Driving Next-Generation Workflows from the Data Plane

Frédéric Suter Rafael Ferreira da Silva Ana Gainaru Scott Klasky

Oak Ridge National Laboratory, Oak Ridge, TN, USA
{suterf, silvarf, gainarua, klasky}@ornl.gov

***Abstract*—We observe the emergence of a new generation of scientific workflows that process data produced at a sustained rate by scientific instruments and large scale numerical simulations. This data is consumed by multiple analysis, visualization, or Machine Learning components not only to enable inference and justify the scientific program, but also to monitor and steer the evolution of these experiments. In such workflows, moving intermediate data efficiently is key to performance, more than efficiently scheduling computational tasks. However, most traditional workflow management systems focus on optimizing task scheduling and then deal with data management, assuming a "move little, compute for long" model, which makes them unfit to the efficient management of this new generation of workflows.**

Therefore, we advocate for a new way to manage scientific workflows. We propose to consider an efficiently and independently managed *data plane* that can store and stream data. Workflows compute components, in the *application plane* can then interact with the data plane, abstracted from complexities of data management. Then, the role of a workflow management system would become that of a *control plane* that allows users to connect services together to execute the workflow and manages connections between the application and data planes.

In this position paper, we characterize several next-generation workflow motifs and describe how their interaction with the data plane is a challenge to traditional workflow management systems. Then, we express a set of requirements that a workflow management system should meet to efficiently manage next-generation workflows at different scales. Based on these requirements, we expose our vision of driving next-generation workflows from the data plane and list remaining open challenges.

***Index Terms*—Workflow management, data management**

I. Introduction

Scientific workflows that compose multiple data-driven computational tasks to produce a scientific result, have become a cornerstone of modern scientific computing [1]. In many scientific fields, workflows have underpinned some of the most significant discoveries of the last two decades and will play a crucial role in the data-oriented and post-Moore's Law computing landscape. Many of these workflows exhibit high computational, storage, and/or communication demands. Their execution thus usually targets a wide range of computing infrastructures, from edge devices to exascale computers [2].

This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a non-exclusive, paid up, irrevocable, world-wide license to publish or reproduce the published form of the manuscript, or allow others to do so, for U.S. Government purposes. The DOE will provide public access to these results in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

To manage the composition, planning, orchestration, and efficient execution of such workflows, a plethora of Workflow Management Systems (WMSs) have been proposed over the years [3]. The first aim and focus of a vast majority of these systems is to optimize the orchestration of the compute part of the workflows, i.e., minimizing the completion time of the workflow. The rationale is that for a long time, workflows had a "move little, compute for long" structure: compute resources were the bottleneck and data transfers could be afforded. However, three recent evolutions must lead to a profound reevaluation of this aim and focus of WMS design: (i) the evolution of processors following the end of Moore's law and the generalization of GPUs led to a dramatic increase of the number of available cores. The compute resources is now the most affordable one, which can even be exploited to reduce communication by introducing redundant computations [4]; (ii) the relative stagnation of network and I/O bandwidth in comparison with the increase in compute performance, making data movement the new bottleneck in many scientific experiments [5]; and (iii) the emergence of **next-generation workflow motifs** in which data access and movement are the most complex tasks to optimize [2].

Therefore, we advocate in this position paper for a new way to manage scientific workflows in which we separate the concerns between a **data plane** that can store, stream, and move data and metadata, an **application plane** in which workflow compute components can either put data in or get data from this data plane, and a **control plane** that connects services together to run the workflow and manages the interactions between the application and data planes. In this approach, each plane can be efficiently and independently managed and hide complex optimization techniques from the other planes.

The contributions of this paper are to:

- Characterize a series of next-generation workflow motifs and describe how their interaction with the data plane is a challenge for traditional workflow management systems (Section III).
- Express a set of requirements that the control and data planes have to meet to efficiently manage such next-generation workflows (Section IV)
- Expose our vision for a workflow ecosystem in which next-generation workflows can be driven from the data plane, based on the expressed requirements (Section VI).
- List remaining open research challenges to achieve this vision (Section VII).

This paper is organized as follows. In Section II, we present what are traditional scientific workflows and how the majority of workflow management systems handle them. Then, in Section III we detail five generic motifs of next-generation workflows and present the challenges they cause to traditional workflow management systems. In Section IV, we list a series of requirements to the development of a next-generation workflow management system and review how related work fulfill these requirements in Section V. Then, we detail our own vision of such a system in Section VI. Finally, we list remaining open research challenges in Section VII before summarizing our work in Section VIII.

II. Traditional Workflows and Workflow Management Systems

Traditional scientific workflows are usually structured as a Directed Acyclic Graph (DAG) such as the one depicted in Fig. 1. Vertices correspond to computational tasks, that take some data, stored in files, as input and produce intermediate, or final, data as output, while the edges represent the flow and control dependencies between the tasks.

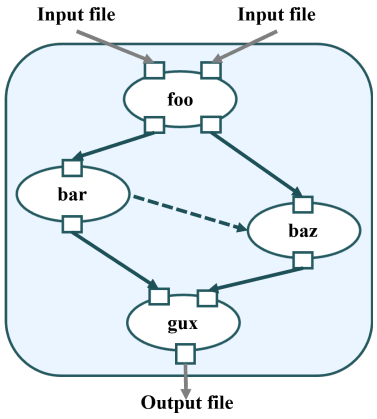

Fig. 1. Generic motif of a traditional workflow represented by a Directed Acyclic Graph. Each compute task takes one or several files as input, and produces one or several files as output. Files transfers (plain arrows) or control dependencies (dashed arrow) between tasks enforce their execution order.

A vast number of workflow management systems [3] have been proposed to describe such workflows in an abstract way, i.e., independent of specific inputs, versions of each components, or mapping to resources, and facilitate the automation of their execution. The workflow engine can then plan and orchestrate a more concrete version of the workflow once the specific inputs and available resources are known. Three main properties have been identified in [6] to characterize these workflow management systems. The *execution model* defines how workflow components interact with each other, i.e., sequentially, concurrently, iteratively, in a strongly coupled way, or with external steering. Then, the *data access methods* supported by the WMS range from in-memory transfers to the use of the entire storage hierarchy through message passing. The third property is related to the capacity of the workflow system to deploy workflows over multiple computing sites.

This study concludes that the majority of the considered workflow management systems focus on the support of workflows based on a "read-compute-write" task model using file-based transfers between components. This traditional model is also at the core of the proposition of the Common Workflow Language (CWL) open standard [7] which has been designed for workflows that loosely connect multiple command-line tools together to perform a complex data analysis. However, this model does not consider that data can be produced throughout the execution of a task and may need to be immediately transferred, or streamed, to another workflow component. This kind of data exchange pattern is central to a new class of scientific workflows, which we describe in the next section, that require new workflow and data management techniques to be executed efficiently.

III. Next Generation Workflows Motifs

In this section, we describe five workflow motifs originating from diverse scientific domains. For each motif, we provide an illustration of their structure and compelling use cases exhibiting this motif. Additionally, we outline the challenges posed by these motifs to traditional workflow management systems. It is important to note that the challenges discussed in this section are not mutually exclusive, and several of them may arise within a single workflow application.

A. Strong Code Coupling and Analytics

Code coupling is a traditional method to implement multi-physics simulations. Each code executes its own numerical simulation and interacts with a central component, the coupler, that enables the sharing and transformation of data between codes and manages the execution and synchronization of the different codes. In practice, such a code coupling can be implemented by running all the codes as a single MPI application with distinct communicators for each of the components or by exchanging data through files stored on a shared file system.

The execution of multi-physics simulations is evolving towards more complex workflows as in addition to the traditional strong coupling of simulation codes, different performance diagnostics, data analytics, and visualizations components are loosely coupled to the simulations as illustrated by Fig. 2.

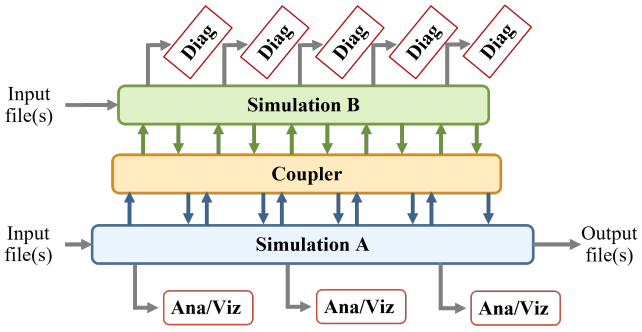

Fig. 2. Generic motif of multi-physics and analytics workflow. Two (or more) numerical simulations are strongly coupled and periodically exchange data through a coupling component. Simulations can also be loosely coupled with analytics, visualization, or performance diagnostic components.

These additional workflow components help scientists monitoring the execution, getting insight on the occurring phenomena, and potentially reacting early to abnormal behaviors.

This workflow motif arises in diverse scientific domains such as fusion science or climate modeling. For instance, the Whole Device Modeling application (WDMApp) project [8] aims to develop a high-fidelity model of magnetically confined fusion plasmas by coupling two highly scalable gyrokinetic codes: XGC [9], a particle-in-cell code for treating the edge plasma, and GENE [10], a continuum code for the core plasma. Then, the data produced by each code can be monitored to detect anomalies, trigger a specific analysis, enable command-and-control of the simulations, or be visualized online on a collaborative dashboard to get early insights on the evolution of the plasma physics. Such complex simulation and analysis workflows are urgently needed to plan experiments on ITER and optimize the design of future fusion facilities.

The Energy Exascale Earth System Model (E3SM) [11] aims at better understanding how water cycle, biogeochemistry, and cryosphere systems govern variability and changes in water availability and storms, air and stream temperature, and coastal flooding and sea level rise. Obtaining answers to the scientific questions underlying these phenomena is crucial to try to limit the effects of global warning. To this end, E3SM strongly couples models of oceans, atmosphere, land surface, rivers, and land and sea ice. E3SM models are then loosely coupled to the Community Data Analysis Tools for analysis and visualization purposes and to the Community Diagnostics Package for performance analyses and provenance capture.

Challenges for traditional WMSs

C#1: Enable the strong-coupling of multiple applications without having to have them running in the same execution domain, e.g., a shared MPI communicator.

C#2: Consume data periodically produced by a compute task before its completion or without having to split the task in multiple sub-tasks and flow dependencies.

C#3: Dynamically add a new analysis, visualization, or diagnostics component at runtime.

B. Ensemble Contributing to a Common Data Set

In many domains, obtaining an answer to a scientific question requires the execution of multiple and independent computational tasks. Each of these tasks can either be a single numerical simulation or a more complex pipeline composed of several stages. The collective execution of such distinct yet connected tasks is usually denominated as an ensemble application [12]. The processing of each individual set of results provides some partial insight and their combination allows scientist to obtain the desired answer.

For this motif, we consider a slight variation of this definition of an ensemble where each individual instance in the ensemble contributes to a common data-set. The difference is that multiple analysis components can then query this data set with a different access pattern, as shown in Fig. 3. For instance, a 3D volume summarizing contributions of (x,y)-planes along the z-axis can be read in an adversarial access pattern along the y-axis, or individual contributions may have to be aggregated in a certain way before getting scientific insight.

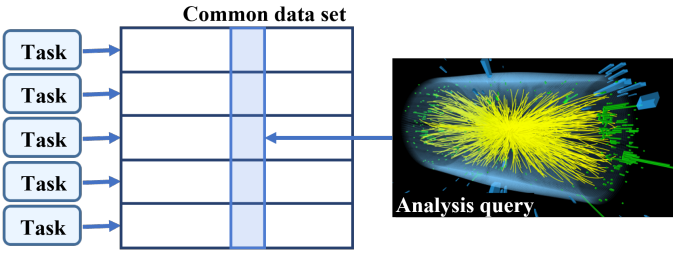

Fig. 3. Generic motif of an ensemble of independent computational tasks that contribute to a common data set. Data is then queried by analysis components using a different access pattern that can span over multiple contributions.

This type of access to a common data set is a typical pattern in high energy physics. For instance, Celeritas [13] is a Monte-Carlo particle transport code used to simulate the behavior of high energy physics experimental detectors. Celeritas collects electro-magnetic particle showers from either primary collisions or subsequent hadronic interactions and simulates their transport on GPUs. The stochastic nature of Monte-Carlo applications requires to run an ensemble of simulations to obtain statistically significant results. The problem of data aggregation is twofold in this application. Each instance of Celeritas produces data that has to be combined at the ensemble level, which is a common practice. However, to obtain these results, it is necessary to reconstruct particle tracks from the hits, i.e., the fact that one particular layer of the detector "saw" a particle going through. These hits are independently generated and written by each GPU thread. This reconstruction of tracks thus accesses to the common data set with a very different pattern than the one used to write each individual contribution.

Challenges for traditional WMSs

C#4: Perform in-transit data reorganization to find a good balance between read and write performance, e.g., changes to the data layout between production and writing.

C#5: Aggregate data into a consistent data set across instances in an ensemble execution and across ensembles.

C. AI/ML-based Steering

The idea of computational steering [14] is to break the prepare-execute-analyze cycle traditionally used to run a series of experiments leading to a scientific discovery. It allows scientists to change parameters of their experiment at runtime and immediately receive feedback on the effect of that change. With the advent of AI/ML models, an ensemble of simulations can be steered by leveraging such models to either avoid undesired results, increase the accuracy of the generated results, or fill gaps in the covered parameter space. In workflows such as the one in Fig. 4, the training of the AI/ML model is performed beforehand and only the inference part is used to take quick decisions for the next batch of simulation instances.

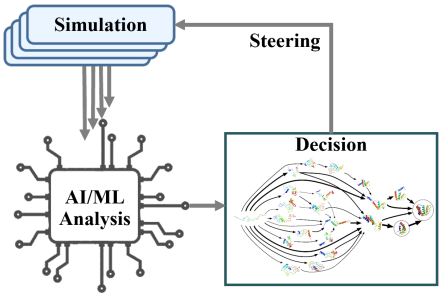

Fig. 4. Generic motif of the AI/ML-based steering of an ensemble of simulations. The results of independent simulation runs are fed to an AI/ML model to give insight to a human-in-the-loop or an autonomic decision-making process and decide on the next set of simulation parameters to consider.

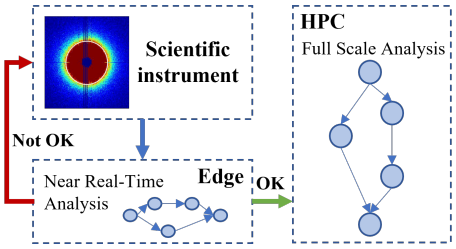

Fig. 5. Generic motif of an edge-to-HPC workflow in which a near real-time analysis is executed on the edge using reduced data to enable quick decision making. Depending on the outcome, raw or pre-processed data may be transferred to an HPC facility for large-scale processing. The latter may have different input parameters or configuration from the one generated by the scientific instrument.

This workflow motif is commonly used in domains such as molecular biology to get insights into how proteins fold. Multiple molecular dynamics (MD) simulations are executed to study the physical movements of atoms and molecules. The initial input of each instance in the ensemble is selected from a predetermined set of low dimensional representations of the studied system. AI/ML-based steering is then used to prevent MD simulations to get stuck in metastable states, and thus ignore entire regions of the solution space. Enhanced sampling techniques are thus periodically applied to stop and steer the ensemble towards new starting points [15].

Challenges for traditional WMSs

C#6: Dynamically plan efficient resource allocations based on decisions made at runtime by the AI/ML model.

C#7: Conciliate adversary data access patterns, i.e., high performance parallel writes by the simulation and sequential reads of many files by the AI model, on a shared file system.

D. Edge-to-HPC Multi-Stage Analysis

Scientific instruments, such as electron microscopes in chemistry or light sources (e.g., the Spallation Neutron Source [16]) in material sciences, take advantage of high-throughput, highly available computing resources located at the edge to analyze the produced data in near real-time. The outcome of this near real-time data processing on edge resources triggers either the transfer of the entire raw data set to an HPC facility to be processed by a full-fledged analysis workflow or the reconfiguration of the experimental parameters of the instrument until a desirable visualization of the studied sample is obtained. The response time of the first workflow stage is very important, as it impacts the time spent using the instrument which is often limited and highly valuable. However, such instruments can produce terabytes of data per hour whose processing is hardly compatible with the near real-time constraint of the initial analyses.

As illustrated in Fig. 5 the first stage of a multi-stage cross-facility workflow (i.e., that spans multiple facilities) is executed under a near real-time constraint. The corresponding sub-workflow may work on a highly reduced (or transformed) version of the data produced by the instrument to satisfy tight time constraints. Indeed, by processing a smaller data set, the execution time of this stage can be significantly decreased. However, the reduced data has to capture the main quantities of interest with sufficient accuracy to enable decision making when the experimental setup has to be reconfigured.

Challenges for traditional WMSs

C#8: Coordinate the execution of a workflow across multiple facilities (i.e., scientific instrument and HPC center).

C#9: Adapt the data reduction level to the time constraints of each stage of the analysis workflow.

E. Digital Twins

The last considered workflow motif can be seen as a combination of the previous motifs. It corresponds to the development and usage of a digital twin of a scientific instrument as illustrated in Fig. 6. Data coming from a mixed ensemble of HPC simulations and surrogate AI/ML models and from the instrument itself is used for the online training of the digital twin. The model is updated based on observed discrepancies between what it predicts and feedback from the simulation. Digital twins are used to steer the simulation and allow for command-and-control on the instrument by tuning parameters automatically in near-real time or with human-in-the-loop interventions. They also allow scientists to quickly get insights that could not be obtained with high-fidelity simulation or observational models due to time or resource constraints.

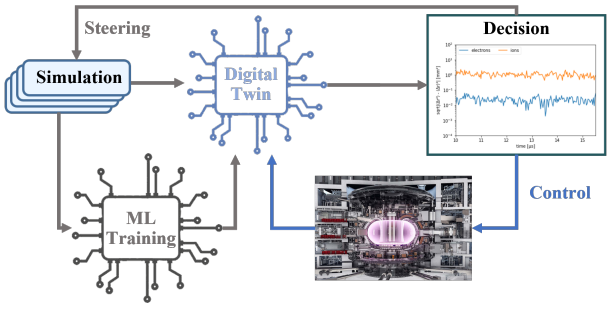

Fig. 6. Generic motif of a workflow including a digital twin. Simulation results, surrogate models, and inputs from the scientific instrument itself are used to train the Digital Twin which is then used to steer the next simulation runs and control the configuration of the instrument.

We can find this motif in the medical and bioinformatics domains where multiscale simulations are combined to AI/ML models to get new insights into disease mechanisms and to help identify new targets and treatment strategies [17], [18]. The resulting digital twin is used to steer the expensive simulation code towards promising targets and help avoiding being trapped around a local optimum. In fusion science, such digital twins are essential to plan experiments on ITER and optimize the design of future fusion facilities.

Challenges for traditional WMSs

C#10: Dynamically retrain the digital twin when its behavior diverges from the simulation outcomes.

C#11: Enable command-and-control of a scientific instrument based on the outcomes of the digital twin.

IV. Requirements to Develop an Integrated Workflow and Data Ecosystem

In this section, we describe four requirements that a modern integrated workflow and data ecosystem should meet to drive next-generation workflows from the data plane and address the challenges we highlighted in the previous section.

Req. #1: Service Oriented Approach of the Data Plane

To answer several of the aforementioned challenges, it is necessary to abstract the data movement and storage methods from both the application that produces data and the different workflow components that consume it. This requires to adopt a service oriented approach, with a dedicated service controlling the data plane with which the workflow and the workflow management system can interact through a publish/subscribe mechanism. Data producers only need to focus on publishing data into the data plane, from which subscribers can easily retrieve and process it. This mechanism addresses challenges **C#1**, by enabling the strong coupling of workflow components through the data plane, **C#2**, as data can be periodically and seamlessly published and retrieved to and from the data plane, and **C#3**, as new subscribing components can be dynamically added at runtime. Then, all the decisions about how and where to move or store data is fully delegated to data plane controller. This data management service should offer fast serial and parallel file-based I/O at all scales as well as the capacity to stream data from one workflow component to another over shared memory, and local and wide area networks. It should also be able to perform online data transformation to address the **C#4** and **C#7** challenges.

Embracing a service-oriented approach entails a small entry cost, as workflow components have to utilize the data management service API for data publication or subscription. However, it offers significant rewards in the long run, as the workflow becomes oblivious to optimization techniques and future changes in the underlying infrastructure. The service should allow the workflow management system to seamlessly switch from traditional file-based I/Os to data streaming, and to take advantage of the entire hierarchy of storage, from tapes to local NVMe, and of the most recent evolutions in storage techniques (e.g., object stores or computational storage) without any change to the application code.

Req. #2: Self-Describing Data

Traditional workflow management systems usually relies on files to transfer data from one component to another. Even when data is transferred over a communication network and not shared through a common file system, the same "file" abstraction is used. Each piece of data is a self-contained object, usually described by a name, and both the producer and consumer of that file have the knowledge of what this file contains and how to interpret the information in it. However, scientific data produced by scientific instruments or HPC simulations usually aggregate multiple objects structurally and temporally in a single hierarchical data container.

Enriching raw data with semantic descriptions of individual objects it encompasses (e.g., a multi-dimensional array capturing spatial distribution or temporal evolution of performance metrics) is essential for maximizing the efficiency of the data plane within a next-generation workflow and data ecosystem. However, the handling of metadata necessitates careful and efficient management, as uncontrolled costs can escalate significantly and consume more space than the actual data.

Rich metadata allows both the application plane and the control plane to easily understand, access, or query (very) specific pieces of information within a data set. For instance, it makes it possible to retrieve the evolution of a variable over a certain number of timesteps, which would be more complicated by accessing a set of files viewed as byte streams with only an offset and a size to navigate. It also allows users to add more annotations (e.g., units or description of the underlying mesh/topology) or extra statistics to the data so that it can be understood better and be leveraged by advanced data management systems to, aggregate data in a certain way (**C#5**), mitigate adversary access patterns (**C#7**), or program threshold-based mechanisms to trigger specific actions (**C#6**, **C#10**, and **C#11**).

Another important aspect related to self-describing data is that in next-generation workflows many different tools will have to operate on the same data (i.e., I/O libraries, data reduction algorithms, analytics and visualization software, AI/ML frameworks, and workflow management systems). This calls for data interoperability across workflow components. However, imposing a common data format would bring more harm than benefit as it would tightly link software components together. A more promising approach would be to define a common language under the form of the open standards of a schema that provides a clear separation between intent and implementation. This would allow each component to expose data in a way that can be understood by other components even though some transformation may be required to ensure the interoperability. Thanks to such a common schema, it becomes easier to dynamically add new components to a workflow (**C#3**) or perform in-transit data reorganization (**C#4**).

Req. #3: Adaptive Data Reduction

As mentioned in the introduction of this paper, one of the key motivations for driving next-generation workflows through the data plane stems from the increasing gap between the evolution of data volumes produced by scientific instruments and large-scale simulations and the computing and storage capacities of HPC centers.

If we consider that a computing software infrastructure is currently correctly sized to execute a workflow for a given amount of data and produce scientific results in a given time, an increase of the data production would mean that either it takes longer to obtain the results or the infrastructure has to be upgraded to deliver results under the same time constraint.

By integrating adaptive data reduction techniques (e.g., decimation, reduction of the precision, lossy or lossless compression) in the management of its data plane, a next-generation workflow and data ecosystem would dispose of an additional degree of freedom to take scheduling decisions under tight time constraints. Interesting trade-offs could then be investigated between time to solution and data quality. In many cases, it can be more interesting to obtain results in time, even though they are slightly less accurate, rather than risking to lose work because processing the raw data took too long. Pushed to the extreme, this approach would allow the workflow and data ecosystem to address **C#9** by allowing the near real-time part of a multi-stage analysis workflow to execute on highly reduced data.

Req. #4: Dynamic Control Plane

A recurring need that arises in the considered next-generation workflow motifs is that of a dynamic control of the execution of the workflow. The allocation of resources may have to be reconsidered as the workflow runs (**C#6**) depending on recommendations made by AI/ML models to steer the execution, because of the detection of an interesting phenomenon in the produced data that would require to spawn new analysis components (**C#3**), to retrain a diverging digital twin (**C#10**), or to perform command-and-control operations of the scientific instrument (**C#11**). All these dynamic use cases require to go beyond the static planning of the workflow execution that is commonly offered by traditional workflow managements and calls for event- or human-driven dynamic decision making processes. The control plane of a modern workflow ecosystem should, as for the data plane, follow a service-oriented approach, where new services could be triggered or stopped according to data-related events.

Another motivation for a service oriented control plane is the management of cross-facility workflows (**C#8**), which are increasingly prominent in computational sciences. These workflows span multiple sites, which could include experiment sites and various computing facilities (e.g., local compute resources, HPC centers, cloud infrastructure, edge computing resources, campus clusters, and edge sensors) [19]. In the context of cross-facility computing, it is crucial for the workflow ecosystem to be designed with the data plane in mind. This entails accommodating diverse representations and underlying storage systems, thereby ensuring flexibility and compatibility across different environments. It is also essential to be able to interact with the different computing services available at the different sites and to enable their composition and cooperation to create the proper execution environment.

V. Related Work

Several works in the literature offer partial solutions to the efficient management of next-generation workflows motifs.

The execution of the *Strong Code Coupling and Analysis* motif has been addressed in [20] by combining the respective features of a traditional workflow management system (Pegasus) and an in situ middleware (Decaf). The authors propose to improve inter-job communication by replacing a Pegasus sub-workflow, launched as a controller-worker MPI program, by a Decaf sub-workflow, run as a Multiple Program Multiple Data MPI job. Indeed, Pegasus uses MPI only to coordinate jobs that still exchange data through files while Decaf also rely on MPI for such data transfers. However, this approach is less flexible than fully delegating the control of the data plane, and thus the choice of the most efficient data transport method, to an efficient data management framework.

Stimulus [21] is a library designed to ingest scientific data into popular AI frameworks in an efficient and portable way. It addresses the problem of data format and access pattern incompatibility when HPC simulations and AI models are combined within a single workflow. Authors claim that a strongly coupled approach in which HPC simulations and AI models would share the same format is intractable and duplicates efforts. Stimulus thus unifies several popular scientific data formats under a single interface that hides the complexity of each, and proposes a generic data ingestion pipeline that can be executed by any tensor-operator-based framework.

The Vera C. Rubin Observatory's Legacy Survey of Space and Time project has proposed the concept of a Data Butler [22] whose objective is for the workflow developer to not have to know from where data is read or to where it is written, nor in which format. Each workflow component interacts with the Data Butler at the beginning and end of its execution to respectively ingest and export data. The Data Butler's implementation relies on a registry that organizes data sets conceptually, but ignores where data is located, and a datastore, that transforms the objects used in the workflow in the specific data format used by the target storage.

The Radical-Cybertools [23] are an example of the development of a complex workflow management system from a cohesive set of building blocks. The rationale is to support the agile development of workflow management systems. Each building block manipulates entities through a well-defined and stable interface that establishes a clear separation between computational and compositional features. Conversion layers are also offered to perform the translation of concepts from one representation to another. The Radical-Cybertools are composed of a toolkit to manage ensembles, a pilot job manager, and a homogeneous interface to batch schedulers.

TABLE I

SUMMARY OF THE CHALLENGES TO TRADITIONAL WORKFLOW MANAGEMENT SYSTEMS BY THE CONSIDERED NEXT-GENERATION WORKFLOW MOTIFS. FOR EACH CHALLENGE, WE INDICATE TO WHICH REQUIREMENT AND PLANE(S) THIS CHALLENGE PERTAINS, THE RISK TO NOT BEING ABLE TO ADDRESS THIS CHALLENGE AND THE POTENTIAL REWARD TO SOLVE IT USING SOFTWARE DEVELOPED AT OAK RIDGE NATIONAL LABORATORY.

Challenge	Requirement	Plane	Risk	Reward	Tool
C#1: Enable strong coupling without a single execution domain	#1	Data/Control	Low	High	ADIOS/EFFIS
C#2: Consume data produced by a workflow component as it runs	#1	Data/Control	Low	High	ADIOS/EFFIS
C#8: Coordinate cross-facility workflows	#4	Control	Medium	High	Zambeze
C#9: Adapt data reduction to time constraints	#3	Data/control	Medium	High	MGARD
C#3: Dynamically add new components to the workflow	#1, #2, and #4	Data/Control	Medium	High	ADIOS/EFFIS
C#10: Dynamically retrain digital twin	#2 and #4	Data/Control	High	High	EFFIS/Zambeze
C#11: Enable Command-and-Control of a scientific instrument	#2 and#4	Data/Control	High	High	ADIOS/EFFIS
C#4: Reorganize data in-transit to balance read/write performance	#1 and #2	Data	Medium	Medium	ADIOS
C#5: Consistently aggregate data within and across ensembles	#2	Data	Medium	Medium	ADIOS
C#7: Conciliate adversary data access patterns	#1 and #2	Data	Medium	Medium	ADIOS
C#6: Dynamically plan AI/ML-informed resource allocations	#2 and #4	Data/Control	High	Medium	EFFIS/Zambeze

In the Managing Event Oriented Workflows (MEOW) [24] framework, workflows are designed from the ground up to be dynamic. Users create individual components by defining *patterns*, that describe what events (e.g., accessing a specific file path, or getting an input file with a certain extension) should result in the processing, defined by *recipes*, of the input files and the production of output files. This original approach where events coming from the data plane trigger data processing is a good candidate to address some of the challenges raised by our workflow motifs, but the current implementation is limited to file system events.

Globus Flows [25] is an automation platform for complex data-intensive processes in distributed computing environments. Users can seamlessly create, execute, and manage workflows, orchestrate tasks, automate data transfers, and integrate with diverse computational resources and storage systems. The platform offers both a user-friendly graphical interface for designing workflows and a programmatic API.

The NERSC's SuperFacility API [26] enables researchers to access and utilize computational resources and data across multiple facilities. It simplifies the management of cross-facility workflows, offering streamlined resource allocation, job submission, data management, and workflow coordination. With the SuperFacility API, researchers can optimize their scientific workflows, collaborate effectively, and leverage the combined computing power of various facilities, fostering scientific discovery and efficient computational problem-solving.

The developers of the Maestro middleware [27] share the observation that data movement is of paramount importance in the exascale era and should be handled by dedicated software. They propose a memory- and data-aware middleware for data orchestration that allows applications to delegate data access and movement to Maestro. The framework is based on two main abstractions: Core Data Objects (CDO) that encapsulate data and their metadata, and a pool of CDOs to which workflow components can contribute or from which they can request CDOs. When an object is in the pool, Maestro has full control over it to move, copy, re-layout, or redistribute it.

VI. TOWARDS DRIVING NEXT-GENERATION WORKFLOWS FROM THE DATA PLANE

To tackle the challenges outlined in Section III and fulfill the requirements described in Section IV, our approach involves harnessing and integrating the capabilities of various tools developed at the Oak Ridge National Laboratory. By leveraging these tools, we aim to develop an integrated workflow and data ecosystem for the next-generation workflows.

Table I summarizes the challenges to traditional workflow management systems caused by the considered next-generation workflow motifs. For each challenge, we indicate to which requirement and plane(s) this challenge pertains, the risk to not being able to address this challenge and the potential reward to solve it, according to the current capabilities of the following software packages.

To manage the data plane, we rely on the ADIOS community-driven high-performance I/O framework [28]. ADIOS provides the necessary abstractions and a publish/subscribe API to allow applications to explicitly describe the data they produce, when it is ready for output, and what data an application needs to read and when. It is thus fully compliant with requirements #1 and #2. A key feature of ADIOS are its multiple engines that can either write directly to the storage system or stream data from the application to the memory of staging nodes or remote resources where it can be consumed by in situ analysis and visualization components. Thanks to this flexibility, these components can connect and disconnect to and from one another as the application runs by directly "taping in" the data plane without any further code modification. ADIOS has been successfully used in the Exascale Computing Project's WDMapp application and was able to achieve write performance over 5 TB/s on Frontier.

ADIOS also exposes the concept of *Operator* to define operations to be applied on ADIOS-managed data. One or many operators can be associated with any, or any group of, data objects and are executed by the selected ADIOS engine. In particular, ADIOS implements lossy and lossless data compression/decompression operators that can be used to

satisfy requirement #3 about adaptive data reduction. One of these operators is MGARD [29] which is a lossy compressor, which leverages multigrid structure, quantization, and lossless encoding. The main strength of MGARD is to achieve high compression ratios while providing error bounds for specific quantities of interest. MGARD has been successfully used to compress data produced by the XGC HPC simulation [9] and double the accuracy of cyclone detection in the E3SM model for about one fourth of the storage footprint.

To manage the control plane and make the connections between the application and data planes, we rely on two complementary tools. The Exascale Framework for High Fidelity coupled Simulations (EFFIS) [8] enables the arbitrary composition of multiple applications, providing users with a uniform syntax to describe their workflows. EFFIS also has the capacity to dispatch data to remote sites, such as a local cluster or laptop, for additional analysis and has been designed to integrate with checkpoint-restart to improve the resilience of the workflow components. One of the key feature of EFFIS is to work in conjunction with ADIOS to both organize and optimize the I/O for large scale runs. Thanks to the combined strengths of EFFIS and ADIOS, it has been possible to couple multiple analysis and visualization components to the XGC fusion code to uncover new physics phenomena [30].

The second tool, called Zambeze [31], is an innovative distributed orchestration framework designed to facilitate experiment campaigns across the edge-to-HPC continuum. It caters to scenarios that demand advanced network capabilities for large data movement, along with processing and storage resources. Zambeze framework seamlessly integrates diverse services, including workflow and data management systems, using distributed agents deployed on each facility. These agents efficiently translate user requests into executable actions on the computational platform. Adopting an adapter system design, Zambeze is adaptable and versatile, making it suitable for different architectures and platform requirements. It excels at orchestrating various workflow systems and facilitating data movement while remaining domain-agnostic.

VII. Open Research Challenges

While the tools presented in the previous section addresses the different challenges caused to traditional workflow management systems by the considered next-generation workflow motifs, some additional open research challenges remain.

ORC #1: On demand Remote and Local Data Access

The proposition to store the data produced by a workflow in a data-plane along with a rich description of its contents and relying on a data management framework that can understand and exploit this description open the way for users to the on-demand remote and local access to scientific data. The motivation is a common practice for scientist that consists in performing a first data analysis on their laptop, which is hardly possible with data sets in the range of the Terabytes or the Petabytes. This is also true for academic institutions which have some computing an storage resources available on premises to analyse data, but not at the same scale as leadership class facilities. In this context, scientists face an interesting problem, which is related to one of the FAIR principles: Accessibility. While huge amounts of data may be accessible at a HPC center, analysing them may not be possible by lack of sufficient local resources. However, depending on the analysis, visualization, or, more generally, processing that scientists want to perform at a given moment, it is not necessary to access the entire raw data set at the best accuracy level available. Instead, scientist may want to access a very specific subset of the entire data set or be satisfied with a much lower accuracy. The objective is then to offer the same kind of tools as those available to person accessing photos stored in the cloud, looking for a picture at a specific event. They will first narrow down the scope (e.g., by date or album) and then browse low-resolution thumbnails to select only group pictures. This vastly reduced set of selected pictures can be downloaded at full resolution to complete the search. Once such on-demand remote access is available, interesting research challenges related to caching and indexing the locally stored data will have to be addressed.

ORC #2: Visualization as a Service

Scientific data visualization is a key component of the decision processes at the core of several of the considered next-generation motifs. However, the current state-of-the-art in that field usually requires calls to the visualization software to be made directly from the application code using bespoke APIs. The Visualization as a Service [32] approach gives ability to break visualization and analysis tasks into pieces that can be deployed, managed, automated by a workflow management system, and be driven from the data plane. The main research challenge associated to this approach is thus to efficiently and dynamically manage the resource allocation of the visualization services in the control plane and to find the best trade-offs between the resource needs of the application and the additional insight brought by visualization.

ORC #3: Resilient Workflow Execution

The resilient execution of scalable cross-facility workflows must be ensured to guarantee the production of new scientific results. However, the dramatic growth in data production poses challenges in terms of replicating all data and computations, as traditionally practiced. To address this, a promising approach is to harness data reduction techniques and leverage the data plane's capability to generate and store multiple versions of a data set with varying levels of accuracy. This would enable the workflow and data ecosystem to execute less accurate replicas of a workflow, which consume fewer resources and execute faster. These replicas could also serve as fallback options in the event of failure during the execution of the original workflow.

ORC #4: Privacy-Preserving Workflows

The *Edge-to-HPC Multi-Stage Analysis* motif manifests not only in traditional scientific workflows but also in federated learning applications. In that context, an interesting privacy-preserving challenge arises that consists in ensuring the confi-

dentiality of individual contributions across the edge-to-HPC continuum during collaborative AI/ML processes. It implies to allow multiple parties to collaborate on training a model while ensuring the local security and privacy of their individual data sets. One promising approach is to employ techniques such as encryption or differential privacy to exchange model updates or gradients among participants, rather than sharing raw data [33]. Investigating such privacy-preserving mechanisms will play a crucial role in enabling secure and privacy-conscious collaborative model improvement.

ORC #5: Provenance Data Capture

With workflows spanning across multiple facilities and the ability to dynamically adjust their structure based on AI/ML-driven steering recommendations, the capture of provenance data is key to ensuring result reproducibility. It becomes even more important with the increasing role of AI/ML for which tracing, understanding, and explaining how data has been transformed to feed the models is mandatory to ensure the transparency and explainability of their results. It is for instance important to keep track of the complete training process, with information about the including parameters, model architecture, data sets, and data transformations.

One of the primary performance challenges associated with provenance data capture lies in striking the right balance between capturing sufficient metadata to be valuable and avoiding overwhelming the system with excessive information. It also amounts to finding the best trade-off between the time and space devoted to scientific data and to provenance data. As for visualization, the right balance between the insight brought by provenance data capture and the additional resources it requires has to be found.

These challenges related to provenance data capture also pertain to the capture of the metadata that can be associated to scientific data to enrich their semantic description. This additional information has to be extracted from the workflow components at the same pace as that of scientific data production, which call for a unified use of the publish/subscribe paradigm and the delegation of delegating data movement and storage to a high-performance data management. It also means that this high-performance data management framework must take advantage of the efficient mechanisms used by provenance and metadata management (i.e., key-value stores, databases, efficient indexing and querying methods) to handle this additional wealth of data.

ORC #6: Performance Assessment

The efficient execution of next-generation workflows will depend on the quality of the decision-making strategies implemented by the workflow ecosystem. These strategies have to be informed by adaptive performance models of both the workflow and the target computing and storage infrastructure. These models can be used in a proactive way to determine the initial deployment of a workflow given the resources available at schedule time, but also in a more reactive way, at runtime, to be able to react to dynamic events and make appropriate decisions. To design such performance models, we must overcome challenges related to the complexity of the considered next-generation workflows in the edge-to-HPC continuum. Performance models have long been pursued in HPC; yet, developing tractable, practical, and accurate models for (nontrivial) use cases remains challenging. This is because capturing complex interactions between activities (i.e., computation, communication, I/O) that use hardware resources concurrently renders the modeling problem highly combinatorial. Thus, most existing performance models make simplifying assumptions that do not hold in practice, causing a theory/practice disconnect. An interesting alternative is to model performance via simulation. Instead of pursuing mathematical models, simulators (i.e., software artifacts) that perform discrete-event simulation of a workflow's execution on a computing and storage infrastructure can be developed. The simulator outputs a time stamped trace of (simulated) workflow execution events, from which performance metrics can be easily computed as they arise naturally from the simulated events. This is the approach followed by the WRENCH cyberinfrastructure and workflow simulation workbench [34].

VIII. Conclusion

Scientific workflows are undergoing a significant transformation from the conventional "move little data, compute for long" model, where computational aspects take precedence, to a more modern "move a lot, compute for cheap" model. This evolution is driven by the increasing volumes of data being generated and the need to efficiently utilize available computational resources. In this shifting paradigm, optimizing data access, transfer, and storage has emerged as the primary focus to enable efficient and cost-effective scientific computations.

In this paper, we provide concrete examples to illustrate this paradigm shift by describing five distinct workflow motifs originating from various scientific domains. These motifs demonstrate the changing landscape where the effective management of data becomes a critical optimization challenge in contemporary scientific workflows. We then identified four requirements we think a modern workflow and data ecosystem should meet and detailed how they are met by software developed at the Oak Ridge National Laboratory. Finally, through six open research challenges, we showed how the considered motifs highlight the need for strategies that balance computational requirements with efficient data management across distributed computing environments.

Acknowledgements

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. This research is partially supported by Laboratory Directed Research and Development Strategic Hire funding No. 11134 from Oak Ridge National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy.

References

[1] R. Badia, E. Ayguadé, and J. Labarta, "Workflows for Science: A Challenge When Facing the Convergence of HPC and Big Data," *Supercomputing Frontiers and Innovations*, vol. 4, no. 1, pp. 27–47, 2017.

[2] R. Ferreira da Silva, R. Badia, V. Bala, D. Bard, T. Bremer, I. Buckley, S. Caino-Lores, K. Chard, C. Goble, S. Jha, D. S. Katz, D. Laney, M. Parashar, F. Suter, N. Tyler, T. Uram, I. Altintas *et al.*, "Workflows Community Summit 2022: A Roadmap Revolution," Oak Ridge National Laboratory, Tech. Rep. ORNL/TM-2023/2885, 2023.

[3] P. Amstutz, M. Mikheev, M. R. Crusoe, N. Tijanić, S. Lampa *et al.*, "Existing Workflow systems," [Online] https://s.apache.org/existing-workflow-systems, updated 2022-09-13, accessed 2023-01-23.

[4] J. Demmel, L. Grigori, M. Hoemmen, and J. Langou, "Communication-optimal Parallel and Sequential QR and LU Factorizations," *SIAM Journal on Scientific Computing*, vol. 34, no. 1, pp. A206–A239, 2012.

[5] H. Luu, M. Winslett, W. Gropp, R. Ross, P. Carns, K. Harms, M. Prabhat, S. Byna, and Y. Yao, "A Multiplatform Study of I/O Behavior on Petascale Supercomputers," in *Proc. of the 24th International Symposium on High-Performance Parallel and Distributed Computing*, 2015, p. 33–44.

[6] R. Ferreira da Silva, R. Filgueira, I. Pietri, M. Jiang, R. Sakellariou, and E. Deelman, "A Characterization of Workflow Management Systems for Extreme-Scale Applications," *Future Generation Computer Systems*, vol. 75, pp. 228–238, 2017.

[7] M. R. Crusoe, S. Abeln, A. Iosup, P. Amstutz, J. Chilton, N. Tijanić, H. Ménager, S. Soiland-Reyes, B. Gavrilović, C. Goble, and T. C. Community, "Methods Included: Standardizing Computational Reuse and Portability with the Common Workflow Language," *Communications of the ACM*, vol. 65, no. 6, p. 54–63, May 2022.

[8] E. Suchyta, S. Klasky, N. Podhorszki, M. Wolf, A. Adesoji, C. S. Chang, J. Choi, P. Davis, J. Dominski, S. Ethier, I. Foster, K. Germaschewski, B. Geveci, C. Harris, K. Huck, Q. Liu, J. Logan, K. Mehta, G. Merlo, S. Moore, T. Munson, M. Parashar, D. Pugmire, M. Shephard, C. Smith, P. Subedi, L. Wan, R. Wang, and S. Zhang, "The Exascale Framework for High Fidelity coupled Simulations (EFFIS): Enabling Whole Device Modeling in Fusion Science," *The International Journal of High Performance Computing Applications*, vol. 36, no. 1, pp. 106–128, 2022.

[9] S. Ku, C. S. Chang, and P. H. Diamond, "Full-f Gyrokinetic Particle Simulation of Centrally Heated Global ITG Turbulence from Magnetic Axis to Edge Pedestal Top in a Realistic Tokamak Geometry," *Nuclear Fusion*, vol. 49, no. 11, p. 115021, Sep. 2009.

[10] K. Germaschewski, B. Allen, T. Dannert, M. Hrywniak, J. Donaghy, G. Merlo, S. Ethier, E. D'Azevedo, F. Jenko, and A. Bhattacharjee, "Toward Exascale Whole-Device Modeling of Fusion Devices: Porting the GENE Gyrokinetic Microturbulence Code to GPU," *Physics of Plasmas*, vol. 28, no. 6, p. 062501, 2021.

[11] L. R. Leung, D. Bader, M. Taylor, and R. McCoy, "An Introduction to the E3SM Special Collection: Goals, Science Drivers, Development, and Analysis," *Journal of Advances in Modeling Earth Systems*, vol. 12, no. 11, p. e2019MS001821, 2020.

[12] V. Balasubramanian, M. Turilli, W. Hu, M. Lefebvre, W. Lei, R. Modrak, G. Cervone, J. Tromp, and S. Jha, "Harnessing the Power of Many: Extensible Toolkit for Scalable Ensemble Applications," in *Proc. of the IEEE International Parallel and Distributed Processing Symposium*, 2018, pp. 536–545.

[13] S. R. Johnson, A. Lund, S. Y. Jun, S. Tognini, G. Lima, P. Romano, P. Canal, B. Morgan, T. Evans, V. R. Pascuzzi, and D. Deeb, "Celeritas," [Computer Software] https://doi.org/10.11578/dc.20221011.1, Jul. 2022.

[14] R. van Liere, J. Mulder, and J. van Wijk, "Computational Steering," *Future Generation Computer Systems*, vol. 12, no. 5, pp. 441–450, 1997.

[15] H. Lee, M. Turilli, S. Jha, D. Bhowmik, H. Ma, and A. Ramanathan, "DeepDriveMD: Deep-Learning Driven Adaptive Molecular Simulations for Protein Folding," in *Proc. of the IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS)*, nov 2019, pp. 12–19.

[16] T. Mason, T. Gabriel, R. Crawford, K. Herwig, F. Klose, and J. Ankner, "The Spallation Neutron Source: A Powerful Tool for Materials Research." [Online]. Available: https://arxiv.org/abs/physics/0007068

[17] K. Gillette, M. Gsell, A. Prassl, E. Karabelas, U. Reiter *et al.*, "A Framework for the Generation of Digital Twins of Cardiac Electrophysiology from Clinical 12-leads ECGs," *Medical Image Analysis*, vol. 71, p. 102080, 2021.

[18] F. Di Natale, H. Bhatia, T. S. Carpenter, C. Neale, S. Kokkila-Schumacher *et al.*, "A Massively Parallel Infrastructure for Adaptive Multiscale Simulations: Modeling RAS Initiation Pathway for Cancer," in *Proc. of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019.

[19] K. Antypas, D. Bard, J. Blaschke, R. S. Canon, B. Enders, M. Shankar, S. Somnath, D. Stansberry, T. Uram, and S. Wilkinson, "Enabling Discovery Data Science Through Cross-Facility Workflows," in *Proc. of IEEE International Conference on Big Data*, 2021, pp. 3671–3680.

[20] T. M. A. Do, L. Pottier, O. Yildiz, K. Vahi, P. Krawczuk, T. Peterka, and E. Deelman, "Accelerating Scientific Workflows on HPC Platforms with In Situ Processing," in *Proc. of the 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing*, 2022, pp. 1–10.

[21] H. Devarajan, A. Kougkas, H. Zheng, V. Vishwanath, and X.-H. Sun, "Stimulus: Accelerate Data Management for Scientific AI applications in HPC," in *Proc. of the 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing*, 2022, pp. 109–118.

[22] T. Jenness, J. Bosch, N. Lust, N. Pease, M. Gower, M. Kowalik, G. Dubois-Felsmann, F. Mueller, and P. Schellart, "The Vera C. Rubin Observatory Data Butler and Pipeline Execution System," 2022. [Online]. Available: https://arxiv.org/abs/2206.14941

[23] M. Turilli, V. Balasubramanian, A. Merzky, I. Paraskevakos, and S. Jha, "Middleware Building Blocks for Workflow Systems," *Computing in Science & Engineering*, vol. 21, no. 4, pp. 62–75, 2019.

[24] D. Marchant, R. Munk, E. O. Brenne, and B. Vinter, "Managing Event Oriented Workflows," in *Proc. of the 2nd IEEE/ACM Annual Workshop on Extreme-scale Experiment-in-the-Loop Computing*, 2020, pp. 23–28.

[25] R. Chard, J. Pruyne, K. McKee, J. Bryan, B. Raumann, R. Ananthakrishnan, K. Chard, and I. Foster, "Globus Automation Services: Research Process Automation Across the Space–Time Continuum," *Future Generation Computer Systems*, 2023.

[26] D. Bard, M. Day, B. Enders, R. Hartman-Baker, J. Riney III, C. Snavely, and G. Torok, "Automation for Data-driven Research with the NERSC Superfacility API," in *High Performance Computing: ISC High Performance Digital International Workshops*, Jun. 2021, pp. 333–345.

[27] C. Haine, U.-U. Haus, M. Martinasso, D. Pleiter, F. Tessier, D. Sarmany, S. Smart, T. Quintino, and A. Tate, "A Middleware Supporting Data Movement in Complex and Software-Defined Storage and Memory Architectures," in *Proc. of the International Conference on High Performance Computing*, 2021, pp. 346–357.

[28] W. F. Godoy, N. Podhorszki, R. Wang, C. Atkins, G. Eisenhauer, J. Gu, P. Davis, J. Choi, K. Germaschewski, K. Huck, A. Huebl, M. Kim, J. Kress, T. Kurc, Q. Liu, J. Logan, K. Mehta, G. Ostrouchov, M. Parashar, F. Poeschel, D. Pugmire, E. Suchyta, K. Takahashi, N. Thompson, S. Tsutsumi, L. Wan, M. Wolf, K. Wu, and S. Klasky, "ADIOS 2: The Adaptable Input Output System. A framework for high-performance data management," *SoftwareX*, vol. 12, p. 100561, 2020.

[29] X. Liang, B. Whitney, J. Chen, L. Wan, Q. Liu, D. Tao, J. Kress, D. Pugmire, M. Wolf, N. Podhorszki, and S. Klasky, "MGARD+: Optimizing Multilevel Methods for Error-Bounded Scientific Data Reduction," *IEEE Transactions on Computers*, vol. 71, no. 7, pp. 1522–1536, 2022.

[30] E. Suchyta, J. Y. Choi, S.-H. Ku, D. Pugmire, A. Gainaru, K. Huck, R. Kube, A. Scheinberg, F. Suter, C.-S. Chang, T. Munson, N. Podhorszki, and S. Klasky, "Hybrid Analysis of Fusion Data for Online Understanding of Complex Science on Extreme Scale Computers," in *Proc. of the IEEE International Conference on Cluster Computing*, 2022, pp. 218–229.

[31] T. J. Skluzacek, R. Souza, K. C. Maheshwari, S. R. Wilkinson, and R. Ferreira da Silva, "Zambeze: Enabling Cross-Facility Workflows Orchestration," [Online] https://github.com/ORNL/zambeze, accessed 2023-07-27.

[32] D. Pugmire, J. Kress, J. Chen, H. Childs, J. Choi, D. Ganyushin, B. Geveci, M. Kim, S. Klasky, X. Liang, J. Logan, N. Marsaglia, K. Mehta, N. Podhorszki, C. Ross, E. Suchyta, N. Thompson, S. Walton, L. Wan, and M. Wolf, "Visualization as a Service for Scientific Data," in *Proc. of the Smoky Mountains Computational Sciences and Engineering Conference*, 2020, pp. 157–174.

[33] A. Clyde, X. Liu, T. Brettin, H. Yoo, A. Partin, Y. Babuji, B. Blaiszik, J. Mohd-Yusof, A. Merzky, M. Turilli *et al.*, "AI-Accelerated Protein-Ligand Docking for SARS-CoV-2 is 100-fold Faster with no Significant Change in Detection," *Scientific Reports*, vol. 13, no. 1, p. 2105, 2023.

[34] H. Casanova, R. Ferreira da Silva, R. Tanaka, S. Pandey, G. Jethwani, W. Koch, S. Albrecht, J. Oeth, and F. Suter, "Developing Accurate and Scalable Simulators of Production Workflow Management Systems with WRENCH," *Future Generation Computer Systems*, vol. 112, pp. 162–175, 2020.