Report from Dagstuhl Seminar 23352

Integrating HPC, AI, and Workflows for Scientific Data Analysis

Rosa M. Badia*[1], Laure Berti-Equille*[2], Rafael Ferreira da Silva*[3], and Ulf Leser*[4]

1 **Barcelona Supercomputing Center, ES.** rosa.m.badia@bsc.es
2 **Research and Development Institute, IRD Montpellier, FR.** laure.berti@ird.fr
3 **Oak Ridge National Laboratory, US.** silvarf@ornl.gov
4 **Humboldt University of Berlin, DE.** leser@informatik.hu-berlin.de

Abstract

The Dagstuhl Seminar 23352, titled "Integrating HPC, AI, and Workflows for Scientific Data Analysis," held from August 27 to September 1, 2023, was a significant event focusing on the synergy between High-Performance Computing (HPC), Artificial Intelligence (AI), and scientific workflow technologies. The seminar recognized that modern Big Data analysis in science rests on three pillars: workflow technologies for reproducibility and steering, AI and Machine Learning (ML) for versatile analysis, and HPC for handling large data sets. These elements, while crucial, have traditionally been researched separately, leading to gaps in their integration. The seminar aimed to bridge these gaps, acknowledging the challenges and opportunities at the intersection of these technologies. The event highlighted the complex interplay between HPC, workflows, and ML, noting how ML has increasingly been integrated into scientific workflows, thereby enhancing resource demands and bringing new requirements to HPC architectures, like support for GPUs and iterative computations. The seminar also addressed the challenges in adapting HPC for large-scale ML tasks, including in areas like deep learning, and the need for workflow systems to evolve to leverage ML in data analysis fully. Moreover, the seminar explored how ML could optimize scientific workflow systems and HPC operations, such as through improved scheduling and fault tolerance. A key focus was on identifying prestigious use cases of ML in HPC and understanding their unique, unmet requirements. The stochastic nature of ML and its impact on the reproducibility of data analysis on HPC systems was also a topic of discussion.

Seminar August 27 – September 1, 2023 – https://www.dagstuhl.de/23352

2012 ACM Subject Classification Computing methodologies → Distributed computing methodologies; Computing methodologies → Machine learning; Computing methodologies → Parallel computing methodologies

Keywords and phrases Large scale data presentation and analysis, Exascale class machine optimization, Performance data analysis and root cause detection, High dimensional data representation

Digital Object Identifier 10.4230/DagRep.13.8.129

* Editor / Organizer

Integrating HPC, AI, and Workflows for Scientific Data Analysis, *Dagstuhl Reports*, Vol. 13, Issue 8, pp. 129–164
Editors: Rosa M. Badia, Laure Berti-Equille, Rafael Ferreira da Silva, and Ulf Leser

Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Rosa M. Badia (Barcelona Supercomputing Center, ES)
Laure Berti-Equille (IRD – Montpellier, FR)
Rafael Ferreira da Silva (Oak Ridge National Laboratory, US)
Ulf Leser (HU Berlin, DE)

The Executive Summary for the Dagstuhl Seminar 23352 on "Integrating HPC AI and Workflows for Scientific Data Analysis" encapsulates a comprehensive discussion on the integration of High-Performance Computing (HPC), Artificial Intelligence (AI), and workflow technologies. The seminar, held from August 27 to September 1, 2023, was pivotal in highlighting the interdependence of these technologies for modern Big Data analysis. With a focus on bridging the gaps between these historically siloed areas, the seminar addressed the augmentation of resource demands due to the integration of AI into scientific workflows, the challenges posed to HPC architectures, and the exploration of AI's potential in optimizing workflow systems and operations, including scheduling and fault tolerance.

The seminar proffered a nuanced understanding of AI+HPC integrated workflows, elaborating on the different modes in which AI and HPC components could be coupled within workflows. These ranged from AI models replacing computationally intensive components (AI-in-HPC) to AI models that operate externally to steer HPC components or generate new data (AI-out-HPC), and to concurrent AI models that optimize HPC runtime systems (AI-about-HPC). Such integration is vital for the future of scientific workflows, where AI and HPC not only coexist but also co-evolve to foster more effective and intelligent scientific inquiry.

A shift in the paradigm of HPC systems towards real-time interaction within workflows was another focal point of the seminar. Moving away from the traditional batch-oriented systems, the seminar shed light on the emerging need for workflows that support dynamic, on-the-fly interactions. These interactions are not only vital for the real-time steering of computations and the runtime recalibration of parameters but also for making informed decisions on cost-value trade-offs, thereby optimizing both computational and financial resources.

The discussion also ventured into the realm of federated workflows, distinguishing them from the conventional grid computing model. Federated workflows, or cross-facility workflows, emphasize the orchestration of workflows across different computational facilities, each with distinct environments and policies. This paradigm advocates for a seamless execution of complex processes, underscoring the necessity of maintaining coherence and coordination throughout the workflow life cycle.

Contractual and quality-of-service (QoS) considerations in federated workflows, especially when crossing organizational boundaries, were identified as critical areas of focus. The seminar highlighted the need for formal contracts to manage the intricate bindings and dynamic interactions between various entities. The role of a federation engine was emphasized as a tool for translating requirements, ensuring compliance, and resolving disputes, thereby ensuring the workflow's needs are met at each federation point.

Moreover, the seminar identified key challenges and opportunities at the intersection of these technologies, such as the stochastic nature of ML and its impact on the reproducibility of data analysis on HPC systems. It highlighted the need for holistic co-design approaches, where workflows are introduced early and scaled from small-scale experiments to large-scale

executions. This approach is essential for integrating the "full" workflow environment, including ML/AI components, early in the process, thereby replacing expensive simulation with fast-running surrogates and enabling interactive exploration with the entire software environment.

In summary, the Dagstuhl Seminar 23352 provided an in-depth exploration of the synergistic relationship between HPC, AI, and scientific workflows. It paved the way for future research directions and practical implementations, aiming to revolutionize scientific data analysis by harmonizing computational power with intelligent, data-driven analysis. The discussions and outcomes of the seminar are poised to influence the development of workflow systems and technologies in the years to come, signaling a shift towards more integrated, adaptive, and efficient scientific computing paradigms.

2 Table of Contents

3 Challenges

3.1 Workflow Dynamics and Management

Rosa M. Badia (Barcelona Supercomputing Center, ES)
Silvina Caino-Lores (INRIA – Rennes, FR)
Kyle Chard (University of Chicago, US)
Wolfgang E. Nagel (TU Dresden, DE)
Fred Suter (Oak Ridge National Laboratory, US)
Domenico Talia (University of Calabria, IT)

A broad definition of an integrated AI+HPC workflow is that of a workflow in which at least one HPC task (i.e., an HPC simulation) and at least one AI task (e.g., a surrogate model) coexist in the same workflow application. We also consider that such workflows can run on HPC systems, although not necessarily all their components are executed on HPC systems.

We can then refine this definition according to how the AI and HPC components are effectively coupled, ordered, and placed within the workflow. The first coupling mode is when an AI model is used to replace a computationally intensive component, or the whole HPC simulation itself, of the workflow (AI-in-HPC). In this scenario, only the inference part of the model is part of the workflow, the training of the model being done offline. The second coupling mode captures scenarios where AI is used to steer the HPC components or generate new data or parameterization. The AI model thus resides "outside" of the main HPC simulation (AI-out-HPC). Here both training and inference can be part of the AI+HPC workflow, e.g., when using reinforcement learning techniques. The last coupling mode is when the AI models are concurrent and coupled to the main HPC tasks and run synergistically with simulations (AI-about-HPC). For instance, AI models can be used to optimize the performance of the HPC runtime system/workflow manager/resource manager/scheduler. Results of the HPC component are thus used to train the AI component as it runs and allow for system-wide predictions and optimization. Note that these three coupling modes are not mutually exclusive and are commonly combined within the same workflow.

3.1.1 Dynamic behavior in AI+HPC workflows

Most managers still treat static programming and execution. Expressing and handling dynamicity will become critical for AI+HPC workflows.

1. Adding components. Users might want to add or interact with components of the workflow. How to address this from a resource allocation and management perspective?
2. Alternative path in the workflow (branching). Workflows can have a traditional HPC component and a surrogate model. These options have different computational complexities and might be triggered by different conditions (e.g., low model accuracy). Dynamic resource allocation is needed in order to run the HPC simulation, for example.
3. Feedback loop (i.e., steering, control loops). Ability to manage detection of errors, thresholds, hooks in general, and change the behavior of the workflow accordingly. This can be due to the behavior of the AI model (bias, overfitting, etc.). For example, monitoring malicious (or not malicious, just because of data corruption) behavior in federated learning workflow, which are becoming increasingly present in HPC (e.g., Integrated circuit model building at the advanced photon source, biomedical workflows across Argonne Leadership Class Facility and Broad with APPFL).

4. Fault management, exception management. Ability to manage faults at the task level (software or hardware, division by error, etc) and change the behavior of the workflow accordingly.
5. Other things to consider: security, privacy, provenance, etc.

Although points 3 and 4 are conceptually different (the former is motivated by application functionality, the latter by application errors or runtime state), underlying mechanics in the runtime might be the same to manage these events. These mechanics might not necessarily be exposed to the same degree to the end user/application.

An additional problem can be how to detect these errors/faults. The workflow manager should provide this functionality.

3.1.2 Resource Management

One of the main activities performed by Workflow Management Systems is resource management, taking into account that the term resources include hardware infrastructure, software components and data.

We consider the following set of stages as the more common set for the workflow lifecycle under current workflow management systems:

- *Stage 1:* Preparation of environment: performs the software dependency deployment (i.e., using containers).
- *Stage 2:* Data and artifacts discovery/staging in. With artifacts we consider, for example, a pre-processed dataset, a trained AI model, etc. This can be for example performed using Apache airflow data pipelines that define the data movements that are needed before the actual computing execution.
- *Stage 3:* Compute resource provisioning. Current practices do not include elements of I/O scheduling and allocation that favor AI workflow executions. Other extensions that can be considered for improving convergence with AI are exploiting locality and reusing data blocks (for example reusing models for inference tasks), both on disk and in memory, but currently there is insufficient locality exposure to other elements in the system. Other ideas that can be leveraged in HPC come from the cloud, like expressing data affinity.
- *Stage 4:* Deployment of the pilot job(s). The dynamicity of the pilot shapes/types to match dynamic workload (e.g., MPI, single-core, GPUs, etc.).
- *Stage 5:* Execution of the actual computational workflows.
- *Stage 6:* Data and artifacts staging out.
- *Stage 7:* Cleanup.

The order of the stages might depend on the actual solution (e.g. using Conda environments, stage 1 is executed after stage 3), or some stages will not exist in some cases (i.e., stage 4 maybe is not needed in some cases).

We consider the following approaches are applied in workflows' resource management:

1. Use of AI to inform the scheduler: AI can be used in multiple forms to assist the resource management. We are listing a few below:
   - To estimate the required resources and run time and manage faults.
   - To learn from previous executions.
   - To perform uncertainty management (knowing that user-provided estimates are inaccurate, mainly for walltime, memory and, storage).

   The challenges arise on how to acquire, clean and assimilate the data necessary to train these models. This includes, Identifying available data sources: user knowledge, scheduler logs, application/workflow monitoring, hardware monitoring (e.g., bluefish, starfish, etc.).

2. Use of elasticity, heterogeneity and performance of AI in HPC systems. Optimally, workflow management systems should be able to request heterogeneous resources to suit the AI and HPC workload on demand (not just the amount of resources but the type of resource that will provide the desired level of performance for a specific task). Currently, some systems implement workarounds to provide elasticity within a static allocation (supported with Slurm and LSF, for example). Existing AI libraries may require reengineering to improve their performance and unlock high scalability on HPC environments (this might involve revisiting their interaction with the HW, I/O system, etc.). For example, some of this is happening e.g., in the DOE world for Intel and AMD GPUs. Also, vendors are working on this for specific hardware (e.g., cerebras, sambanova, graphcore, etc.).

3.1.3 Integration of Different Platforms and Systems

AI motivates the need for further integration of systems:

1. The integration of HPC, which typically operates on a batch-queue basis, with cloud technologies like Kubernetes (K8s), presents a unique challenge in workflow management. A work in progress approach is currently being explored, aiming to delegate parts of workflows to existing managers within each facility, such as Zambeze and Fluence, without necessitating oversight of the entire computation. This raises a pertinent question about the need to reevaluate and possibly redimension storage services in HPC facilities to effectively accommodate the increasing demands of AI workloads.
2. The management and locality of data in distributed or cross-facility workflows present significant challenges, particularly in terms of authentication, authorization, and policy adherence. Initiatives like EU's Gaia-X and Fenix, as well as the US's OneID, are making strides in addressing these issues. However, policy constraints often emerge as the most limiting factor in these contexts, impacting how data is accessed and shared across different facilities.
3. Also workflow management (see example of federated learning).
4. In the Edge-to-HPC paradigm, a key strategy is to position tasks, such as training, close to the data source at the edge, optimizing the use of edge nodes and HPC nodes. This involves adjusting the granularity of tasks and the size of data in relation to the computing capabilities of these nodes. A critical aspect of this approach is investigating the trade-offs between time-to-prediction and accuracy, especially when implementing compression and filtering techniques. These methods can impact the accuracy of AI models but are beneficial in reducing the data transfer and processing load. This approach is not only applicable to AI models but can also extend to other data generation stages, like HPC simulations, and is particularly relevant in scenarios requiring urgent computing and real-time steering, where efficiency and response time are crucial.
5. The integration of programming models and environments for workflows that combine HPC and AI presents a unique challenge, given the significant differences between HPC and AI programming environments. To bridge this gap, there are ongoing efforts such as PyCOMPSs [1], which is based on Python, aiming to harmonize these distinct environments. This initiative represents a step towards creating a more unified and efficient programming model that can cater to the diverse requirements of both HPC and AI, facilitating smoother integration and more effective workflow management in these complex computational domains.

In conclusion, while leveraging AI for scheduling in HPC systems promises efficiency gains, it also raises the critical question of its impact on energy consumption. The trade-off between enhanced scheduling performance and increased energy demands necessitates careful consideration. The path forward lies in steering AI towards green computing, where AI not only optimizes computational tasks for performance but also aligns with energy-efficient practices. This approach requires a delicate balance, ensuring that the benefits of AI in HPC do not inadvertently escalate energy consumption, but rather contribute to a more sustainable and environmentally conscious computing paradigm.

References

1 Natalia Poiata, Javier Conejero, R Badia, and Jean-Pierre Vilotte. Data-streaming workflow for seismic source location with pycompss parallel computational framework. *EGU General Assembly*, 10, 2021.

3.2 Sustainability Concerns

Laure Berti-Equille (IRD – Montpellier, FR)
Timo Kehrer (Universität Bern, CH)
Christine Kirkpatrick (San Diego Supercomputer Center, US)
Dejan Milojicic (HP Labs, US)
Sean R. Wilkinson (Oak Ridge National Laboratory, US)

In this section, we delve into the crucial questions raised during our brainstorming sessions focusing on the integration of sustainability in HPC and AI-driven scientific workflows. These discussions centered around three main queries: (1) how to cultivate sustainability awareness, (2) the categorization of key sustainability challenges specific to HPC+AI workflows, and (3) pinpointing where these challenges predominantly arise in the workflow lifecycle. To tackle the first question on building sustainability awareness, we drew insights from existing initiatives such as the Sustainability Awareness Framework (SusAF) [1], adopting a multi-faceted approach that encompasses environmental, economic, social, and technical dimensions of sustainability [2]. Addressing the second question, Fig. 1 provides a structured categorization of these sustainability challenges, recognizing that the list is not exhaustive and acknowledging the existence of cross-cutting challenges that span multiple dimensions.

For example, among cross-cutting challenges, **sustainability awareness** is a central challenge as people (social dimension) don't always understand how to measure the energy they're consuming (this is especially difficult on a shared system) (technical dimension) and also the carbon footprint of their workflows (environmental dimension) [3]. This is also difficult to measure if the amounts (energy, CO2) are not even exposed somehow.

The **re-usability** challenge also belongs to both technical and social as for the latter, how to convince potential users to re-use (parts of) existing workflows and actually make this re-use technically possible and easy for various kinds of users (DevOps, domain scientist, data scientist, etc.). Another example of central challenges, at the governance level and at the convergence of social, technical, economic and environmental dimensions of sustainability is the **deployment of sustainability-rated multi-objective policies** where some priorities can be defined adaptively depending of the application context and needs to favor either environmental, social or economic considerations for example. As it's commonly the case

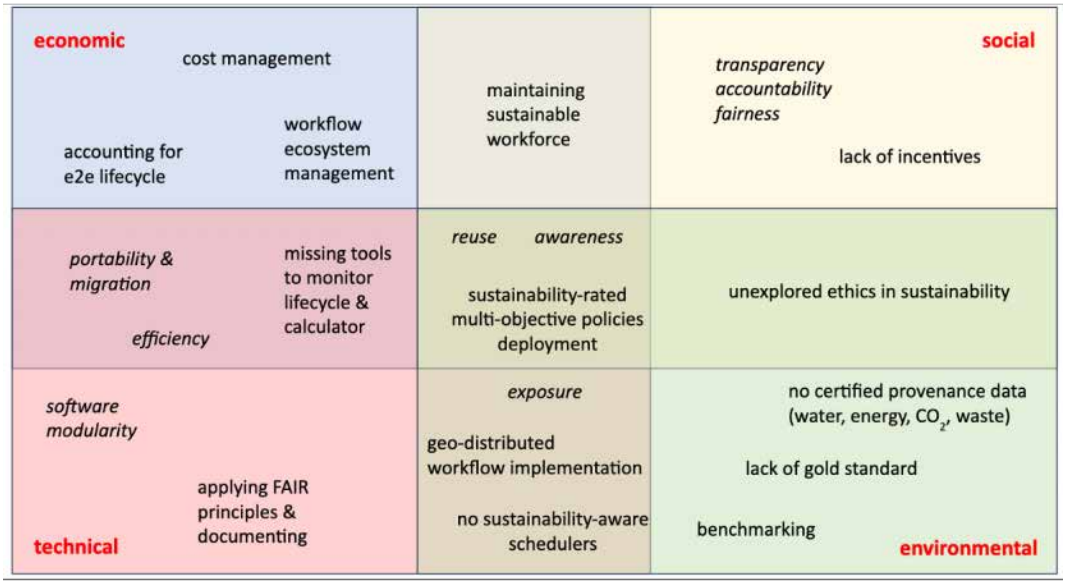

Figure 1 Categorized sustainability challenges in HPC+AI workflows.

in many sustainability studies since the UN 2030 agenda and its SDGs [4], we can also distinguish between high-level objectives and activities which need to be improved to reach the objectives since both are challenging per se. In the figure, we note in italic the objectives whereas the activities are in normal font.

3.2.1 Main Sustainability Challenges

- **Environmental Sustainability.** The main challenge we identified under the environmental sustainability dimension is how to get **certified provenance data** to quantify or estimate the impact on the environment and natural resources, i.e., provenance data about CO2 emissions, energy or water consumption from authoritative sources [5]. The lack of traceability metadata makes it very difficult to know, for example, where the electricity comes from and distinguish between renewable vs non-renewable energy, clean or green energy powering some given HPC+AI workflows [6]. Similarly, reliable and continuously up-to-dated data about water-usage, gas emission or waste management is currently missing, not detailed enough, or not trustworthy and we cannot drill-down and quantify the impact of a single workflow on the environment (or drill-up for a set of workflows of a given application) [7]. As a consequence, there is a **lack of benchmark** and a **lack of gold standard** that could be used as a reference for virtuous practice. More generally, this leads to a lack of **exposure** and **awareness** regarding environmental impacts of the workflow lifecycle.
- **Social Sustainability.** A central challenge is the **lack of incentives** individual researchers, service providers, organizations, and funders have for nurturing sustainability. This is especially true for researchers, who are incentivized to experiment, analyze, and publish results. Nowhere in the academic process that relates to tenure and promotion are sustainability concerns accounted for, such as making computing choices that are economically and environmentally sustainable. Similarly, funders have little incentive to promote sustainability as they do not directly suffer the consequences of poor economic or environmental choices. Another challenge in the social realm is making choices that

maintain a sustainable workforce. For example, a researcher has no incentive to use the workflow package or machine learning (ML) software with the largest market share or that is most widely used and supported at a computing center. If more researchers stayed with market share and understood platforms, it is easier on research computing staff to support fewer packages, as well as software that is maintained with security patches and bug fixes. The incentives for such dimensions are with service providers, who have few ways to motivate researchers to factor other variables into their choice of research computing components. One way to promote social sustainability values, including **transparency, accountability, and fairness**, is to embed these principles in shared community values. The US National Science Foundation (NSF) funded EarthCube initiative brought together geoscientists and cyberinfrastructure builders to build innovative tools and invigorate the community with advanced computing techniques. The EarthCube community, led by leaders elected by the membership and a funded coordination office, wished to imbue the context of the collective work with shared principles. The principles included Responsibility, Dependability, Service, Openness-Transparency (among others). By defining the shared values, EarthCube was able to tune strategic actions to model and encourage these seemingly intrinsic qualities. This led to impactful work as shown in the recently released EarthCube retrospective [8].

- The **Technical sustainability** of HPC+AI workflows can be considered in (at least) three ways: (1) that of the HPC system itself, (2) that of the AI models used as tools and constructed by the workflow, and (3) that of the workflow itself when considered as research software. Because current HPC systems are short-lived and highly specialized, HPC workflows are usually tailored specifically to the systems where they will run, and this causes problems in workflow portability (running a workflow on another current HPC system) and workflow migration (running a workflow on a future HPC system).
- The **Economic Sustainability** relates to cost management of HPC+AIworkflows, as well as the entire workflow ecosystem management, accounting for the end-to-end lifecycle of the workflows. Some of the above challenges can be dependent (correlated or anti-correlated). Therefore, corrective actions may have various indirect positive or negative impacts beyond what we can expect and there is not yet a principled way of estimating and predicting the collateral effects of improving one dimension or one objective over the others. For example, if we decrease the energy consumption of a workflow the overall cost can however be increasing; loose-coupling of softwares can have a positive effect and facilitate reuse, but may also increase the computation cost [9]. Nevertheless, we should not compromise in making research progress.

3.2.2 Sustainability Challenges Across the Workflow Lifecycle

Finally, we attempt to address the third question "Can we pinpoint where the sustainability challenges are predominant in the workflow lifecycle?" We decompose the workflow lifecycle into several stages and consider the human, data, and AI planes (Fig. 2) [10].

Although it is not represented as a cycle, the workflow lifecycle is iterative and can have multiple feedback loops. The data processing block can be decomposed into multiple blocks (data acquisition/collection, storage, preprocessing, analytics, use, distribution, archival), and again not necessarily executed in a sequential manner and some blocks (such as archival) may not be present in some workflows. Similarly, the ML model development block can be decomposed into various blocks such as feature engineering, model building, training, tuning, validation and exec monitoring. Data and AI planes are predominantly supported by HPC and AI softwares respectively. Our exercise consisted in mapping and locating the challenges

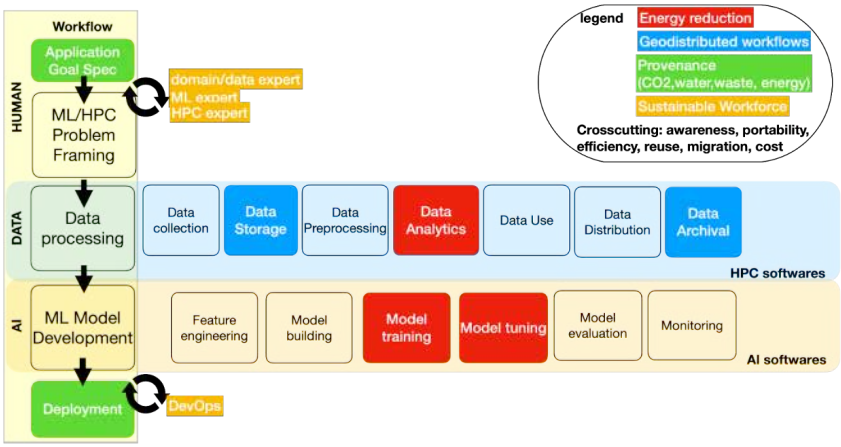

Figure 2 Quasi reference model for sustainability in HPC+AI workflows.

we identified previously into the generic workflow lifecycle because all the challenges may not be not occurring at every stage and when they occur they may not be predominant or have a critical impact on sustainability. Empirical experiments will surely be needed to verify our assumptions here.

For example, energy consumption could be reduced essentially targeting the data analytics stage with more frugal methods, in particular during model training and tuning stages. Ensuring that the workforce (such as domain and data scientists as well as Devops engineers) is sustainable is important during the early stage of the application domain and goal specifications for designing the adequate workflow and framing the ML on HPC problem and also during the deployment as well as continuously gathering reliable data characterizing the environmental impact of the workflow (energy, gas emission, etc.). Reallocating computing resources or data storage (vicinity) can save energy and reduce technical or environmental costs related to computation, data storage and archival.

A follow-up of this preliminary discussion can be to identify the leverages in the workflow lifecycle where a small improvement of some targeted tasks can have a huge positive impact on the sustainability dimensions. Next, we could define Whatif scenarios and experiments to simulate the gain/loss in terms of sustainability and support sustainability-rated multi-objective policies.

3.2.3 Use Cases

The following are some use cases that drive sustainability concerns:

- *Physical limits:* some sites cannot bring more power than they are designed for, therefore it is required to do power-throttling and minimizing power consumption to conduct computation. Both this and the next use case mean doing more (computation) with less (energy)
- Economically-driven: energy costs a lot of money, running an exascale computer costs even more than buying it. Therefore it is required to be mindful when leveraging these computers to get most out of large scale computations.
- Save the planet: many executives are making pledges to make their corporations net-zero or even net-positives. It is non-trivial to achieve this and in many cases it means payments towards clean energy. It entails both upstream and downstream explorations.
- Broader good: there is a public pressure for less consumption and for clean energy use. HPC and AI computers consume lots of energy, so being mindful to use clean energy most of the time is one way to alleviate the problem.

References

1. Sustainability Awareness Framework (SusAF). https://www.suso.academy/en/sustainability-awareness-framework-susaf/, 2023.
2. Cullen Bash, Kirk Bresniker, Paolo Faraboschi, Tiffani Jarnigan, Dejan Milojicic, and Pam Wood. Ethics in sustainability. *IEEE Design & Test*, 2023.
3. Baolin Li, Siddharth Samsi, Vijay Gadepally, and Devesh Tiwari. Sustainable hpc: Modeling, characterization, and implications of carbon footprint in modern hpc systems. *arXiv preprint arXiv:2306.13177*, 2023.
4. Transforming our world: the 2030 Agenda for Sustainable Development. https://sdgs.un.org/2030agenda, 2023.
5. Sadasivan Shankar and Albert Reuther. Trends in energy estimates for computing in ai/machine learning accelerators, supercomputers, and compute-intensive applications. In *2022 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–8. IEEE, 2022.
6. Dan Zhao, Nathan C Frey, Joseph McDonald, Matthew Hubbell, David Bestor, Michael Jones, Andrew Prout, Vijay Gadepally, and Siddharth Samsi. A green (er) world for ai. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 742–750. IEEE, 2022.
7. Sirui Qi, Dejan Milojicic, Cullen Bash, and Sudeep Pasricha. Shield: Sustainable hybrid evolutionary learning framework for carbon, wastewater, and energy-aware data center management. *arXiv preprint arXiv:2308.13086*, 2023.
8. EarthCube: A Retrospective 2011-2022. https://library.ucsd.edu/dc/object/bb0180794q, 2023.
9. Amir Nassereldine, Safaa Diab, Mohammed Baydoun, Kenneth Leach, Maxim Alt, Dejan Milojicic, and Izzat El Hajj. Predicting the performance-cost trade-off of applications across multiple systems. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 216–228. IEEE, 2023.
10. Nicolas Dube, Duncan Roweth, Paolo Faraboschi, and Dejan Milojicic. Future of hpc: The internet of workflows. *IEEE Internet Computing*, 25(5):26–34, 2021.

3.3 Integration and Standardization

Ilkay Altintas (San Diego Supercomputer Center, US)
Rosa Filgueira (University of St Andrews, GB)
Ana Gainaru (Oak Ridge National Laboratory, US)
Shantenu Jha (Rutgers University, US & Brookhaven National Laboratory, US)
Ulf Leser (HU Berlin, DE)
Bertram Ludäscher (University of Illinois at Urbana-Champaign, US)
Jeyan Thiyagalingam (Rutherford Appleton Lab., GB)

3.3.1 Reference Architectures From a Workflow Perspective

To date, no established reference architecture for scientific workflow systems exists. Furthermore, there neither exists a characterization of the precise functionalities a workflow system should encompass nor where it should interface (and with which API) with other components of a distributed infrastructure. The general components of a distributed infrastructure steered by a workflow system are depicted in the idealized architecture shown in Fig. 3. From top to bottom, these include:

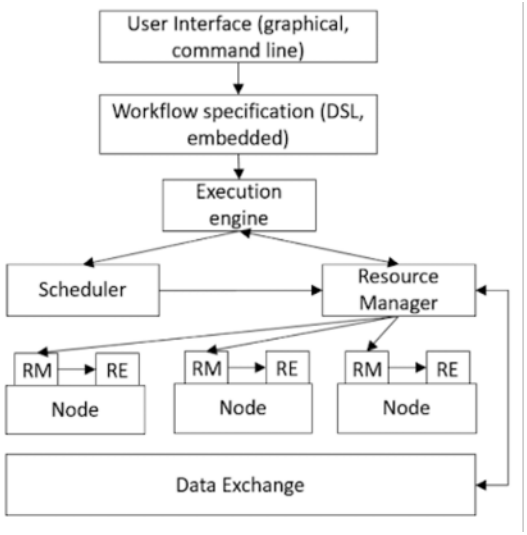

Figure 3 Main components of a typical distributed infrastructure steered by a workflow system.

- **User interface.** While many systems target developers and favor command line interfaces [1], others offer comprehensive graphical interfaces using the DAG-structure of workflows as metaphor [2]. Graphical interfaces often are also associated with access to libraries of available workflow tasks or control structures of a workflow language [3].
- **Workflow specification.** Many systems use specific domain specific languages for specifying a workflow, which might resemble programming languages [4] or come in the form of flat file formats [5]. Other systems offer workflow functionality as extensions to a host programming language without having a proper syntax [TBA+17]. Workflow specifications describe an abstract workflow; during execution, tasks defined abstract workflow often lead to multiple physical instances.

- **Workflow engine.** The workflow specification (or workflow program) is executed by a workflow engine, whose main purpose is the control of the dependencies between workflow tasks. As such, the workflow engine, at every stage of a workflow execution, must be able to determine the set of currently executable tasks, which requires some form of bidirectional communication with the scheduler to be informed about finished tasks. At this stage, typically also the compilation of the abstract workflow into a physical one is performed. In dynamic workflow systems, where the set and structure of tasks is data dependent, the communication must include further aspects, such as number of files in a directory (for scatter operations) or intermediate data files themselves (for conditionals) [6].
- **Scheduler.** The scheduler is informed by the workflow engine about the set of ready -to-run tasks and determines their assignment to the set of available (virtual) compute nodes. To this end, it communicates with a resource manager to obtain free resources and to access the characteristics of free nodes (e.g., main memory, accelerators etc.). Schedulers typically are not individual components, but their functionality instead is included in that of another system (see below).
- **Resource Managers and Runtime Environment.** Resource managers have two different purposes (RM). At the global level, the RM oversees the status of all registered nodes together with their functional characteristics and offers controlled access to them for its clients, possibly with an associated QoS. At the local level, each node runs an instance of the RM to control the node itself and to provide a local runtime environment (RE) for workflow tasks (such as Docker or Singularity containers).
- **Data exchange.** Finally, the tasks of a workflow execution must exchange data along their data dependencies. There are different ways how this can be achieved. In a streaming setting, all data exchange is handled through the network [13]. Batch-processing often assumes availability of a shared data space, for instance provided by a parallel file system like Lustre [21] or CePH [7].

However, concrete systems often deviate heavily from this architecture. For instance, scheduling usually is not implemented as a separate component, but instead is performed inside the workflow engine or by the resource manager – and sometimes by both [18]. Systems with graphical user interfaces might not have an explicit language or format to express a workflow specification but instead directly interpret the workflow graph. Workflow engines are typically tightly coupled to a workflow specification language and not capable of executing any other specifications; moving from one system to another therefore requires program translation. Systems trying to co-optimize task placement and data locality need more expressive interfaces to resource managers, schedulers and file systems and explicit ways of manipulating placement [14]. Furthermore, there is no agreement on the interfaces between components, such as between workflow engines, schedulers, and resource managers or between a graphical user interface and a workflow engine. Even at interfaces where standards exists, such as POSIX for file access or DRMAA for resource managers, these are not implemented by all systems. For instance, HFDS is not Posix compliant, and Kubernetes does not support DRMAA.

3.3.2 Role of AI in workflows and HPC

The role of AI in scientific workflows and HPC is increasingly prominent and multifaceted. Here is a summary of the role of AI in these both of domains:

1. Enhancing Scientific Workflows:
   - *Workflow Optimization:* AI techniques, such as reinforcement learning [19, 40], can optimize the execution of complex scientific workflows.
   - *Discovering Similar Workflows:* Semantic code search powered by AI can assist researchers in finding workflows that are functionally or structurally similar to their own. Semantic techniques [31, 30] can be especially beneficial when looking for existing solutions to similar scientific problems, thus promoting knowledge sharing and collaboration.
   - *Identifying Workflow Components:* Researchers can use AI and LLMs models to perform semantic code searches [38] and identify specific components or tasks within workflows that perform certain tasks.. This enables reusing and adapting existing components to build new workflows more efficiently.
   - *Automated Documentation:* AI and LLMs can generate automated documentation for scientific workflows [38]. This documentation enhances the understanding of the workflow, and also fosters reproducibility, making it easier for researchers to share and replicate their work.
2. Data-Driven HPC-workflows:
   - *Data Analysis:* AI and machine learning enable sophisticated data analysis within scientific workflows, extracting valuable insights from large datasets [16, 26].
   - *Pattern Recognition:* AI [34] can identify patterns and correlations in scientific data, aiding researchers in discovering hidden relationships and making data-driven decisions.
   - *Real-time Insights:* In HPC simulations [24, 22], AI [27] can provide real-time insights, facilitating adaptive simulations based on changing conditions. This is especially relevant in fields like seismology, climatology, etc, where quick responses are critical.
3. HPC and AI Synergy:
   - *Resource Allocation:* AI optimizes resource utilization in HPC clusters by dynamically [8] allocating computing resources based on workload demands or previous runs [25]. This leads to cost savings and improved efficiency in resource usage.
   - *Reducing Energy Consumption for Exascale:* One of the critical challenges in exascale computing is the immense energy consumption of supercomputers. AI contributes to energy-efficient [29] computing by optimizing cooling systems, and power usage. Machine learning models can predict workload patterns and dynamically adjust power consumption to match computing demands, significantly reducing energy waste.
   - *Enhancing Fault Tolerance:* HPC systems are prone to hardware failures and errors due to their sheer complexity. AI-driven [23] fault tolerance mechanisms can detect anomalies in real-time and initiate corrective actions. Machine learning models can predict hardware failures before they occur, allowing for proactive maintenance and minimizing downtime.
   - *Predictive Analytics:* AI models can diagnose [32] and predict IO bottlenecks [37], data transfer issues, or compute node failures [DA+18] in advance, allowing for proactive mitigation and improved workflow efficiency.
   - *Deep Learning:* HPC infrastructures are crucial for training deep learning models [11, 20] on vast datasets. The synergy allows scientists to use AI for tasks like image recognition, natural language processing, and simulation optimization.

In summary, AI plays a vital role in scientific workflows and HPC by optimizing processes, extracting insights from data, and fostering synergy between these domains, ultimately advancing research and innovation in numerous fields.

3.3.3 AI within the Reference Architectures: Issues and Challenges

AI can play an important role for many components of a workflow system running on HPC. However, integrating such functionality also faces a number of challenges for which no good solutions exist today. These are:

- **Scheduling** in many systems is not an operation performed once at a single component, but instead often is accomplished in an iterative, hierarchical fashion. For instance, in a dynamic workflow at every point in time when a scheduler needs to take a decision, only the immediate next steps are known, while further downstream tasks are not. Decisions considering long-range implications, which are typical for AI-based schedulers, are thus impossible [15]. Another example for PilotJobs, which are scheduled by the resource managers as single tasks, but which during execution actually are expanded into multiple tasks which then need to be scheduled by the PilotJob itself [33]. Such two level scheduling currently is neither supported by AI-based solutions nor adequately reflected in the reference architectures.
- **Resource predictions.** Many advanced solutions for scheduling and resource management rely on precise predictions for the resource the execution of a task will require, such as runtime, memory, bandwidth, or energy. While many methods for performing such predictions are currently developed [35, 12], from an architectural point of view it is not clear where this functionality should be placed. The methods often assume access to current or past log files, which requires a positioning deep in the stack; on the other hand, their results are required by the workflow engine, the scheduler, and the resource manager. In pure online prediction systems, which try to predict resource requirements only based on the currently run workflow, there exists a dependency between predictions and scheduling, as the scheduler must take into account for which tasks and at which accuracy predictions are possible, for instance to prefer tasks for which this is not possible yet [36]. A further dependency that is missing in the reference architectures exists between resource predictions and resource managers, as a prediction regarding, for instance, main memory, are an important input for right-sizing of containers and virtual machines.
- **Metrics.** Our discussion so far focused much on conventional features of workflow/-workload execution, such as runtime and resource requirements. However, AI-based workflows also bring entirely new metrics that must be taken into account. Two particularly important ones are accuracy and transparency. Accuracy describes the quality of the result produced by an AI-based workflow. From a user perspective, optimizing accuracy actually might be more important than optimizing runtime, but requires entirely different means of user support [17]. Transparency describes the property of a workflow answer to be explainable from a user perspective, and is an important cornerstone of trust in data analysis results. Again, optimizing for transparency calls for different actions than optimizing resource demands.
- **Cloud-based architectures.** Our reference architecture does not consider the typical properties of cloud infrastructures. For instance, elasticity, i.e., a growing and shrinking of the available compute nodes during workflow execution, is not considered, but could be ideally combined with AI based workload predictions. It would require that tasks during their execution can acquire more computational resources, which breaks the hierarchical nature of the architecture sketch in Fig. 3. In cloud environments, function-as-a-service has become popular recently (also in workflow systems [28]) as a means to provide serverless and thus easier to maintain workflow executions. Such approaches require an adaptation of the reference architectures, as not anymore discrete tasks are the basic unit of operations, but instead asynchronous function calls, which requires a different understanding of resource management and scheduling.

- **Changing user groups.** The recent "democratization of data science[1]" results in a drastic change in the types of users that workflow and HPC systems must support. IN a nutshell, the user base grows enormously, while at the same time the typical technical capabilities and resources associated to a user or user project shrinks. This calls for new and easier to use interfaces (e.g., graphical metaphors, integration with research data management, improved result visualization, interactive and human-in-the-loop interfaces etc.) and expanded user support (e.g., other means of workflow design and adaptations, workflow testing, and workflow debugging; personalized assistance; improved and context-dependent documentation etc.). AI can play an important role here to make interfaces personalized and more context-dependent.

3.3.4 The Role of Benchmarking for Workflows, AI and HPC

Given the broader role of AI, HPC and workflows on data analysis, it is almost difficult to ascertain the suitability of a workflow engine, or workflow, or AI or HPC technique(s) for a given data analysis task. This becomes further complicated if performance (either runtime or scientific task performance) becomes a qualifying metric for the final decision around a particular AI technique or workflow or choice of a workflow engine. This complexity entails a need for a mechanism that can aid scientists (or stakeholders) in making such a decision. Benchmarking has been the cornerstone of solving such issues since the inception of software systems. As such, it is conceivable that a mechanism akin to benchmarking would be ideal to understand the interplay between these aspects.

Although one can resort to simply benchmark a workflow (or any aspect of interest), in the absence of a well-defined or well-established basis, such efforts would become meaningless. One option is to establish a common set of benchmarks that would capture a number of realistic mix of cases of different data analysis problems. A set of applications (whether realistic or synthetic) make up a benchmark suite, and are very specific to the case in hand, and in our case, workflows.

The notion of benchmarking for scientific workflows is not a novel concept, and in fact, can be widely found in the literature [9]. However, a benchmark suite that captures the integration of AI and HPC for data analysis workflows, especially in the context of recent developments in AI (such as LLMs), adds additional complexities. This is further complicated with the recent developments around detector rates, capability of modern facilities (such as AI at the edge), and modern AI-specific architectures. However, given the diverse range of scientific applications, it would be a monumental effort if the suite were to provide a full coverage of all application cases. Instead, one can envisage building a benchmark suite based on application classes or beamline types.

Identifying such a class or themes of benchmarks is not only useful for quantifying the performance capabilities of different workflow solutions, but also very instrumental in for understanding and assessing the functional capabilities of different workflow engines.

3.3.5 Benchmarking Techniques

In this section, we delve into the intricacies of benchmarking within Research Agendas (RAs), particularly focusing on workflow engines and their deployment across various infrastructures. The benchmarking process in this context is multifaceted, encompassing every component of a RA. This includes conducting integration tests, which are crucial for ensuring that different components of a workflow engine function cohesively.

[1] https://hbr.org/2018/07/the-democratization-of-data-science

One effective approach is to run multiple workflows within the same workflow engine on different infrastructures. This method is not only beneficial for testing the robustness and scalability of the system but also helps in identifying potential bottlenecks and optimization points. Such tests are exemplified by initiatives like nf-core [10], which demonstrate the practical application of these benchmarking strategies. However, the challenge arises when attempting to implement the same problem across multiple workflow engines, each in their respective language, and on different or similar infrastructures. This approach is invaluable for comparative analysis, providing insights into the relative strengths and weaknesses of various systems.

Despite its benefits, this methodology is not without its drawbacks. The primary challenge lies in the significant effort and resources required to implement and manage these benchmarks. Setting up the same problem across different workflow engines involves considerable time and expertise, particularly in adapting the problem to the nuances of each engine's language and infrastructure compatibility. Moreover, the need to run these benchmarks on various infrastructures adds another layer of complexity, requiring careful planning and coordination to ensure accurate and meaningful comparisons. Thus, while these benchmarking techniques offer substantial benefits in evaluating and improving research agendas, they also demand a considerable investment in terms of effort and resources.

3.3.6 Benchmarking Targets

In this section, we explore the diverse aspects and environments where workflow benchmarking is applied, emphasizing the need for a comprehensive approach to adequately assess and improve these systems. The benchmarking targets are multifaceted, ranging from the nature of the workflows (stream, batch, task-based) to the underlying data handling mechanisms (main memory versus file-based). Each of these aspects presents unique challenges and opportunities for optimization, making them critical targets for benchmarking efforts.

One of the key considerations in workflow benchmarking is the approach to data processing, with distinctions between synchronous and asynchronous workflows. Synchronous workflows, where tasks are executed in a predetermined order, and asynchronous workflows, which allow tasks to run independently and often concurrently, each have their own performance characteristics and optimization needs. The benchmarking process also needs to account for the diversity in programming languages used in workflow systems. Multi-language support is essential to cater to the varied requirements and preferences of different user groups. Additionally, the infrastructure on which these workflows are executed plays a crucial role. This includes the type of resource manager, the availability and use of GPUs, and the size and configuration of clusters. Benchmarking must therefore encompass a wide range of infrastructure setups to ensure comprehensive evaluation and optimization.

Another important target for benchmarking is the expressiveness of the languages used to define workflows. This includes evaluating how well a language or system supports complex constructs like conditionals and recursion. The ability of a workflow system to handle these elements effectively can significantly impact its usability and efficiency, making it a crucial aspect of benchmarking. By focusing on these varied targets, workflow benchmarking can provide critical insights into the performance and capabilities of these systems, guiding improvements and ensuring they meet the diverse needs of their users.

3.3.7 Issues with Benchmarking

In this section, we address the complexities and challenges that arise in accurately evaluating workflow performance. A primary concern is identifying and scrutinizing the performance-critical parts of workflow execution. This includes assessing the efficiency of various components such as the scheduler, the methods used for dependency resolution, and the workflow interpreter, as well as the quality of the individual task implementations. Each of these elements can significantly impact the overall performance of a workflow, making their thorough assessment essential for a comprehensive understanding of the system's capabilities.

Another pivotal issue in workflow benchmarking is the choice between real measurements and simulations. Real measurements offer tangible data on system performance under actual conditions, but they may have limitations in terms of scalability and broader applicability. In contrast, simulations, like those executed using WorkflowSim on platforms such as CloudSim and NetSim, are invaluable for scalability tests and can provide insights that are not feasible in real-world settings. However, simulations might not capture all the intricacies of real-world performance. The challenge is to balance the insights gained from simulations with the practicalities of real-world measurements, especially considering the risk of overfitting in benchmarks. This issue is evident in benchmarks like Linpack for high-performance computing systems, where optimization often focuses more on the benchmark than on practical applications.

Furthermore, the diversity of applications complicates the process of workflow benchmarking. Workflows vary greatly, from being heavily reliant on AI, to focusing on streaming data, to being based on complex simulations, or even being hybrids of these types. Each category requires specific approaches to benchmarking to accurately assess performance and efficiency. Therefore, there's a growing need for a benchmarking standard that can adapt to these diverse requirements, providing a balance between broad applicability and specific, actionable insights. This standard differs from internal benchmarks used for tuning or procurement, which often focus on optimizing specific aspects of a system's performance. Developing effective benchmarking strategies that can navigate these issues is critical for guiding the optimization and development of workflow systems in a way that is both efficient and applicable to a wide range of real-world scenarios.

References

1 Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B Hall, Christopher H Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O Twardziok, Alexander Kanitz, et al. Sustainable data analysis with snakemake. *F1000Research*, 10, 2021.

2 Jeremy Goecks, Anton Nekrutenko, James Taylor, and Galaxy Team team@ galaxyproject. org. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, 11:1–13, 2010.

3 Paolo Missier, Stian Soiland-Reyes, Stuart Owen, Wei Tan, Alexandra Nenadic, Ian Dunlop, Alan Williams, Tom Oinn, and Carole Goble. Taverna, reloaded. In *Scientific and Statistical Database Management: 22nd International Conference, SSDBM 2010, Heidelberg, Germany, June 30–July 2, 2010. Proceedings 22*, pages 471–481. Springer, 2010.

4 Paolo Di Tommaso, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. Nextflow enables reproducible computational workflows. *Nature biotechnology*, 35(4):316–319, 2017.

5 Peter Couvares, Tevfik Kosar, Alain Roy, Jeff Weber, and Kent Wenger. Workflow management in condor. *Workflows for e-Science: Scientific Workflows for Grids*, pages 357–375, 2007.

6 Marc Bux, Jörgen Brandt, Carl Witt, Jim Dowling, and Ulf Leser. Hi-way: Execution of scientific workflows on hadoop yarn. In *20th International Conference on Extending Database Technology, EDBT 2017, 21 March 2017 through 24 March 2017*, pages 668–679. OpenProceedings. org, 2017.

7 Ceph. `https://docs.ceph.com/en/quincy/`, 2023.

8 Wei-Che Chien, Chin-Feng Lai, and Han-Chieh Chao. Dynamic resource prediction and allocation in c-ran with edge artificial intelligence. *IEEE Transactions on Industrial Informatics*, 15(7):4306–4314, 2019.

9 Taina Coleman, Henri Casanova, Ketan Maheshwari, Loïc Pottier, Sean R Wilkinson, Justin Wozniak, Frédéric Suter, Mallikarjun Shankar, and Rafael Ferreira da Silva. Wfbench: Automated generation of scientific workflow benchmarks. In *2022 IEEE/ACM International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, pages 100–111. IEEE, 2022.

10 Philip A Ewels, Alexander Peltzer, Sven Fillinger, Harshil Patel, Johannes Alneberg, Andreas Wilm, Maxime Ulysse Garcia, Paolo Di Tommaso, and Sven Nahnsen. The nf-core framework for community-curated bioinformatics pipelines. *Nature biotechnology*, 38(3):276–278, 2020.

11 Diogo R Ferreira and JET Contributors. Using hpc infrastructures for deep learning applications in fusion research. *Plasma Physics and Controlled Fusion*, 63(8):084006, 2021.

12 Rafael Ferreira da Silva, Gideon Juve, Mats Rynge, Ewa Deelman, and Miron Livny. Online task resource consumption prediction for scientific workflows. *Parallel Processing Letters*, 25(03):1541003, 2015.

13 Rosa Filguiera, Amrey Krause, Malcolm Atkinson, Iraklis Klampanos, and Alexander Moreno. dispel4py: A python framework for data-intensive scientific computing. *The International Journal of High Performance Computing Applications*, 31(4):316–334, 2017.

14 Salvatore Giampà, Loris Belcastro, Fabrizio Marozzo, Domenico Talia, and Paolo Trunfio. A data-aware scheduling strategy for executing large-scale distributed workflows. *IEEE Access*, 9:47354–47364, 2021.

15 Naveen Kumar Gondhi and Ayushi Gupta. Survey on machine learning based scheduling in cloud computing. In *Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*, pages 57–61, 2017.

16 Shantenu Jha, Vincent R Pascuzzi, and Matteo Turilli. Ai-coupled hpc workflows. *arXiv preprint arXiv:2208.11745*, 2022.

17 Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access*, 2023.

18 Fabian Lehmann, Jonathan Bader, Friedrich Tschirpke, Lauritz Thamsen, and Ulf Leser. How workflow engines should talk to resource managers: A proposal for a common workflow scheduling interface. *arXiv preprint arXiv:2302.07652*, 2023.

19 Chin Poh Leong, Chee Sun Liew, Chee Seng Chan, and Muhammad Habib Ur Rehman. Optimizing workflow task clustering using reinforcement learning. *IEEE Access*, 9:110614–110626, 2021.

20 Xiaoyi Lu, Haiyang Shi, Rajarshi Biswas, M Haseeb Javed, and Dhabaleswar K Panda. Dlobd: A comprehensive study of deep learning over big data stacks on hpc clusters. *IEEE Transactions on Multi-Scale Computing Systems*, 4(4):635–648, 2018.

21 Lustre. `https://www.lustre.org/`, 2023.

22 Federica Magnoni, Emanuele Casarotti, Pietro Artale Harris, Mike Lindner, Andreas Rietbrock, Iraklis Angelos Klampanos, Athanasios Davvetas, Alessandro Spinuso, Rosa Filgueira, Amy Krause, et al. Dare to perform seismological workflows. In *AGU Fall Meeting Abstracts*, volume 2019, pages IN13C–0726, 2019.

23 Avinab Marahatta, Qin Xin, Ce Chi, Fa Zhang, and Zhiyong Liu. Pefs: Ai-driven prediction based energy-aware fault-tolerant scheduling scheme for cloud data center. *IEEE Transactions on Sustainable Computing*, 6(4):655–666, 2020.

24 Manolis Marazakis, Marc Duranton, Dirk Pleiter, Giuliano TAFFONI, and Hans-Christian Hoppe. Hpc for urgent decision-making. 2022.

25 Jargalsaikhan Narantuya, Jun-Sik Shin, Sun Park, and JongWon Kim. Multi-agent deep reinforcement learning-based resource allocation in hpc/ai converged cluster. *Computers, Materials & Continua*, 72(3), 2022.

26 Azita Nouri, Philip E Davis, Pradeep Subedi, and Manish Parashar. Exploring the role of machine learning in scientific workflows: Opportunities and challenges. *arXiv preprint arXiv:2110.13999*, 2021.

27 Scott Dale Peckham. How can advances in ai help the working geoscientist? In *AGU Fall Meeting Abstracts*, volume 2018, pages IN12A–05, 2018.

28 Sebastián Risco, Germán Moltó, Diana M Naranjo, and Ignacio Blanquer. Serverless workflows for containerised applications in the cloud continuum. *Journal of Grid Computing*, 19:1–18, 2021.

29 Alberto Scionti, Paolo Viviani, Giacomo Vitali, Chiara Vercellino, Olivier Terzo, et al. Enabling the hpc and artificial intelligence cross-stack convergence at the exascale level. In *HPC, Big Data, and AI Convergence Towards Exascale: Challenge and Vision*, pages 37–58. CRC Press, 2022.

30 Nikolay A Skvortsov and Sergey A Stupnikov. A semantic approach to workflow management and reuse for research problem solving. *Data Intelligence*, 4(2):439–454, 2022.

31 Johannes Starlinger, Bryan Brancotte, Sarah Cohen-Boulakia, and Ulf Leser. Similarity search for scientific workflows. *Proceedings of the VLDB Endowment (PVLDB)*, 7(12):1143–1154, 2014.

32 Ozan Tuncer, Emre Ates, Yijia Zhang, Ata Turk, Jim Brandt, Vitus J Leung, Manuel Egele, and Ayse K Coskun. Diagnosing performance variations in hpc applications using machine learning. In *High Performance Computing: 32nd International Conference, ISC High Performance 2017, Frankfurt, Germany, June 18–22, 2017, Proceedings 32*, pages 355–373. Springer, 2017.

33 Matteo Turilli, Mark Santcroos, and Shantenu Jha. A comprehensive perspective on pilot-job systems. *ACM Computing Surveys (CSUR)*, 51(2):1–32, 2018.

34 Dan Wang, Qiang Du, Tong Zhou, Antonio Gilardi, Mariam Kiran, Bashir Mohammed, Derun Li, and Russell Wilcox. Machine learning pattern recognition algorithm with applications to coherent laser combination. *IEEE Journal of Quantum Electronics*, 58(6):1–9, 2022.

35 Carl Witt, Marc Bux, Wladislaw Gusew, and Ulf Leser. Predictive performance modeling for distributed batch processing using black box monitoring and machine learning. *Information Systems*, 82:33–52, 2019.

36 Carl Witt, Dennis Wagner, and Ulf Leser. Feedback-based resource allocation for batch scheduling of scientific workflows. In *2019 International Conference on High Performance Computing & Simulation (HPCS)*, pages 761–768. IEEE, 2019.

37 Michael R Wyatt, Stephen Herbein, Todd Gamblin, and Michela Taufer. Ai4io: A suite of ai-based tools for io-aware scheduling. *The International Journal of High Performance Computing Applications*, 36(3):370–387, 2022.

38 Zaynab Zahra, Zihao Li, and Rosa Filgueira. Laminar: A new serverless stream-based framework with semantic code search and code completion. In *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, pages 2009–2020, 2023.

39 Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. Data-centric artificial intelligence: A survey. *arXiv preprint arXiv:2303.10158*, 2023.

40 Shuo Zhang, Zhuofeng Zhao, Chen Liu, and Shenghui Qin. Data-intensive workflow scheduling strategy based on deep reinforcement learning in multi-clouds. *Journal of Cloud Computing*, 12(1):125, 2023.

3.4 Human Interaction and Accessibility

Rafael Ferreira da Silva (Oak Ridge National Laboratory, US)
Daniel Laney (Lawrence Livermore National Laboratory, US)
Paolo Missier (Newcastle University, GB)
Jedrzej Rybicki (Jülich Supercomputing Centre, DE)
Matthias Weidlich (Humboldt University of Berlin, DE)

In the rapidly evolving landscape of HPC, the integration of modern science applications necessitates the reevaluation of traditional workflow paradigms, particularly in the context of real-time interacting during workflow execution. Whether mediated by humans, AI, or specialized algorithms, these interactions introduce a new layer of complexity and opportunity across the experiment lifecycle when deployed on HPC environments. Such interactions serve critical roles, from real-time steering of ongoing computations to runtime recalibration of parameters, enabling adaptive adjustments that can feed back into the experiment itself for iterative improvement. Furthermore, these interactions enable data-driven decisions on cost/value trade-offs, thereby optimizing both computational and financial resources.

Building on the notion of real-time interactions, an example is the iterative process of workflow creation aimed at solving intricate problems based on initial specifications. In such cases, multiple workflow versions often emerge and are refined through expert-mediated (or possibly AI- or algorithm-mediated) interactions. The main goal is to perform an exploration of the solution space of a given problem rather than obtaining a single solution. Two key scenarios emerge in this context: (1) the independent exploration of alternative designs that can execute in parallel; and (2) a phased exploration where designs are sequentially refined based on observed results. To understand these challenges, we examine them within a framework of stakeholder interactions that occur throughout the standard lifecycle of modern HPC workflows.

Fig. 4 suggests a reference framework for placing three main stakeholders: Scientists, SysAdmins, and Data Scientists, in the context of current HPC-based workflow practice, and how their interaction contributes to extending the current batch-oriented model of workflow execution. Specifically, Scientists include a class of users who "own" the scientific problems that the workflow aims to address, and are responsible for the design of workflows that contribute to their solution. Examples of these are given below. A validation step is then typically required to ensure that the workflow fits the system requirements of the underlying HPC infrastructure. Optimizers and SysAdmin roles with expert knowledge of the HPC architecture and its resource allocation policies mediate this step, as suggested in the Figure. In a standard HPC batch submission model, the resulting validated workflow maps to jobs that are scheduled through a process of resource allocation.

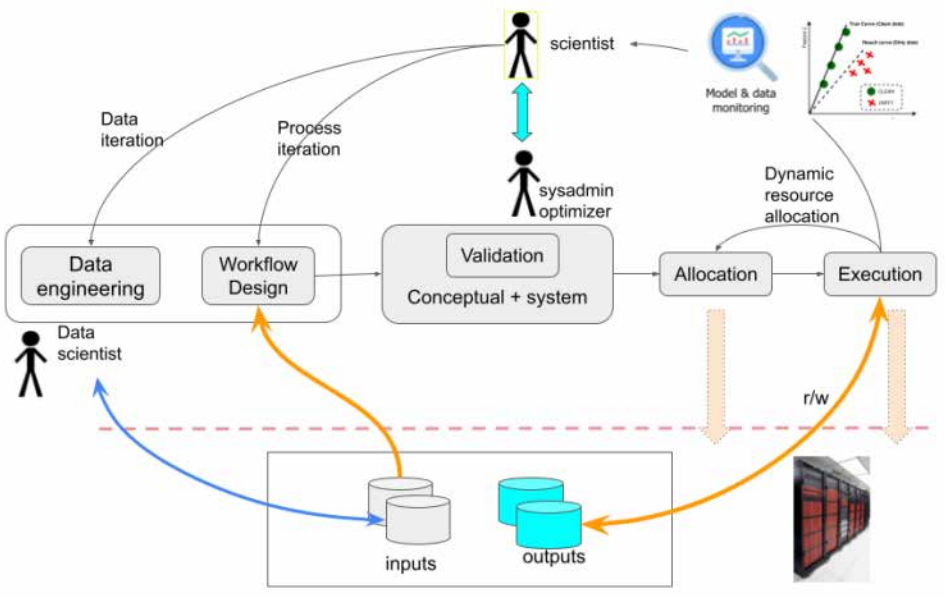

Figure 4 Simplified Framework of Stakeholder Roles in HPC Workflows, highlighting the collaboration between Scientists, SysAdmins, and Data Scientists to advance beyond standard batch-oriented workflow models.

We suggest that data-centric workflow design may bring new and potentially disruptive elements into the framework. Two main scenarios are common. Firstly, a familiar iterative exploration of the solution space by the scientists, which is required to converge on a stable workflow design for a new problem. This involves a repeated interleaving of four steps: batch execution, analysis of results, workflow refinement, and resubmission.

Workflow design naturally accounts for the type and structure of the underlying datasets (indicated as "inputs" in the Figure), however a realistic iterative refinement process needs to account for the need to refine the data, in addition to the workflow, at each iteration. To achieve this, we envision an additional Data Scientist role, who is responsible for data engineering and pre-processing tasks as required to align the input datasets to the workflow requirements. A notable class of problems where this additional step is needed is in the context of so-called Data-Centric AI [39], where the workflow is designed to deliver a model (for example, to predict some outcome), and the corresponding data tasks at each iteration include cleaning, correcting for bias, generating synthetic data to complement a training set, experimenting with alternative data imputation strategies, and more.

A second scenario occurs when each batch execution itself may be interrupted and broken down into separate components, with a decision process in the middle. For instance, during a process of parameter sweeping (exploration) that consists of an array of parallel tasks, it may be possible to identify promising regions of the parameter space and thus to steer the exploration, by pruning some of the tasks and starting new ones. In many cases this can be determined algorithmically, however we can also envision a human-in-the-loop scenario where the roles identified in the Figure directly participate in the decision process.

3.4.1 An Overview of the Challenges

Since HPC systems are primarily batch oriented, users evolve a cadence of job submission, analysis, and new submissions, sometimes over periods of days or weeks. With advances in ML/AI, we expect both human-in-the-loop and AI-driven workflows will become more

common. The challenge is that this new paradigm, of dynamic workflows, is at odds with existing HPC center policies and capabilities in many cases. Furthermore, human interaction with large scale HPC resources means that the benefit of the interaction or dynamism must be quite high to justify the costs associated with pausing, restarting, or spawning computations.

A key observation is that users often begin at a small scale where interaction is far less costly in terms of "wasted" compute, in order to understand and begin to design their workflow. However, at these small scales workflow execution tools are not needed, and can be an impediment due to their complexity, and are thus not introduced into the process until it is time to scale up the problems and number of simulations. ML/AI provides a way to introduce the "full" workflow environment early in the process, replacing expensive simulation with fast-running surrogates, and enabling interactive exploration with the entire software environment. We believe that a holistic, co-design approach is needed, in which workflows can be introduced early and scaled through the entire process of workflow creation, validation, and execution, from laptop scale to exascale.

In this section, we survey the challenges to human-in-the-loop and AI-driven workflows in current HPC centers:

- **Resource Allocation:** The integration of dynamic, AI-driven workflows into HPC systems faces a significant challenge due to the discrepancy between traditional HPC resource allocation policies and the requirements of these modern workflows. Traditional HPC centers, primarily designed for batch-oriented tasks, struggle to accommodate non-batch workflows like human-in-the-loop or AI-driven, data-centric processes. The absence of elasticity in resource allocation further exacerbates this issue, limiting the adaptability essential for dynamic workflows and leading to workload unpredictability. Consequently, there is an urgent need to develop HPC policies that support on-demand, AI-directed workflows, distinguishing between "compute projects" suited for batch processing and 'data projects' that demand a more flexible, interactive approach. Adopting a holistic, co-design strategy is crucial, allowing workflows to be introduced early and scaled efficiently from small-scale experiments to extensive executions, integrating ML/AI to create fast-running surrogates for interactive exploration and efficient resource utilization from the outset.
- **Data Management:** Data management and workflows overlap in many ways. Data are triggers, inputs, and outputs of scientific workflows; they also serve as an "integration layer" between workflow steps, and thus can help debug their execution. Workflows are also helpful in understanding the provenance of a particular dataset by describing what kind of processing led to its creation. For the intelligence (human or artificial) in the loop, the data plays a critical role. Decisions about how to proceed with the workflow are based on data, such as intermediate results. All this poses many challenges in terms of data management and workflow execution. The availability and placement of data plays a role in the execution plan of a workflow. The data created in the workflow must be made available to the external entity (intelligence in the loop) in a timely manner to enable interaction with the workflow execution. Finally, workflows and data have different lifetimes (the data remains important and valid even after the workflow execution has ended), so the resource manager responsible for workflow execution must be able to make both short-term and long-term decisions.
- **Debugging and Provenance:** This challenge centers around the need for deep checkpointing and meticulous action tracking. Essential to this challenge is the provision of provenance data in real-time, which would significantly enhance decision-making during workflow execution. However, this raises potential issues, such as the risk of real-time

provenance tracking interfering with the execution of workflows. Additionally, there are concerns regarding trust and the integrity of provenance data, which are crucial for reliable and verifiable scientific computations. Addressing these challenges requires a careful balance between providing detailed, real-time insights into workflow processes and ensuring that these mechanisms do not disrupt the efficient execution of complex computational tasks.

- **AI Integration:** This challenge revolves around preparing for high levels of unforeseen automation and the complexities it brings. To future-proof systems against this, a co-design approach is essential, where both hardware and software stacks are developed with workflow applications and their unique requirements in mind. Another critical aspect is ensuring that these AI-integrated workflows are explainable, which can be achieved by leveraging provenance and execution traces to provide clarity and understanding of the AI's decision-making process. However, this integration is not without its difficulties, as competing AIs within the same ecosystem may attempt to optimize at the application level for their own benefit, leading to uncertainties and complexities in workflow management. Addressing these challenges requires a sophisticated balance between advancing AI capabilities and maintaining control and transparency over automated processes in computational workflows.
- **Abstractions for Human Interaction:** This challenge involves facilitating human involvement at every stage, including creation, deployment, resource allocation, and execution. This requires designing roles and interfaces that cater to different stakeholders such as users, workflow experts, facility personnel, and data scientists, ensuring their input is valuable and feasible at various stages of the workflow. Additionally, AI can be integrated as a surrogate for "any human in the loop," performing tasks or making decisions in places where human intervention is typically required. Key types of user interaction that need to be abstracted include changing parameters of the workflow, starting or killing jobs based on real-time needs, and modifying the data, such as adjusting the samples used in the computation. The design of these abstractions must be intuitive and flexible, allowing for efficient and effective human-AI collaboration in managing complex computational processes.

In confronting the challenges of integrating advanced workflows into HPC systems, lessons from the history of HPC and its applications are invaluable. History shows that as HPC evolved, its increasing complexity often created barriers to automation and higher-order reasoning, a pattern now emerging in workflow systems as they accrue complex notations and concepts. This parallel suggests the need for a careful approach to developing workflow systems, ensuring they do not become so intricate that they hinder the very progress they are designed to facilitate. As we design these systems, it is crucial to consider the cost/value trade-offs in every decision, recognizing that "cost" can be multifaceted, encompassing computational resources, ease of use, and adaptability to future technologies. The key lies in learning from the past to build workflow systems that are robust, efficient, and accessible, thereby enabling them to be powerful tools in the advancement of scientific research and applications, rather than becoming cumbersome obstacles.

4 Future Perspectives

4.1 Optimization and Efficiency

Ana Gainaru (Oak Ridge National Laboratory, US)
Shantenu Jha (Rutgers University, US & Brookhaven National Laboratory, US)
Christine Kirkpatrick (San Diego Supercomputer Center, US)
Daniel Laney (Lawrence Livermore National Lab, US)
Wolfgang E. Nagel (TU Dresden, DE)
Jedrzej Rybicki (Jülich Supercomputing Centre, DE)
Domenico Talia (University of Calabria, IT)

For many years, HPC has been driven by extremely skilled individuals who are able to capture the complete execution of their scientific code, debug it, and optimize it to squeeze the last drops of performance out of the underlying infrastructures. We argue that this model is no longer sustainable. Scientific endeavors require multi-step workflows, the underlying infrastructures are becoming heterogeneous, and there are constant advances in methods that require the integration of new codes. As a first step, we need a cultural change that incentivizes the sharing of codes and workflows. In addition, the technical solutions should be designed so that the replacement of individual computational steps in workflows to incorporate new implementations or methods should be seamless. Workflow performance optimization will become a team effort, with scientists supported by HPC experts who consider not only the maximum performance of individual steps, but also a holistic view of the execution, including, for example, data movement. Future HPC centers will evolve in much the same way that cloud providers have evolved from offering Infrastructure-as-a-Service to higher abstractions such as Software-as-a-Service or Function-as-a-Service. Workflow descriptions can become the vehicle that drives this movement to higher abstractions and performance optimization as a team effort.

4.1.1 Performance Models and Reasoning

- We still do not reason well about performance of workflows, and facilities and procurements still focus primarily on individual application performance patterns.
- We often exclude data considerations, location, movement, etc.
- We need to build workflow benchmarks to help the community understand the more complex performance characteristics of workflows.
  - Throughput, makespan, responsiveness, time to solution
  - Understanding how these relate to each other so you can reason about them.
- Collective performance measures are different than for individual user/app.
  - We need to be able to see the entire workflow and reason.
  - We need to take a closer look at data, which can become bottleneck point.
- Empiricism: observing workflow behavior, and incrementally improving via learning from the behavior of the system.
  - Develop best practices.
  - Converge on principals.
- We thus need to have telemetry data on our workflows that allows this learning to occur; Post-mortem & Digital Twins.

4.1.2 Optimizing the Quality of Science

- We have to find definition on how to measure scientific quality for our workflows
- We need to define API's and architectures that enable future workflows to optimize under these currently unknown measures of quality
- We need to reason from examples:
  - Protein folding: standard, simple approach, is the baseline, approaches exist for interrogating the simulation and adjusting it to accelerate the folding.
    - Result of optimization is the same structure.
    - But time to solution can be 1000x shorter.
    - Still users often default to the simple, easy approach.
    - A Workflow system, perhaps augmented by AI.
- Preconditioners: many codes of iterative linear solvers in them (sometimes many such solvers).
  - Experts research preconditioners to accelerate solution.
  - Expert users choose among these for their problem, sometimes via experimentation.
  - We envision that a workflow system with AI could detect or predict solver behavior and choose preconconders to optimize execution.
- Coupled multi-application workflows under propagation of uncertainties.
  - In these workflows, users often set up and optimize each application in the workflow independently.
  - AI + workflow could understand parameter sensitivities and uncertainties, and optimize for scientific output, potential reducing cost of simulations.

4.1.3 A Vision for HPC & Data Centers

- **We have to think beyond simple scheduling and move towards multi-dimensional + temporal scheduling.**
  - It is not just about when to run but where you get the data from.
  - It needs to be spatio-temporal, co-location.
  - There is scheduling at the workflow level, workload level, and task level
  - The challenge is that it is relatively easy to schedule at one level or in one dimension, but how do enable usage of information across all levels to optimally schedule work.
  - Open research question: should information flow only upwards from lower levels (separation of concerns), or should it be globally available.

4.2 Lifecycle Management in Federated Workflows

Rosa M. Badia (Barcelona Supercomputing Center, ES)
Kyle Chard (University of Chicago, US)
Timo Kehrer (Universität Bern, CH)
Dejan Milojicic (HP Labs – Milpitas, US)
Fred Suter (Oak Ridge National Laboratory, US)
Sean R. Wilkinson (Oak Ridge National Laboratory, US)

"Federated workflows," a term also known as "cross-facility workflows" [1], represent a paradigm shift from traditional "grid computing." Grid computing typically refers to a distributed computing model where resources from various locations are pooled together to work on large-scale problems, often with a focus on maximizing resource utilization and computational power across a network of machines. In contrast, the concept of federated workflows extends beyond just resource sharing. It involves the integration and orchestration of workflows across different computational facilities, each possibly with its own unique environment, policies, and capabilities. This approach is not only about aggregating computational power but also about seamlessly executing complex processes that span across these varied environments, maintaining coherence and coordination throughout the workflow life cycle.

Understanding how federated workflows differ from traditional grid computing models, such as those exemplified by the Worldwide LHC Computing Grid (WLCG), is crucial. While grid computing primarily addresses the challenge of resource scarcity through distributed computing, federated workflows focus on the integration and interoperability of distinct computational workflows across various facilities. This distinction underscores a more nuanced approach to handling diverse computational tasks, data management practices, and workflow optimizations in a unified manner.

While it is possible to manage simpler federated workflows manually, the development of a dedicated federated engine could significantly enhance the efficiency and scalability of these systems. Such an engine would automate the execution of federated tasks, eliminating the need for building ad-hoc workflows for each new project. This would not only streamline the workflow execution across different facilities but also reduce the time and effort required for coordination and integration, thereby enabling more complex and dynamic federated workflows to be executed with greater ease and efficiency. The move towards federated workflows marks a significant evolution in how we approach distributed computing, emphasizing the need for sophisticated integration and orchestration tools to manage the complexities of modern computational tasks.

4.2.1 Use Cases for Federation

There are numerous use cases which demonstrate the value of federated workflows. We list a few here, but the list is not exhaustive.

- **Overflow / offload / cloud bursting.** Our first use case involves temporarily providing more compute and data resources upon demand by users, potentially in ways that are seamless and invisible to the user. For example, a user of an on-premise HPC system might require more resources than are currently available to run a workflow, and a potential solution might be to offload their workflow to execute on cloud resources.

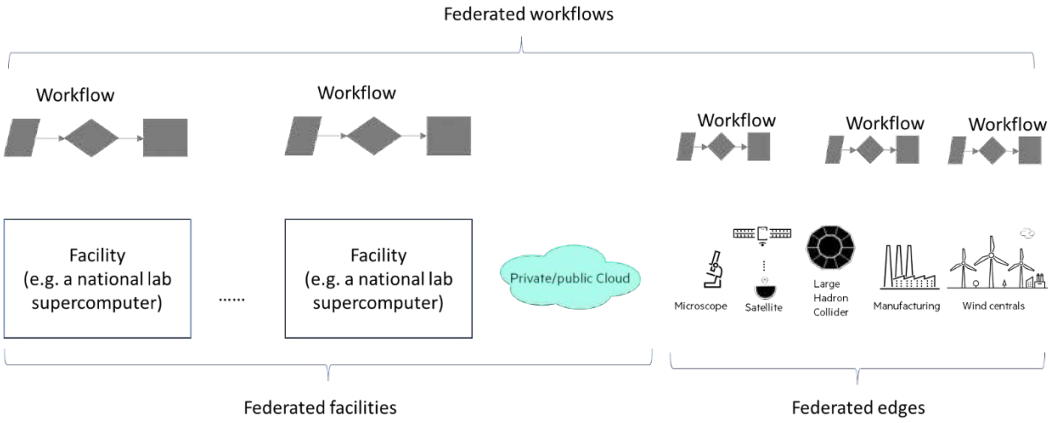

Figure 5 Conceptual Diagram of Federated Workflow Architecture. This figure illustrates the flow of data and tasks across a federated system, incorporating both private and public cloud resources.

- **Redundancy.** Similarly, during outages at HPC facilities, which may be planned or unplanned, users still need to execute their workflows. Federated workflows which execute across multiple facilities need to be capable of retargeting sites for execution dynamically in cases where facilities may be completely unavailable.
- **Urgent computing.** Sometimes, even offloading to the cloud may be insufficient to provide resources because the situation demands all possible resources from HPC facilities, clouds, and the like. Truly urgent situations like wildfires [2] and the recent COVID-19 pandemic [3] need to minimize time-to-solution in order to save lives, so they need to execute at high priority and the largest scales.
- **Internet of Things.** Federated resources also include very "small" compute and data resources like Internet of Things (IoT) devices. This use case involves edge devices which may have slow processors, power constraints, and reduced network connectivity as compared with HPC facilities. These devices can still be sufficient for federated learning, however, and this use case is therefore important for certain kinds of AI workflows [4].
- **Coupling edge computing with HPC.** There are also architectures in which edge devices can be used to process data in near-real-time, while coupling with analyses executed on true HPC resources. For example, a scientist at a light source might run parts of a workflow locally while collecting observational data at the same time that HPC resources are crunching numbers to help steer the data collection process [5].
- **Exotic hardware like quantum computers.** Federated workflows also take advantage of unique capabilities available at different facilities, such as quantum computers or classical HPC resources with exotic architectures. For example, OLCF has on-premises classical HPC resources, but the quantum computing resources are provided to its users by the cloud; these can still be used in the same workflow [6]. Different HPC facilities' flagship systems often have different architectures which lend themselves well to different kinds of problems, such as CPU-intensive or GPU-intensive tasks.
- **Collaborative data analysis.** In this use case, scientists use a shared dashboard to display analyses and visualizations of data produced by an HPC simulation. New analyses and visualizations can be dynamically added to that dashboard, potentially triggering/steering new computations (link). This use case thus relies on federated computing resources – supercomputer, analysis/visualization cluster, and cloud service for the dashboard.

- **Multi-instrument federation.** Another use case for federating facilities would allow not only for federating compute resources, but also for federating the instruments themselves. For example, an event-driven workflow in multi-messenger astrophysics could react to a neutrino detection event by DUNE by aiming telescopes at a region of interest.

4.2.2 Policies

In this section, we address the complex and essential role that policies play in the successful implementation and management of federated workflows. While technical aspects often dominate the concerns of engineers, policies are equally critical in making federated workflows viable and effective. These policies encompass a wide range of considerations, including governmental regulations like the General Data Protection Regulation (GDPR), which dictates who can access and execute workflows, as well as more technical aspects like the duration and frequency of workflow connections and invocations. Another key policy area involves identity verification, such as the requirements for Multi-Factor Authentication (MFA) and the timeframe within which it must be validated. Additionally, policies regarding user accounts at various facilities, including the necessity for signed agreements, play a crucial role in governing how federated workflows operate within and across different organizational domains.

The management of policies in federated workflows builds upon existing policies for individual workflows. However, when these workflows cross organizational boundaries, they begin to resemble aspects of past Grid computing work, albeit with unique challenges and considerations. Policies in federated workflows are critical not only for their deployment but also for their widespread adoption. Without well-defined and enforced policies, it becomes challenging to manage key aspects such as security, quality of service (QoS), sustainability, and availability in a manner that aligns with both organizational objectives and cross-organization collaboration. These challenges make the policy landscape for federated workflows a fertile ground for research, where the development of new policies and the adaptation of existing ones can significantly impact the efficiency and effectiveness of these complex computational ecosystems.

4.2.3 Workflow Federation Patterns/Motifs

We identified a series of execution patterns for future federated workflows. These patterns are built incrementally, starting from an existing and well defined scientific use case and then adding new features and/or constraints to this initial scenario.

- **One to many facilities.** Our initial use case corresponds to the execution pattern underlying the processing of data produced by the four High Energy Physics experiments deployed on the Large Hadron Collider at CERN (i.e., ATLAS, CMS, ALICE, and LHCb). These experiments rely on the worldwide LHC computing grid (WLCG) for about two decades. This computing grid comprises multiple computing centers across the world that are federated to execute a single workload composed of millions of independent jobs (MC simulations). Each center has a contractual commitment to provide resources with a given availability from other sites to be able to run on their sites. There is also an upstream control of the distribution of the jobs among the sites according to their current load and a common authentication overlay based on proxy certs to enable the federated infrastructure. However, the management of the federation is human-engaged.

- **Federated facility.** To automate the management of a "one to many facilities" federation and form a truly federated facility, we consider collecting data to train an AI model that optimizes what jobs to run and where to run them, in terms of performance. Additionally, we propose that all contractual agreements across the different participants of the federated facility are verified at any point of time during the execution of the workload.
- **Sustainable federated facility.** The next stage is then to add a sustainability dimension, in the environmental and social meanings of the term, to the proposed federated facility. The objective there is to complement the mapping and scheduling decisions based solely on performance by using AI to help decide not only where to execute the workload according to the environment sustainability profile of the different sites but also whether it is better to not run some subset of the workload than wasting precious resources.
- **Federatable workflows.** We then introduce the concept of federatable workflow, i.e., a workflow published by one of the facilities belonging to the federation that can be composed with other federatable workflows to form a more complex federated workload. A federatable workflow can be considered as a end point of the end-to-end federation along with computing and experimental (e.g., telescopes, microscopes, light sources, etc) facilities.
- **Data-constrained federated workflows.** Sometimes data cannot move due to governance/policy constraints, such as data produced in the European research space or in the case of biomedical data). In that case, it becomes necessary to "send compute to the data". Such constraints on data movements add geographical constraints that shape the workflow and further motivate a federation of workflows.

4.2.4 Contracts for Federated Workflow

Because federated workflows can cross organizational boundaries, some support for contracts is required to formalize the bindings between invocations that can cross administrative, legal, financial, and other boundaries. These contracts could be very simple, such as offline agreement that all dynamic bindings between certain entities are either permissible or not, or they could be substantially more complex in terms of financial, legal and other dynamic bindings that need to take place at any location.

The non-trivial aspect comes from the fact that QoS can have different meanings in different organizations. For this reason this remapping can be built into the federation engine which will be able to translate requirements, observe compliance and address disputes, violations, etc. We can express QoS and authorization as a part of the federation at federation time, so that we know whether from responsiveness, throughput, resources, etc. the federation point will work for the needs of the workflow.

Addressing privacy concerns of regional areas (Europe, US, Asia) or individual entities (e.g., farms (about their crops), businesses (e.g., medical)) can also be done at the federation engine. Federation engines can become a basis for federation of facilities, workflows, and anything in between (devices, workflow tasks).

References

1 Katerina B Antypas, DJ Bard, Johannes P Blaschke, R Shane Canon, Bjoern Enders, Mallikarjun Arjun Shankar, Suhas Somnath, Dale Stansberry, Thomas D Uram, and Sean R Wilkinson. Enabling discovery data science through cross-facility workflows. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3671–3680. IEEE, 2021.

2 Ilkay Altintas, Jessica Block, Raymond De Callafon, Daniel Crawl, Charles Cowart, Amarnath Gupta, Mai Nguyen, Hans-Werner Braun, Jurgen Schulze, Michael Gollner, et al. Towards an integrated cyberinfrastructure for scalable data-driven monitoring, dynamic prediction and resilience of wildfires. *Procedia Computer Science*, 51:1633–1642, 2015.

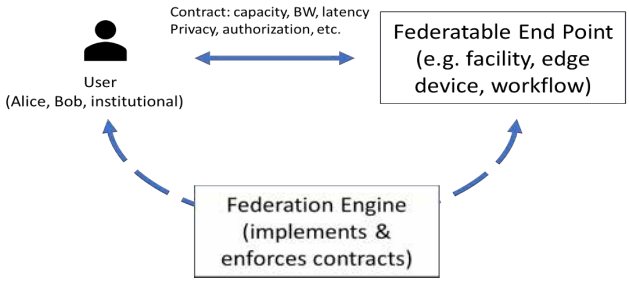

Figure 6 Schematic Representation of a Federation Engine within a Federated Workflow System enforcing contracts.

3 Hyungro Lee, Andre Merzky, Li Tan, Mikhail Titov, Matteo Turilli, Dario Alfe, Agastya Bhati, Alex Brace, Austin Clyde, Peter Coveney, et al. Scalable hpc & ai infrastructure for covid-19 therapeutics. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pages 1–13, 2021.

4 Omer Rana, Theodoros Spyridopoulos, Nathaniel Hudson, Matt Baughman, Kyle Chard, Ian Foster, and Aftab Khan. Hierarchical and decentralised federated learning. *arXiv preprint arXiv:2304.14982*, 2023.

5 James E McClure, Junqi Yin, Ryan T Armstrong, Ketan C Maheshwari, Sean Wilkinson, Lucas Vlcek, Ying Da Wang, Mark A Berrill, and Mark Rivers. Toward real-time analysis of synchrotron micro-tomography data: accelerating experimental workflows with ai and hpc. In *Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and AI: 17th Smoky Mountains Computational Sciences and Engineering Conference, SMC 2020, Oak Ridge, TN, USA, August 26-28, 2020, Revised Selected Papers 17*, pages 226–239. Springer, 2020.

6 Samuel T Bieberich, Ketan C Maheshwari, Sean R Wilkinson, In-Saeng Suh, Rafael Ferreira da Silva, et al. Bridging hpc and quantum systems using scientific workflows. *arXiv preprint arXiv:2310.03286*, 2023.

4.3 Advanced Techniques and Innovations

Laure Berti-Equille (IRD – Montpellier, FR)
Rosa Filgueira (University of St Andrews, GB)
Ulf Leser (HU Berlin, DE)
Bertram Ludäscher (University of Illinois at Urbana-Champaign, US)
Paolo Missier (Newcastle University, GB)
Jeyan Thiyagalingam (Rutherford Appleton Lab, GB)

4.3.1 LLMs for workflows

Large Language Models (LLMs), such as UniXCoder, codeBERT, codex, GPT-4, GraphCodeBERT, CodeT5, have already proven their utility in enhancing developers' capabilities to manage and produce more efficient code across a range of domains. These models have opened up exciting possibilities for automating and optimizing software development processes. However, it is equally pertinent to recognize the immense potential LLMs hold in the

domain of scientific workflows. Here, we summarise the different ways LLMs can be used to improve the management and efficiency of scientific workflows. Below, we present a list of use cases in which LLMs can offer benefits to the scientific workflow community:

- **Discovering Similar Workflows.** Large language models can help researchers discover similar workflows in their domain. By analyzing descriptions or code snippets of existing workflows, these models can identify patterns and similarities, aiding researchers in finding relevant examples and benchmarks for their own projects. This functionality can significantly reduce the time and effort required to design new workflows.
- **Identifying Similar Workflow Components.** Within a given scientific workflow, there are often recurring components such as data preprocessing, analysis modules, and visualization tools. Language models can assist in identifying similar components across different workflows. This capability can enhance code reusability and promote the sharing of standardized components within the scientific community.
- **Autocompleting Tasks and Workflows.** Language models can serve as intelligent code completion tools, making it easier for researchers to write and refine workflow scripts. As scientists input their desired tasks or steps, these models can suggest code snippets, offer parameter recommendations, and assist in handling dependencies, resulting in more efficient and error-free workflow development.
- **Describing and Summarizing Workflows and Tasks.** Scientific workflows often involve intricate data manipulation and analysis procedures that can be challenging to document comprehensively. Large language models can automatically generate human-readable descriptions and summaries of workflows and individual tasks. This enhances the accessibility of the workflow for collaborators and future reference.
- **Interoperability Across Workflow Engines/Frameworks.** Diverse workflow management systems often feature distinct syntax and structures. Large language models can simplify the process of translating workflows from one platform to another. Researchers can input their workflow in one system's language, and the model can generate equivalent code in the target platform's format, streamlining the migration process and minimizing errors.
- **Workflow Optimization and Tuning.** Large language models could also assist in optimizing and fine-tuning scientific workflows. By analyzing performance data and user-defined goals, these models can suggest improvements in resource allocation, parallelization strategies, and parameter tuning to achieve faster and more efficient execution.

4.4 Collaboration and Community Building

Ilkay Altintas (San Diego Supercomputer Center, US)
Silvina Caino-Lores (INRIA – Rennes, FR)
Rafael Ferreira da Silva (Oak Ridge National Laboratory, US)
Christine Kirkpatrick (San Diego Supercomputer Center, US)
Matthias Weidlich (HU Berlin, DE)

4.4.1 Community Building Initiatives

In the burgeoning field of HPC+AI workflow integration, there is a notable absence of dedicated venues for community engagement and knowledge exchange. To fill this void, initiatives can potentially evolve from Workflow Community Initiatives (WCI) and interdisciplinary-oriented venues, creating spaces where best practices for workflows, including pattern submissions linked to workflow module or pattern commons, can be discussed and refined. These venues could operate on a model of continuous submission, allowing for the real-time tracking of emerging or popular patterns. However, publishing these insights in a manner that engages communities unfamiliar with workflow intricacies remains a question. Furthermore, identifying key stakeholders within this ecosystem is essential to understand how systems should be architected to facilitate the interactions between these parties.

Addressing cultural differences in workflow structures is another significant challenge. Scientists often create monolithic workflows that are resistant to the modular, adaptable approaches emerging in modern workflow composition. To encourage the adoption of new practices, there must be tangible incentives for end users, and equitable methods to transfer development best practices from the computer science community to practitioners. Advocating for these cultural shifts at higher levels, such as funding agencies, is also critical and requires champions who can effectively communicate the value of these changes. Capturing the varied needs of intersecting communities, such as domain science and AI, presents both a challenge and an opportunity to distill knowledge from specialized groups.

Defining a specification for workflow assessment and evaluation tailored to specific use cases is an ongoing challenge. Distinguishing "workflow challenges" from benchmarks could help, with metrics that measure not only technical performance but also innovation, as exemplified by the DEBS community's approach to streaming datasets. Competitions, like those on Kaggle or potential student contests at focused venues, could concentrate on workflow patterns and their applications. The suitability of existing repositories for static workflows, such as WorkflowHub, Snakemake, BioBB, and MyExperiment, needs to be evaluated against the requirements of modern, dynamic workflows. The Workflow Module Commons on top of WCI could serve as an equitable service for building blocks, as suggested by Ilkay's work, but this raises the policy challenge of engaging institutions to host such services. AI could support automatic workflow generation, optimizing performance, and composing workflows from existing building blocks based on descriptions. However, this introduces the challenges of collecting training data, defining workflow requirements—particularly in terms of data and system specifications—and ensuring the certification of generated workflows and predictions for resource allocation.

Recommendation. build a community ecosystem for adding value to the collection, documentation and sharing of workflows, use cases and building blocks, promoted by supporting publications to derive best practises and dedicated venues hosting dedicated events (e.g., hackathons to apply best practises and patterns, workflow challenges).

4.4.2 Technical Challenges for Adoption of Data-Centric AI+HPC Approaches

The shift towards data-centric AI+HPC approaches poses several technical challenges for adoption, particularly in environments that have traditionally favored a task-centric perspective. A key question arises: how can the HPC community, which typically prioritizes tasks, be incentivized to adopt a data-centric approach that is essential for integrating AI components into workflows? This paradigm shift requires not only a recognition of the efforts involved in creating data-centric building blocks and generating valuable datasets but also strategies to democratize data within HPC, making it more accessible for use, sharing, and retrieval. Currently, many computing facilities operate in silos, with distinct responsibilities for computing or data management. Some institutions, like UCSD, have successfully developed institutional repositories, but this is not yet widespread. The challenge extends to the discovery of data and repositories linked to HPC, which often falls prey to resource prioritization issues rather than technical feasibility. To address these hurdles, there is a critical need to educate the next generation of HPC professionals and researchers in data-centric workflow thinking, laying a foundational understanding that will drive the future of integrated AI+HPC solutions.

Recommendation. Incensitivize integrating the role of data at the same level as tasks in HPC communities. Promote this idea in workflow venues, promote BOFs integrating HPC/data gap, foster partnerships with industry, incentivise publishing in data journals systematic description of datasets (e.g., DOIs for workflow components). Respond to RFIs, disseminate this Dagstuhl report, try to influence solicitations to ask for data-driven architectures, give incentives to be data forward, focusing on workflows as the interface/application layer for users/researchers.

Recommendation. User-centric software (workflows) should have simplified ways of showing provenance and habituating researchers to looking for verified/trustworthy data via provenance iconography that can be expanded for full details.

Current workflow systems are generally unprepared to handle the intricacies of the substantial hurdles in transitioning to workflows that are not just propelled by input or intermediate data, but also by metadata. This raises the question of whether there is an opportunity to reshape the design of these systems to integrate a data-focused perspective more thoroughly. A significant challenge lies in breaking away from established practices in HPC workflow development and system design, which have traditionally undervalued the role of data. Overcoming this will require a paradigm shift towards recognizing data not only as a passive element but as a dynamic driver of workflow processes, necessitating a fundamental re-evaluation of how data is integrated and leveraged within the HPC environment.

Recommendation. Machines will need to evolve in their architecture to accommodate hybrid workloads. Workflow managers shall be enabled to incorporate enriched data with metadata capturing properties of data to drive workflow management decisions in an ecosystem that encompasses system and task execution information.

Participants

- Ilkay Altintas
San Diego Supercomputer Center – La Jolla, US
- Rosa Maria Badia
Barcelona Supercomputing Center, ES
- Laure Berti-Equille
IRD – Montpellier, FR
- Silvina Caino-Lores
INRIA – Rennes, FR
- Kyle Chard
University of Chicago, US
- Rafael Ferreira da Silva
Oak Ridge National Laboratory, US
- Rosa Filgueira
University of St Andrews, GB
- Ana Gainaru
Oak Ridge National Laboratory, US
- Shantenu Jha
Rutgers University – Piscataway, US & Brookhaven National Laboratory – Upton, US
- Timo Kehrer
Universität Bern, CH
- Christine Kirkpatrick
San Diego Supercomputer Center – La Jolla, US
- Ulf Leser
HU Berlin, DE
- Bertram Ludäscher
University of Illinois at Urbana-Champaign, US
- Dejan Milojicic
HP Labs – Milpitas, US
- Paolo Missier
Newcastle University, GB
- Wolfgang E. Nagel
TU Dresden, DE
- Jedrzej Rybicki
Jülich Supercomputing Centre, DE
- Frédéric Suter
Oak Ridge National Laboratory, US
- Domenico Talia
University of Calabria, IT
- Jeyan Thiyagalingam
Rutherford Appleton Lab. – Didcot, GB
- Matthias Weidlich
HU Berlin, DE
- Sean R. Wilkinson
Oak Ridge National Laboratory, US