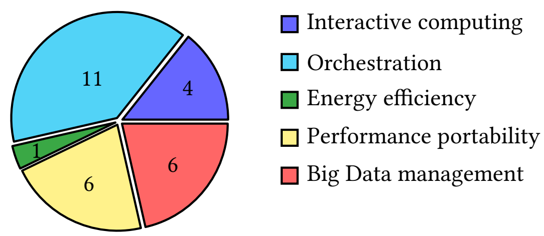

Figure 4: Pie chart reporting how the tools selected for integration distribute over the five identified research domains.

findings underline how collaborative initiatives are crucial for providing direct links between highly specialized groups and building a holistic research ecosystem around scientific workflows.

Q3. Which research directions address a critical need for modern scientific applications?

Fig. 4 shows how the tools selected by each application provider are distributed over the five identified research domains. Compared with Fig. 2, the distribution here is much more unbalanced, ranging from a single vote (below 3.6%) in the energy efficiency domain to 11 votes (above 39%) in the orchestration domain.

Developing a solid orchestration infrastructure targeting Computing Continuum is critical for current scientific workloads. Flexibility, intended as portability across potentially heterogeneous computing environments, and the possibility to create dynamic federations stand out as the most desirable properties of such infrastructure. Along this line, advanced placement algorithms such as BDMaaS+ [17] are perceived as crucial to optimize resource allocations in such complex environments.

Interactive computing, performance portability, and Big Data management have aroused significant interest, with at least three application providers identifying them as critical needs (see Table 2). Jupyter Workflow [20] and the Interactive Computing Service [19] are seen as a way to increase application accessibility to the end users through high-level interfaces to HPC centres. Performance portability libraries can improve performance with little to no modification to the codebase, i.e., without affecting maintainability. Along this line, high-performance Big Data runtimes like ParSoDA [10], and WindFlow [49] can inject data parallelism in scientific workloads without significantly modifying existing data structures.

The little interest in energy efficiency that emerges from Fig. 2 is partly justified by the fact that two out of three collected tools target specific algorithms (i.e., clustering [41] and hyperspectral image classification [46]), reducing their applicability to a limited range of applications. Only Serverledge enumerates energy-efficient orchestration in the Continuum as a critical need (see Sec. 3.5).

5 CONCLUSION AND FUTURE WORK

Aiming to outline the main current research directions in the Italian community and determine how they address the critical needs of modern scientific applications, this work collected and analyzed 25 tools and 10 applications from several Italian research institutions in the context of the ICSC Spoke 1 initiative.

The tools were clustered into five research directions, i.e., interactive computing, orchestration, energy efficiency, performance portability, and Big Data management. These directions overlap with recent literature's beliefs about future directions of scientific workflows, with the interesting exception of energy efficiency. Indeed, despite being crucial to target both Edge Computing and next-generation supercomputing platforms, energy efficiency is rarely perceived as a first-class citizen in WMSs' development roadmap.

Applications providers were asked to identify the tools they deemed valuable to improve the current status of their workload, with a specific focus on workflow execution in the Compute Continuum. The results of this selection highlight a prominent interest in advanced workflow orchestration and a still significant interest in all research directions but energy efficiency, partly due to the domain-specific nature of the collected tools.

The next phases of the Spoke 1 of ICSC work program will be focused on implementing the proposed tool integrations and testing them on pipelines from applicative Spokes and industries. This is expected to produce two scientific results. Firstly, to select tools that are suitable to be integrated into the larger software ecosystems, for example, the European Open Stack promoted by EuroHPC JU [28]. Secondly, to evaluate the actual added value the tools and their integration bring to the scientific workflow ecosystem. This evaluation will be the subject of the forthcoming project deliverables and follow-up scientific reports. As a byproduct, the close cooperation between different research institutions will contribute to creating a solid workflow research community at the national level, with this report serving as a primer for new researchers and institutions that may wish to join.

ICSC also targets cooperation with the industrial sector, meaning that additional applications with higher Technology Readiness Levels (TRLs) will be introduced in the ecosystem. Assessing if industrial and academic applications share the exact critical needs will be an interesting subject for further investigation.

ACKNOWLEDGMENTS

This work has been supported by ICSC – Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU.

REFERENCES

[1] Tyler Akidau, Robert Bradshaw, Craig Chambers, Slava Chernyak, Rafael Fernández-Moctezuma, Reuven Lax, et al. 2015. The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing. *Proceedings of the VLDB Endowment* 8, 12 (2015), 1792–1803. https://doi.org/10.14778/2824032.2824076

[2] Aymen Al-Saadi, Dong H. Ahn, Yadu N. Babuji, Kyle Chard, James Corbett, Mihael Hategan, et al. 2021. ExaWorks: Workflows for Exascale. In *2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS)* (St. Louis, MO, USA). IEEE, 50–57. https://doi.org/10.1109/WORKS54523.2021.00012

[3] Sadaf R. Alam, Javier Bartolome, Sanzio Bassini, Michele Carpenè, Mirko Cestari, Frederic Combeau, et al. 2020. Archival Data Repository Services to Enable HPC and Cloud Workflows in a Federated Research e-Infrastructure. In *IEEE/ACM International Workshop on Interoperability of Supercomputing and Cloud Technologies, SuperCompCloud@SC 2020* (Atlanta, GA, USA). IEEE, 39–44. https://doi.org/10.1109/SuperCompCloud51944.2020.00012

[4] Marco Aldinucci, Giovanni Agosta, Antonio Andreini, Claudio A. Ardagna, Andrea Bartolini, Alessandro Cilardo, et al. 2021. The Italian research on HPC key technologies across EuroHPC. In *ACM Computing Frontiers* (Virtual Conference, Italy). ACM, 279–286. https://doi.org/10.1145/3457388.3458508

[5] Marco Aldinucci, Marco Danelutto, Peter Kilpatrick, and Massimo Torquati. 2017. *FastFlow: high-level and efficient streaming on multi-core.* John Wiley & Sons, Ltd, Chapter 13, 261–280. https://doi.org/10.1002/9781119332015.ch13

[6] Marco Aldinucci, Sergio Rabellino, Marco Pironti, Filippo Spiga, Paolo Viviani, Maurizio Drocco, et al. 2018. HPC4AI: an AI-on-demand federated platform endeavour. In *Proceedings of the 15th ACM International Conference on Computing Frontiers, CF 2018* (Ischia, Italy). ACM, 279–286. https://doi.org/10.1145/3203217.3205340

[7] Vasco Amaral, Beatriz Norberto, Miguel Goulão, Marco Aldinucci, Siegfried Benkner, Andrea Bracciali, et al. 2020. Programming languages for data-Intensive HPC applications: A systematic mapping study. *Parallel Comput.* 91 (2020), 102584. https://doi.org/10.1016/j.parco.2019.102584

[8] Aitor Arjona, Pedro García López, Josep Sampé, Aleksander Slominski, and Lionel Villard. 2021. Triggerflow: Trigger-based orchestration of serverless workflows. *Future Generation Computer Systems* 124 (2021), 215–229. https://doi.org/10.1016/j.future.2021.06.004

[9] Daniel Balouek-Thomert, Eduard Gibert Renart, Ali Reza Zamani, Anthony Simonet, and Manish Parashar. 2019. Towards a computing continuum: Enabling edge-to-cloud integration for data-driven workflows. *International Journal of High Performance Computing Applications* 33, 6 (2019), 1159–1174. https://doi.org/10.1177/1094342019877383

[10] Loris Belcastro, Fabrizio Marozzo, Domenico Talia, and Paolo Trunfio. 2019. ParSoDA: high-level parallel programming for social data mining. *Social Network Analysis and Mining* 9, 1 (2019), 4:1–4:19. https://doi.org/10.1007/s13278-018-0547-5

[11] Tal Ben-Nun, Todd Gamblin, Daisy S. Hollman, Hari Krishnan, and Chris J. Newburn. 2020. Workflows are the New Applications: Challenges in Performance, Portability, and Productivity. In *IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC, P3HPC@SC 2020* (Atlanta, GA, USA). IEEE, 57–69. https://doi.org/10.1109/P3HPC51967.2020.00011

[12] Nicola Bonelli, Fabio Del Vigna, Alessandra Fais, Giuseppe Lettieri, and Gregorio Procissi. 2022. Programming socket-independent network functions with nethuns. *SIGCOMM Computer Communication Review* 52, 2 (2022), 35–48. https://doi.org/10.1145/3544912.3544917

[13] Khadija Bousselmi, Zaki Brahmi, and Mohamed Mohsen Gammoudi. 2016. Energy Efficient Partitioning and Scheduling Approach for Scientific Workflows in the Cloud. In *IEEE International Conference on Services Computing, SCC 2016* (San Francisco, CA, USA). IEEE Computer Society, 146–154. https://doi.org/10.1109/SCC.2016.26

[14] Riccardo Cantini, Fabrizio Marozzo, Alessio Orsino, Domenico Talia, Paolo Trunfio, Rosa M. Badia, et al. 2022. Block size estimation for data partitioning in HPC applications using machine learning techniques. *CoRR* abs/2211.10819 (2022), 10 pages. https://doi.org/10.48550/arXiv.2211.10819 arXiv:2211.10819

[15] Fei Cao, Michelle M. Zhu, and Chase Q. Wu. 2014. Energy-Efficient Resource Management for Scientific Workflows in Clouds. In *2014 IEEE World Congress on Services, SERVICES 2014* (Anchorage, AK, USA). IEEE Computer Society, 402–409. https://doi.org/10.1109/SERVICES.2014.76

[16] Matteo Catena and Nicola Tonellotto. 2017. Energy-Efficient Query Processing in Web Search Engines. *IEEE Transactions on Knowledge and Data Engineering* 29, 7 (2017), 1412–1425. https://doi.org/10.1109/TKDE.2017.2681279

[17] Walter Cerroni, Luca Foschini, Genady Ya Grabarnik, Filippo Poltronieri, Larisa Shwartz, Cesare Stefanelli, and Mauro Tortonesi. 2022. BDMaaS+: Business-Driven and Simulation-Based Optimization of IT Services in the Hybrid Cloud. *IEEE Transactions on Network and Service Management* 19, 1 (2022), 322–337. https://doi.org/10.1109/TNSM.2021.3110139

[18] Eugenio Cesario, Paschal I. Uchubilo, Andrea Vinci, and Xiaotian Zhu. 2022. Multi-density urban hotspots detection in smart cities: A data-driven approach and experiments. *Pervasive and Mobile Computing* 86 (2022), 101687. https://doi.org/10.1016/j.pmcj.2022.101687

[19] CINECA. Accessed: 2023-08-16. Interactive Computing Service (IAC). https://wiki.u-gov.it/confluence/display/SCAIUS/UG2.7.2%3A+Interactive+Computing

[20] Iacopo Colonnelli, Marco Aldinucci, Barbara Cantalupo, Luca Padovani, Sergio Rabellino, Concetto Spampinato, et al. 2022. Distributed workflows with Jupyter. *Future Generation Computer Systems* 128 (2022), 282–298. https://doi.org/10.1016/j.future.2021.10.007

[21] Iacopo Colonnelli, Barbara Cantalupo, Ivan Merelli, and Marco Aldinucci. 2021. StreamFlow: cross-breeding cloud with HPC. *IEEE Transactions on Emerging Topics in Computing* 9, 4 (2021), 1723–1737. https://doi.org/10.1109/TETC.2020.3019202

[22] Iacopo Colonnelli, Bruno Casella, Gianluca Mittone, Yasir Arfat, Barbara Cantalupo, Roberto Esposito, et al. 2022. Federated Learning meets HPC and cloud. In *Machine Learning for Astrophysics: Proceedings of the ML4Astro International Conference* (Catania, Italy) *(Astrophysics and Space Science Proceedings)*. Springer Cham.

[23] Alessandro Costantini, Giuseppe Di Modica, Jean Christian Ahouangonou, Doina Cristina Duma, Barbara Martelli, Matteo Galletti, et al. 2022. IoTwins: Toward Implementation of Distributed Digital Twins in Industry 4.0 Settings. *Computers* 11, 5 (2022), 67. https://doi.org/10.3390/computers11050067

[24] Pierluigi Crescenzi, Emanuele Natale, and Paulo Bruno Serafim. 2022. *WorldDynamics.jl: v0.1.0.* https://doi.org/10.5281/zenodo.7093580

[25] Rafael Ferreira da Silva, Rosa M. Badia, Venkat Bala, Debbie Bard, Peer-Timo Bremer, Ian Buckley, et al. 2023. Workflows Community Summit 2022: A Roadmap Revolution. *CoRR* abs/2304.00019 (2023), 31 pages. https://doi.org/10.48550/arXiv.2304.00019 arXiv:2304.00019

[26] Nicolas Dubé, Duncan Roweth, Paolo Faraboschi, and Dejan S. Milojicic. 2021. Future of HPC: The Internet of Workflows. *IEEE Internet Computing* 25, 5 (2021), 26–34. https://doi.org/10.1109/MIC.2021.3103236

[27] H. Carter Edwards, Christian R. Trott, and Daniel Sunderland. 2014. Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. *J. Parallel and Distrib. Comput.* 74, 12 (2014), 3202–3216. https://doi.org/10.1016/j.jpdc.2014.07.003

[28] EuroHPC JU 2023. Multi-Annual Strategic Programme 2021-2027 (version 2023). EuroHPC Joint Undertaking. https://eurohpc-ju.europa.eu/system/files/2023-06/Decision%2008_2023_%20Amendment%20MASP%202021-2027_0.pdf

[29] Wu-chun Feng and Kirk W. Cameron. 2007. The Green500 List: Encouraging Sustainable Supercomputing. *Computer* 40, 12 (2007), 50–55. https://doi.org/10.1109/MC.2007.445

[30] Paolo Ferragina and Giovanni Manzini. 2010. On compressing the textual web. In *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010* (New York, NY, USA). ACM, 391–400. https://doi.org/10.1145/1718487.1718536

[31] Bruce Fryxell, Kevin Olson, Paul M. Ricker, Francis X. Timmes, Michael Zingale, DDonald Q. Lamb, et al. 2000. FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes. *The Astrophysical Journal Supplement Series* 131, 1 (nov 2000), 273. https://doi.org/10.1086/317361

[32] Edgar Gabriel, Saber Feki, Katharina Benkert, and Michael M. Resch. 2010. Towards performance portability through runtime adaptation for high-performance computing applications. *Concurrency and Computation: Practice and Experience* 22, 16 (2010), 2230–2246. https://doi.org/10.1002/cpe.1586

[33] Enrico Galimberti, Bruno Guindani, Federica Filippini, Hamta Sedghani, Danilo Ardagna, Germán Moltó, and Miguel Caballer. 2023. OSCAR-P and aMLLibrary: Performance Profiling and Prediction of Computing Continua Applications. In *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering, ICPE 2023* (Coimbra, Portugal). ACM, 139–146. https://doi.org/10.1145/3578245.3584941

[34] Roger W. Hockney and James W. Eastwood. 1988. *Computer simulation using particles.* Bristol: Hilger, 1988.

[35] INAF. 2015. VisIVO. https://visivo.readthedocs.io/

[36] Loïc Lannelongueand Michael Inouye. 2023. Carbon footprint estimation for computational research. *Nature Reviews Methods Primers* 3, 1 (16 Feb 2023), 9. https://doi.org/10.1038/s43586-023-00202-5

[37] Marco Iorio, Fulvio Risso, Alex Palesandro, Leonardo Camiciotti, and Antonio Manzalini. 2022. Computing Without Borders: The Way Towards Liquid Computing. *IEEE Transactions on Cloud Computing* 11, 3 (2022), 2820–2838. https://doi.org/10.1109/TCC.2022.3229163

[38] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E. Granger, Matthias Bussonnier, Jonathan Frederic, et al. 2016. Jupyter Notebooks - a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas, 20th International Conference on Electronic Publishing.* IOS Press, Göttingen, Germany, 87–90. https://doi.org/10.3233/978-1-61499-649-1-87

[39] Loïc Lannelongue. 2022. Carbon footprint: the (not so) hidden cost of high performance computing. *ITNOW* 63, 4 (Jan 2022), 12–13. https://doi.org/10.1093/itnow/bwab100

[40] Loïc Lannelongue, Jason Grealey, and Michael Inouye. 2021. Green Algorithms: Quantifying the Carbon Footprint of Computation. *Advanced Science* 8, 12 (2021), 2100707. https://doi.org/10.1002/advs.202100707

[41] Marco Lapegna, Walter Balzano, Norbert Meyer, and Diego Romano. 2021. Clustering Algorithms on Low-Power and High-Performance Devices for Edge Computing Environments. *Sensors* 21, 16 (2021), 5395. https://doi.org/10.3390/s21165395

[42] Chris Lattner and Vikram S. Adve. 2004. LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation. In *2nd IEEE / ACM International Symposium on Code Generation and Optimization, CGO 2004* (San Jose, CA, USA). IEEE Computer Society, 75–88. https://doi.org/10.1109/CGO.2004.1281665

[43] Chris Lattner, Mehdi Amini, Uday Bondhugula, Albert Cohen, Andy Davis, Jacques A. Pienaar, et al. 2021. MLIR: Scaling Compiler Infrastructure for Domain Specific Computation. In *IEEE/ACM International Symposium on Code Generation and Optimization, CGO 2021* (Seoul, South Korea). IEEE, 2–14. https://doi.org/10.1109/CGO51591.2021.9370308

[44] Yuanzhe Li and Shangguang Wang. 2018. An Energy-Aware Edge Server Placement Algorithm in Mobile Edge Computing. In *2018 IEEE International Conference on Edge Computing, EDGE 2018* (San Francisco, CA, USA). IEEE Computer Society, 66–73. https://doi.org/10.1109/EDGE.2018.00016

[45] Junwen Liu, Ziyun Xiao, Shiyong Lu, and Dunren Che. 2021. Deep-Learning-as-a-Workflow (DLaaW): An Innovative Approach to Enabling Deep Learning in Scientific Workflows. In *2021 IEEE International Conference on Big Data (Big Data)* (Orlando, FL, USA). IEEE, 3101–3106. https://doi.org/10.1109/BigData52589.2021.9671626

[46] Gianluca De Lucia, Marco Lapegna, and Diego Romano. 2023. A GPU Accelerated Hyperspectral 3D Convolutional Neural Network Classification at the Edge with Principal Component Analysis Preprocessing. In *Parallel Processing and Applied Mathematics*. Springer International Publishing, Cham, 127–138.

[47] Alberto Riccardo Martinelli, Massimo Torquati, and Marco Aldinucci. 2020. CAPIO. https://github.com/alpha-unito/capio

[48] Alberto Riccardo Martinelli, Massimo Torquati, Iacopo Colonnelli, Barbara Cantalupo, and Marco Aldinucci. 2023. CAPIO: a Middleware for Transparent I/O Streaming in Data-Intensive Workflows. In *30th IEEE International Conference on High Performance Computing, Data, and Analytics, HiPC 2023* (Goa, India). IEEE.

[49] Gabriele Mencagli, Massimo Torquati, Andrea Cardaci, Alessandra Fais, Luca Rinaldi, and Marco Danelutto. 2021. WindFlow: High-Speed Continuous Stream Processing With Parallel Building Blocks. *IEEE Transactions on Parallel and Distributed Systems* 32, 11 (2021), 2748–2763. https://doi.org/10.1109/TPDS.2021.3073970

[50] Paul Messina. 2017. The Exascale Computing Project. *Computing in Science & Engineering* 19, 3 (2017), 63–67. https://doi.org/10.1109/MCSE.2017.57

[51] Alessandro Mingotti, Federica Costa, Diego Cavaliere, Lorenzo Peretto, and Roberto Tinarelli. 2021. On the Importance of Characterizing Virtual PMUs for Hardware-in-the-Loop and Digital Twin Applications. *Sensors* 21, 18 (2021), 6133. https://doi.org/10.3390/s21186133

[52] Claudia Misale, Maurizio Drocco, Marco Aldinucci, and Guy Tremblay. 2017. A Comparison of Big Data Frameworks on a Layered Dataflow Model. *Parallel Processing Letters* 27, 1 (2017), 1740003:1–1740003:20. https://doi.org/10.1142/S0129626417400035

[53] Distributed System Group University of Ferrara. 2015. Sieve, Process and Forward (SPF). https://github.com/DSG-UniFE/spf

[54] Eliana Pastor, Elena Baralis, and Luca de Alfaro. 2023. A Hierarchical Approach to Anomalous Subgroup Discovery. In *39th IEEE International Conference on Data Engineering, ICDE 2023* (Anaheim, CA, USA). IEEE, 2647–2659. https://doi.org/10.1109/ICDE55515.2023.00203

[55] Eliana Pastor, Luca de Alfaro, and Elena Baralis. 2021. Looking for Trouble: Analyzing Classifier Behavior via Pattern Divergence. In *SIGMOD '21: International Conference on Management of Data, June 20-25* (Virtual Event, China). ACM, 1400–1412. https://doi.org/10.1145/3448016.3457284

[56] Eliana Pastor, Andrew Gavgavian, Elena Baralis, and Luca de Alfaro. 2021. How Divergent Is Your Data? *Proceedings of the VLDB Endowment* 14, 12 (2021), 2835–2838. https://doi.org/10.14778/3476311.3476357

[57] Simon J. Pennycook, Jason D. Sewall, and Victor W. Lee. 2019. Implications of a metric for performance portability. *Future Generation Computer Systems* 92 (2019), 947–958. https://doi.org/10.1016/j.future.2017.08.007

[58] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. 2008. Systematic Mapping Studies in Software Engineering. In *12th International Conference on Evaluation and Assessment in Software Engineering, EASE 2008* (Bari, Italy). BSC, 68—-77.

[59] Carlo Puliafito, Luca Conforti, Antonio Virdis, and Enzo Mingozzi. 2022. Server-side QUIC connection migration to support microservice deployment at the edge. *Pervasive and Mobile Computing* 83 (2022), 101580. https://doi.org/10.1016/j.pmcj.2022.101580

[60] Daniel A. Reed and Jack J. Dongarra. 2015. Exascale computing and Big Data. *Commun. ACM* 58, 7 (2015), 56–68. https://doi.org/10.1145/2699414

[61] Daniel A. Reed, Dennis Gannon, and Jack J. Dongarra. 2022. Reinventing High Performance Computing: Challenges and Opportunities. *CoRR* abs/2203.02544 (2022), 22 pages. https://doi.org/10.48550/arXiv.2203.02544 arXiv:2203.02544

[62] Sebastián Risco, Germán Moltó, Diana M. Naranjo, and Ignacio Blanquer. 2021. Serverless Workflows for Containerised Applications in the Cloud Continuum. *Journal of Grid Computing* 19, 3 (2021), 30. https://doi.org/10.1007/s10723-021-09570-2

[63] Christian Ritter, Benoit Côté, Falk Herwig, Julio F. Navarro, and Chris L. Fryer. 2018. SYGMA: Stellar Yields for Galactic Modeling Applications. *The Astrophysical Journal Supplement Series* 237, 2 (Aug. 2018), 42. https://doi.org/10.3847/1538-4365/aad691

[64] Lorenzo Rosa and Andrea Garbugli. 2022. Poster: INSANE - A Uniform Middleware API for Differentiated Quality using Heterogeneous Acceleration Techniques at the Network Edge. In *42nd IEEE International Conference on Distributed Computing Systems, ICDCS 2022* (Bologna, Italy). IEEE, 1282–1283. https://doi.org/10.1109/ICDCS54860.2022.00134

[65] Rohan Basu Roy, Tirthak Patel, Vijay Gadepally, and Devesh Tiwari. 2022. Mashup: making serverless computing useful for HPC workflows via hybrid execution. In *PPoPP '22: 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (Seoul, Republic of Korea). ACM, 46–60. https://doi.org/10.1145/3503221.3508407

[66] Gabriele Russo Russo, Tiziana Mannucci, Valeria Cardellini, and Francesco Lo Presti. 2023. Serverledge: Decentralized Function-as-a-Service for the Edge-Cloud Continuum. In *IEEE International Conference on Pervasive Computing and Communications, PerCom 2023* (Atlanta, GA, USA). IEEE, 131–140. https://doi.org/10.1109/PERCOM56429.2023.10099372

[67] Mandeep Kaur Saroa and Rajni Aron. 2022. An Energy-Efficient Load Balancing Approach for Scientific Workflows in Fog Computing. *Wireless Personal Communications* 125, 4 (2022), 3549–3573. https://doi.org/10.1007/s11277-022-09724-9

[68] Mahadev Satyanarayanan. 2017. The Emergence of Edge Computing. *Computer* 50, 1 (2017), 30–39. https://doi.org/10.1109/MC.2017.9

[69] Eva Sciacca, Ugo Becciani, Alessandro Costa, Fabio Vitello, Pietro Massimino, Marilena Bandieramonte, et al. 2015. An integrated visualization environment for the virtual observatory: Current status and future directions. *Astronomy and Computing* 11 (2015), 146–154. https://doi.org/10.1016/j.ascom.2015.01.006

[70] Helen Shen. 2014. Interactive notebooks: Sharing the code. *Nature* 515, 7525 (11 2014), 151—152. https://doi.org/10.1038/515151a

[71] Estela Suarez, Norbert Eicker, and Thomas Lippert. 2019. *Modular Supercomputing Architecture: From Idea to Production*. Vol. 3. CRC Press, FL, USA, Chapter 9, 223–251.

[72] Orazio Tomarchio, Domenico Calcaterra, Giuseppe Di Modica, and Pietro Mazzaglia. 2021. TORCH: a TOSCA-Based Orchestrator of Multi-Cloud Containerised Applications. *Journal of Grid Computing* 19, 1 (2021), 5. https://doi.org/10.1007/s10723-021-09549-z

[73] Matteo Turisini, Giorgio Amati, and Mirko Cestari. 2023. LEONARDO: A Pan-European Pre-Exascale Supercomputer for HPC and AI Applications. *CoRR* abs/2307.16885 (2023), 16 pages. https://doi.org/10.48550/arXiv.2307.16885 arXiv:2307.16885

[74] Andy B. Yoo, Morris A. Jette, and Mark Grondona. 2003. SLURM: Simple Linux Utility for Resource Management. In *Job Scheduling Strategies for Parallel Processing, 9th International Workshop, JSSPP 2003* (Seattle, WA, USA) *(LNCS, Vol. 2862)*. Springer, 44–60. https://doi.org/10.1007/10968987_3

[75] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauly, et al. 2012. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. In *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI* (San Jose, CA, USA). USENIX Association, 15–28.

[76] Florian Zaruba, Fabian Schuiki, Torsten Hoefler, and Luca Benini. 2020. MLIR abstraction for a RISC-V architecture. https://github.com/pulp-platform/snitch

[77] Florian Zaruba, Fabian Schuiki, Torsten Hoefler, and Luca Benini. 2021. Snitch: A Tiny Pseudo Dual-Issue Processor for Area and Energy Efficient Execution of Floating-Point Intensive Workloads. *IEEE Trans. Comput.* 70, 11 (2021), 1845–1860. https://doi.org/10.1109/TC.2020.3027900