Towards Resilient Near Real-Time Analysis Workflows in Fusion Energy Science

Frédéric Suter Norbert Podhorszki Scott Klasky
Oak Ridge National Laboratory, Oak Ridge, TN, USA
{suterf, pnorbert, klasky}@ornl.gov

***Abstract*—Nuclear fusion holds the promise of an endless source of energy. Several research experiments across the world and joint modeling and simulation efforts between the nuclear physics and high performance computing communities are actively preparing the operation of the International Thermonuclear Experimental Reactor (ITER). Both experimental reactors and their simulated counterparts generate data that must be analyzed quickly and in a resilient way to support decision making for the configuration of subsequent runs or prevent a catastrophic failure. However, the cost if the traditional techniques used to improve the resilience of analysis workflows, i.e., replicating datasets and computational tasks, becomes prohibitive with explosion of the volume of data produced by modern instruments and simulations. Therefore, we advocate in this paper for an alternate approach based on data reduction and data streaming. The rationale is that by allowing for a reasonable, controlled, and guaranteed loss of accuracy it becomes possible to transfer smaller amounts of data, shorten the execution time of analysis workflows, and lower the cost of replication to increase resilience. We develop our research and development roadmap towards resilient near real-time analysis workflows in fusion energy science and present early results showing that data streaming and data reduction is a promising way to speed up the execution and improve the resilience of analysis workflows.**

I. Introduction

Nuclear fusion holds the promise of an endless source of energy. Being able to replicate on Earth the processes that power the Sun and stars could address the ever-increasing energy needs of our modern world [1]. One of the leading design to implement a practical fusion reactor is the Tokamak design which consists in confining a plasma in the shape of an axially-symmetrical torus thanks to a powerful magnetic field. Several research tokamak experiments across the world (e.g, Korea Superconducting Tokamak Advanced Research (K-STAR) in South Korea, the Mega Ampere Spherical Tokamak (MAST) in the United Kingdom, or DIII-D in the United States) are actively preparing the operation of the International Thermonuclear Experimental Reactor (ITER). Moreover, joint efforts between the nuclear physics and high performance computing (HPC) communities on the modeling and simulation of tokamaks have led to the development of multiple codes, such as the Exascale Computing Project's (ECP) Whole Device Modeling application (WDMapp) that allow scientists to better understand the evolution of the plasma and the confinement chamber during a 'shot' [2].

Both experimental reactors and their simulated counterparts are equipped with multiple probes and diagnostics to monitor the state of the plasma and of the tokamak itself. The generated data is then processed by complex in situ analysis and visualization workflows [3]. We identified two scenarios in which the time to obtain the results of such analyses is critical to allow for timely decision making. First, operation parameters or simulation inputs can be reconfigured from one shot to another, based on the analysis of the data collected during the former shot, to optimize the behavior of the plasma or address detected issues. Second, analysis monitoring data during a shot can allow for the early detection of precursors of a catastrophic event, and give enough information and time to operators to take the decision of stopping the reactor. The processing of these analysis workflows not only has to be done in *near real-time*, but also must be *resilient*. Being unable to get results in time, because of unexpected delays or resource failures, may either delay the start of the next shot in the first scenario or lead to a catastrophic failure in the second one.

These two scenarios are an illustration of a broader need for the support of *time-sensitive workflows* in fusion energy science (FES) and other scientific domains. A recent meta-analysis [4] of 74 high-priority case studies from the US Department of Energy (DOE) Office of Science Programs identified that 45% of the workflows related to these case studies exhibit a time-sensitive pattern. In FES alone, half of case studies express this need as "*FES experimentation is by nature time dependent, since a single 'shot' of a reactor produces a burst of data that must be quickly analyzed, with the output of the analysis supporting decision making for the configuration of the next shot*". Near real-time analysis workflows will thus be one of the major category of workflows to be supported by the forthcoming DOE Integrated Research Infrastructure [5].

To make a near real-time workflow, or any scientific application, more resilient to delays and failures, one of the most common technique in the literature is to replicate and distribute both datasets and (parts of) the analysis workflow on different compute clusters (i.e., to survive to failures) [6]–[9]. However, the number of replicas that can be deployed to improve resilience is limited by resource availability, budgetary

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).

constraints, or energy-efficiency concerns, and this approach simply becomes prohibitively expensive with explosion of the volume of data produced by modern instruments and simulations. Therefore, we advocate for an alternate approach based on data reduction and data streaming. The rationale is that by allowing a for a reasonable, controlled, and guaranteed loss of accuracy in the data produced by an fusion reactor or an HPC simulation, it becomes possible to: i) transfer smaller amounts of data from the data source to the analysis resources; ii) shorten the execution time of analysis workflows by processing smaller datasets; and iii) lower the cost of replicating data and computations to increase resilience.

In this paper, after having briefly described the structure of near real-time analysis workflows in fusion energy science in Section II, we develop our research and development roadmap to make these workflow resilient in Section III. The different steps of this roadmap are to:

- Rely on a high-performance I/O and data management framework to reduce and stream the data produced by (simulations of) fusion reactors (Section III-A);
- Generate multiple reduced data streams to both increase the resilience of the analysis workflows and meet their near real-time constraints (Section III-B);
- Solve a multi-criteria optimization problem and design resource allocation and scheduling algorithms to guarantee the respect of the time constraint imposed on the execution of the analysis while offering a certain amount of resilience at a controlled additional cost (Section III-C);
- Design appropriate performance evaluation tools to assess the quality and robustness of the designed algorithm over a broad range of experimental scenarios (Section III-D).

Finally we review related work in Section IV before summarizing our approach in Section V.

II. Near Real-Time Analysis in Fusion Science

Fig. 1 illustrates the different phases of a generic near real-time analysis workflow in FES. First, the instrument, or its simulated counterpart, produces data that need to be *acquired* before being processed. In the specific case of nuclear fusion reactors, this corresponds to dozens of diagnostics (e.g., cameras, spectrometers, bolometers, monitoring sensors, and probes) and uses various data formats. Commonplace approaches for data acquisition are to write the generated data into files or to create a data stream. The second step is to *transport* the acquired data to computing resources dedicated to analysis. These resources constitute the destination endpoint of a data stream, while, in the case of a file-based transport, it implies moving the files over the network and read them at destination. The *analysis* step can be a simple program (e.g., computing derived quantities of interest or an histogram) or a more complex workflow (e.g., combining a Poincaré puncture plot to heat load and diffusion calculations [3]). This analysis usually comprises a *detection* mechanism (e.g., going over a given threshold) that, if triggered, sends a *feedback* to operators. Getting this feedback in a timely fashion allows the operators to take informed decision about the next action to

Fig. 1. Phases of a generic near real-time workflow in Fusion Energy Science.

perform (e.g., reconfigure the instrument for the next shot or shutdown the reactor to prevent a catastrophic event).

The Experiment-to-Experiment cycle illustrated by Fig. 1 is generally executed periodically over the lifespan of the operation of a fusion reactor. The main difference between the two considered near real-time scenarios lies in Δ_t, the length of the period between two iterations of this cycle. The time operators have between two shots of a fusion reactor to perform a post-mortem analysis of the produced data and proceed with the reconfiguration of the reactor is of the order of an hour. However, to be able to detect precursors of a catastrophic event, this cycle must be repeated on the order of every few seconds.

Different techniques can be used to ensure that operators and scientists can obtain the needed analysis results in near real-time. Data produced by sensors and probes usually go through several filtering steps to reduce the volume of information to analyze at each step and focus on the most relevant information. These filters can be implemented directly in hardware using custom-made electronics or use algorithms whose complexity and execution time increase from one filtering step to another. However, we believe that filtering out data and simplifying the analysis algorithms may lead to losing more information than opting for a reduction-based approach that trades some accuracy to respect near real-time constraints.

III. Towards Resilient Near-Real Time Workflows

In this section we detail the research and development we deem necessary to make near real-time analysis workflows in fusion energy science resilient. Our roadmap includes three main steps in which we plan to: (i) extend the structure of the analysis workflow to leverage data reduction techniques and create redundant analyses; (ii) design resource allocation and scheduling algorithms to offer more resilience at an affordable cost; and (iii) evaluate the quality of the designed algorithms for the two considered use case scenarios under various experimental conditions.

A. Enabling Data Streaming and Data Reduction

A essential preliminary step towards resilient near real-time analysis workflows is to be able to *stream* and *reduce* the data produced by the fusion experiment. Following recommendations made in [10], which identified near real-time analysis workflows as a next-generation workflow motif, we propose to decouple data management, reduction, transport, and storage from the workflow components that produce or consume data.

To this end, we base our work on two software frameworks developed at Oak Ridge National Laboratory (ORNL). To enable data streaming from a fusion experiment to computing resources dedicated to data analysis, we rely on the ADIOS community-driven high-performance I/O framework [11]. ADIOS exposes a publish/subscribe API to allow a data generator to explicitly describe the produced data and when it is ready for output, and for an analysis workflow to express what data it needs to read and when. 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, an analysis workflow can directly "tap into" data without disrupting the data generator.

ADIOS also exposes the concept of *Operator* to define operations to be applied on ADIOS-managed data. In particular, ADIOS implements lossy (e.g., MGARD [12], SZ [13], or ZFP [14]) and lossless (e.g., bzip2) data compression/decompression operators, and data refactoring operators (e.g., MGARD-DR or SIRIUS [15]).

ADIOS and MGARD have been integrated into the ECP's WDMapp codes to achieve write performance over 5 TB/s on Frontier and drastically reduce the volume of produced data. Thanks to ADIOS streaming capacities, the time to transfer data produced by the KSTAR experiment in South Korea to NERSC in California has been reduced from 12 hours to 10 minutes [16]. More recently, data from the MAST experiment have been made available in the ADIOS file format.

B. Generating Reduced Data Streams

The second step on our roadmap corresponds to the core of our approach to make analysis workflows in FES resilient and ensure that they are executed in near real-time. It consists in producing additional data streams once data has be acquired from the instrument or HPC simulation and applying data reduction techniques to these additional data streams. Fig. 2 illustrates the addition of such a reduced data stream to the generic analysis workflow introduced in Fig. 1.

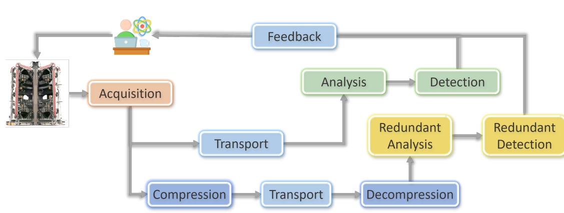

Fig. 2. Near real-time workflow in Fusion Energy Science with an additional reduced data stream to enable redundant analysis and detection.

The second stream depicted at the bottom of Fig. 2 comprises two additional steps. First, data has to be reduced, using techniques ranging from a simple decimation (i.e., only retaining one data point for every N data points) to advanced lossy compressors [12]–[14], through frequency space techniques such as wavelet transform or fast Fourier transform for images. Second, once data has been transported to its destination, it may, depending on the chosen reduction technique, have to be decompressed or reconstructed before being analysed.

While these extra steps to compress/refactor and decompress/reconstruct data may add some compute time to the analysis process, they also bring several benefits that justify the additional cost. First, reducing data mechanically reduces the time needed to transfer them from the instrument to where the analysis is performed. To illustrate the potential gain on transfer time brought by data reduction, we transferred actual data produced by the MAST experiment from two different location (i.e., the original MAST data repository in Cambridge, United Kingdom, and a copy stored on the file system of the Frontier supercomputer at ORNL) to a laptop also located at ORNL. Fig. 3 shows the measured data transfer times for these two scenarios when scaling the resolution of the original data up and down. Note that the considered MAST dataset is composed of seven 3D signals. Then, scaling down (resp up.) the resolution by a factor of two reduces (resp. increases) the data size by a factor of eight. The dataset to transfer represents 8.6 MB at its original resolution.

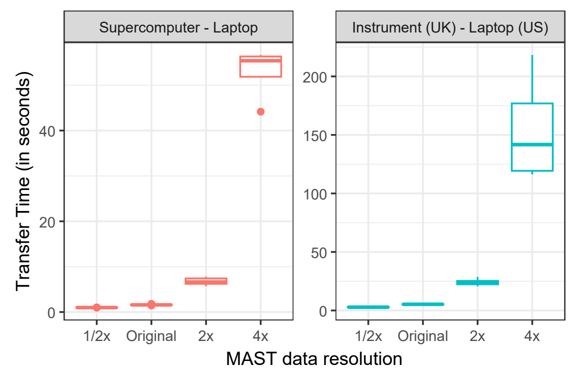

Fig. 3. Evolution of the time to transfer data acquired by MAST, from either the experiment site in the United Kingdom or a supercomputer in the United States to a laptop in the United States, when the data resolution changes.

We can see that reducing the size of MAST data by halving its original resolution also reduces the transfer time by a factor of two. As the dataset is rather small, the achievable gain is thus fairly limited but still significant with a near real-time objective in mind. However, the scaled up versions of the original dataset (i.e., with doubled and quadrupled resolution) are more representative of the volume and nature of data the MAST experiment will produce after its upgrade. We observe that with increased data sizes, the transfer time becomes prohibitive for near real-time analyses. Moreover, the transfer of the dataset whose resolution has been scaled up four times exhibits high variability which would lead to uncertainty for the operators in their capacity to get results in time.

We hypothesize that while the data produced by MAST, once fully upgraded, will have a size of the order of the '4x' scenario in Fig. 3 (i.e, about half a gigabyte per shot), the data produced by ITER will be about a hundred thousand times larger. At that scale, performing near real-time analyses on the raw dataset at full accuracy becomes hardly possible. These preliminary measurements show that data reduction can allow redundant analyses to start much earlier and thus make the distribution of reduced replicas of the original dataset over geographically distributed computing sites a viable option to increase the resilience of the analysis workflows.

A second advantage of the generation of reduced data streams is that the time to perform the analyses themselves can also be reduced. This requires that once decompressed or reconstructed, the data size remains smaller than the original size. Reduction techniques such as lossy compression do not satisfy this requirement as decompressed data usually have a size commensurate to that of the original dataset. Conversely, simple techniques such as decimation would reduce the size of the data to be analyzed, but without any guarantee on the accuracy of the reduced data. We thus propose to rely on more advanced refactoring techniques [17] in which data is decomposed and refactored into a hierarchical reduced representation. The upper levels of this hierarchy would be made of lower accuracy, but much smaller in size, versions of the original data, with both accuracy and size increase as we progress towards the lower levels of the hierarchy. The key feature here is to guarantee a certain error, or loss of accuracy, for each of the refactoring levels. This way, it becomes possible to select how accuracy is traded for the reduction of the data size, hence for the acceleration of both transfer time and analysis time.

Finally, generating multiple reduced data streams and analyzing them also mechanically improves resilience by replicating the entire analysis process. Note that these replicas working on less accurate data, the confidence the operator has in the analysis results may be lower. However, using the right reduction techniques, the introduced errors can be bounded. Moreover, to regain some of the lost confidence, the results obtained with the different replicas can be aggregated, provided they converge on the detection of the same phenomenon.

C. Multi-Objective Optimization and Algorithm Design

Our objective to enable resilient near real-time analysis workflows for fusion energy science can translate to the following multi-objective optimization problem. Given an analysis workflow W processing a dataset D, we want to determine a resilient execution scheme that satisfies:

- A time constraint ΔT before which the results of the analysis have to be sent back to operators;
- A minimum resilience factor R, meaning that the analysis can be completed despite the occurrence of R failures (or delays preventing the respect of the time constraint);
- A maximum resource budget B allocated to resilience.

To this end, we generate R reduced data streams $D^j_{e_i}$ with $1 \leq j \leq R$ and e_i, an error after reduction ($i \in \mathbb{R}^+$). We denote by $D^0_{e_0}$, the original dataset and by $T(W, D^0_{e_0}, C_0)$ the time the complete the analysis workflow while operating on the original dataset. This execution time includes the time to transfer $D^0_{e_0}$ from the data production site P_0 to C_0, a primary set of computing resources dedicated to analysis.

Each of the reduced data streams can be analysed on a set of computing resources C_k leading to an execution time of $T(W, D^j_{e_i}, C_k)$. Here, this execution time not only includes the time to transfer data from P_0 to C_k but also the time to compress/refactor and decompress/reconstruct data at destination. Finally, $cost_k$ represents the cost (in node.hours or in a given currency) to use resources C_k.

Our objective is then to design an algorithm that decides of: (i) the number of reduced data streams to generate; (ii) the accepted error for each of these streams; and (iii) the set of computing resources to use for the analysis of each stream, in order to maximize R, such that:

$$\exists i \in \mathbb{R}^+, 0 \leq j \leq R, k \geq 0 \; s.t. \; max\left(T(W, D^j_{e_i}, C_k)\right) \leq \Delta t,$$

i.e., the primary analysis workflow or at least of one its replicas completes in less than the time constraint, and:

$$\sum_{k>0} cost_k \leq B,$$

i.e., the sum of the costs associated to the execution of the workflow replicas to ensure its resilience remains under the maximum budget allowed.

A key parameter in the design of such a resource allocation and scheduling algorithm is e_i, the error associated to a reduced data stream. We will consider different scenarios in this part of our roadmap. We will start with a case where no bound on the error can be guaranteed, for instance when using simple techniques such as decimation, to focus on the capacity to drastically reduce data size in order to meet the time, resilience, and budget constraints. Then, we will consider a scenario in which there exists a certain error threshold that guarantees that the analysis results can be exploited for decision making. Finally, we will consider a more realistic scenario in which the confidence in the analysis results is inversely proportional to the tolerated error (i.e., the smallest the error, the highest the confidence). This scenario will add another optimization objective to our algorithm, that of maximizing the confidence in the results of the workflow replicas operating on reduced data.

D. Simulation-based Performance Evaluation

Another important part of the design of a resource allocation and scheduling algorithm is to be able to determine how this algorithm would perform in various scenarios (e.g., optimization goals, application configuration, target compute and storage infrastructure, or failure injection pattern). Such an evaluation requires to compute application execution metrics which is often done using analytical models. However, when it becomes important to account for the sharing of network resources, contention in accesses to a shared parallel file system, or overlap between computation, I/O, and network communication activities, as it is the case with near real-time analysis workflows, the analytical modeling problem becomes highly combinatorial. Models must thus make simplifying assumptions that do not hold in practice. A commonplace alternative is to resort to discrete-event simulation that can reproduce the behavior of a particular workflow executed on a given compute and storage infrastructure. The performance metrics needed to evaluate the quality of an algorithm are then computed from the event trace produced by the simulation.

In our roadmap we propose to build on the well established SimGrid framework [18]. SimGrid provides accurate and scalable simulation models and low-level abstractions for simulating distributed applications, runtime systems (e.g. implementations of the MPI standard), and distributed computing infrastructures ranging from commodity clusters to grid, clouds, and leadership class HPC systems. Typical simulators written with SimGrid comprises multiple *actors* (i.e., simulated sequential processes, defined by a main procedure written by the user), each of which creating one or more *activities* (i.e., computations, network communications, or I/O operations). Each activity is defined by a total amount of work to do (e.g., bytes to read, compute operations to perform) and the set of resources (i.e., CPUS, network links, or disks) that are used to perform this work. The simulation models at the core of SimGrid then determine the respective completion date of each of these activities to make the simulated time advance. SimGrid decouples the simulator that implements the logic of the application to study (e.g., a data generator and an analysis workflow) from the description of the simulated resources on which the application run and exposes mechanisms to inject delays and failures into simulations.

Data streaming and data reduction being at the core of the proposed approach to make near real-time workflow resilient, we specifically developed a versatile simulated *data transport layer* that covers the main driving principles of the ADIOS framework. Integrated to any SimGrid-based simulator, this standalone library allows simulated actors to publish or subscribe to self-described data objects. The library then handles the transport of data from publishers to subscribers using either files or streams. Data reduction can be implemented as part of the logic of the actors composing the simulated version of the considered near real-time analysis workflows. Thanks to this library and the aforementioned features of SimGrid, we thus have the necessary tools to assess the performance of the designed resource allocation and scheduling algorithms.

IV. Related Work

To improve the resilience of scientific workflows, commonplace approaches add more redundancy to data and computations to ensure the availability of the former and the completion of the latter. These include replication [6], [7] (i.e., produce multiple copies of the data, processes, and/or workflows to avoid single points of failure) and erasure coding [19] (i.e., generate parities that can be used to reconstruct lost data chunks in case of failures). However, as the volume of scientific data produced becomes astounding and the complexity of the data processing workflow increases, these conventional approaches to improve resilience become infeasible in many cases. The redundancy involved by these approaches grows linearly with the original data size which can cause significant storage and data movement overhead. The development of data reduction [12]–[14] and refactoring [15], [17] techniques offers new opportunities for addressing this challenge, that we intend to leverage to make near real-time workflows in fusion energy science more resilient.

The resilience of scientific workflow has also been considered from the scheduling and fault tolerance perspectives with the design of algorithms that allow workflows to continue their execution despite the failure of some computing or network resources. These scheduling algorithms rely on the replication of (some of) the components of a workflow and the distribution of theses replicas across distinct set of computing resources [8], [9], or exploit the production of checkpoints by the workflow components to ensure the resilient execution of the workflow thanks to rollback-and-recover mechanisms [20]. The effect that adding such fault tolerance mechanisms to the workflow scheduling process has on the quality of service experienced by the application has also been quantified [21]. However, none of these works considers the use of data reduction techniques. Then, all replicates and checkpoints are created using the full dataset at full accuracy, which we believe is not cost-effective and not a sustainable approach in light of the ever-increasing volume of data generated by modern scientific experiments and large-scale HPC simulations.

V. Conclusion

The operation of commercial nuclear fusion reactors will heavily rely on the analysis of the monitoring data produced by all the sensors and probes attached to these reactors. These analyses are very time sensitive to maximize the operation time of nuclear reactors and prevent catastrophic events. They also must be resilient as the lack of analysis results because of unexpected delays or failures may prevent operators to take the necessary informed decision in time.

In this paper we have developed our research and development roadmap towards resilient near real-time analysis workflows in fusion energy science. The proposed approach leverages data streaming, data reduction, and data refactoring techniques to trade some accuracy for both a faster execution of the analysis workflows and a reduction of the additional cost related to the replication of workflow components for resilience purposes. The combined use of these techniques leads to a complex multi-objective optimization problem for which we propose to design advanced resource allocation and scheduling algorithms. We have also presented how we do plan to evaluate the quality of the designed algorithms under various experimental conditions thanks to the development of comprehensive and realistic discrete-event simulators.

Our future work will be to research and implement the different steps of this roadmap and to validate the proposed approach by analyzing actual data coming from different prototypes of nuclear fusion reactors. To this end we will leverage and extend the capacities of existing software packages that already provide the necessary foundations to realize the presented roadmap.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number ”ERKJ414 – Resilient Federated Workflows in a Heterogeneous Computing Environment”.

References

[1] S. Hunyadi Murph and M. Murph, “Nuclear Fusion: the Promise of Endless Energy,” *Physical Sciences Reviews*, vol. 8, no. 10, pp. 3095–3118, 2023.

[2] 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.

[3] 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*, pp. 218–229, 2022.

[4] E. Dart, J. Zurawski, C. Hawk, B. L. Brown, and I. Monga, “ESnet Requirements Review Program Through the IRI Lens: A Meta-Analysis of Workflow Patterns Across DOE Office of Science Programs,” tech. rep., US Department of Energy (USDOE), 11 2023.

[5] B. L. Brown, W. L. Miller, D. Bard, A. Boehnlein, K. Fagnan, C. Guok, E. Lançon, S. J. Ramprakash, M. Shankar, and N. Schwarz, “Integrated Research Infrastructure architecture blueprint activity (final report),” tech. rep., US Department of Energy (USDOE), 2023.

[6] C. Engelmann, H. Ong, and S. Scott, “The Case for Modular Redundancy in Large-Scale High Performance Computing Systems,” in *Proceedings of the 8^{th} IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN)*, (Innsbruck, Austria), pp. 189–194, Feb. 2009.

[7] H. Lamehamedi, B. Szymanski, Z. Shentu, and E. Deelman, “Data Replication Strategies in Grid Environments,” in *Proceedings of the Fifth International Conference on Algorithms and Architectures for Parallel Processing*, pp. 378–383, 2002.

[8] X. Zhu, J. Wang, H. Guo, D. Zhu, L. T. Yang, and L. Liu, “Fault-Tolerant Scheduling for Real-Time Scientific Workflows with Elastic Resource Provisioning in Virtualized Clouds,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 12, pp. 3501–3517, 2016.

[9] R. Sirvent, R. M. Badia, and J. Labarta, “Graph-Based Task Replication for Workflow Applications,” in *Proceedings of the 11th IEEE International Conference on High Performance Computing and Communications*, pp. 20–28, 2009.

[10] F. Suter, R. F. Da Silva, A. Gainaru, and S. Klasky, “Driving Next-Generation Workflows from the Data Plane,” in *Proceedings of the IEEE 19th International Conference on e-Science (e-Science)*, (Limassol, Cyprus), pp. 1–10, Oct. 2023.

[11] 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.

[12] 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.

[13] S. Di and F. Cappello, “Fast Error-Bounded Lossy HPC data Compression with SZ,” in *Proceedings of the 30th IEEE International Parallel and Distributed Processing Symposium*, pp. 730–739, 2016.

[14] P. Lindstrom, “Fixed-Rate Compressed Floating-Point Arrays,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2674–2683, 2014.

[15] Z. Qiao, T. Lu, H. Luo, Q. Liu, S. Klasky, N. Podhorszki, and J. Wang, “SIRIUS: Enabling Progressive Data Exploration for Extreme-Scale Scientific Data,” *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 900–913, 2018.

[16] R. M. Churchill, C. S. Chang, J. Choi, R. Wang, S. Klasky, R. Kube, H. Park, M. J. Choi, J. S. Park, M. Wolf, R. Hager, S. Ku, S. Kampel, T. Carroll, K. Silber, E. Dart, and B. S. Cho, “A Framework for International Collaboration on ITER Using Large-Scale Data Transfer to Enable Near-Real-Time Analysis,” *Fusion Science and Technology*, vol. 77, no. 2, 2021.

[17] X. Liang, Q. Gong, J. Chen, B. Whitney, L. Wan, Q. Liu, D. Pugmire, R. Archibald, N. Podhorszki, and S. Klasky, “Error-controlled, Progressive, and Adaptable Retrieval of Scientific Data with Multilevel Decomposition,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2021.

[18] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, “Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms,” *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2899 – 2917, 2014.

[19] H. Weatherspoon and J. Kubiatowicz, “Erasure Coding Vs. Replication: A Quantitative Comparison,” in *Proceedings of the First International Workshop on Peer-to-Peer Systems*, vol. 2429 of *Lecture Notes in Computer Science*, pp. 328–337, 2002.

[20] G. Aupy, A. Benoit, H. Casanova, and Y. Robert, “Scheduling Computational Workflows on Failure-Prone Platforms,” in *Proceedings of the 17th Workshop on Advances on Parallel and Distributed Processing Symposium*, 2015.

[21] L. Ramakrishnan and D. A. Reed, “Performability modeling for scheduling and fault tolerance strategies for scientific workflows,” in *Proceedings of the 17th International Symposium on High Performance Distributed Computing*, (Boston, MA), p. 23–34, 2008.