

Data federation challenges in remote near-real-time fusion experiment data processing

Jong Youl Choi¹, Ruonan Wang¹, Michael Churchill², Ralph Kube², Minjun Choi⁴, Jinseop Park⁴, Jeremy Logan¹, Kshitij Mehta¹, Greg Eisenhauer³, Norbert Podhorszki¹, Matthew Wolf¹, CS Chang², and Scott Klasky¹

¹ Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA

² Princeton Plasma Physics Laboratory, Princeton, NJ 08536, USA

³ Georgia Institute of Technology, Atlanta, GA 30332, USA

⁴ National Fusion Research Institute, Daejeon, South Korea

Abstract. Fusion energy experiments and simulations provide critical information needed to plan future fusion reactors. As next-generation devices like ITER move toward long-pulse experiments, analyses, including AI and ML, should be performed in a wide range of time and computing constraints, from near-real-time constraints, between-shot analysis, and to campaign-wide long-term analysis. However, the data volume, velocity, and variety make it extremely challenging for analyses using only local computational resources. Researchers need the ability to compose and execute workflows spanning edge resources to large-scale high-performance computing facilities.

We present *Delta*, a system to address data analysis challenges, including AI/ML, in fusion science, by leveraging the ADIOS I/O library and middleware, to support executing science workflows over the wide area network for near-real-time streaming. We discuss the data federation challenges in performing remote workflows, focusing on on-going research work in (1) managing, reducing, and streaming data to minimize I/O and data movement overheads, (2) decompressing and reorganizing data for analysis, and (3) executing workflows for automated data analysis. We introduce examples for deep-learning based data analysis for the fusion domain and demonstrate how we use *Delta* to construct end-to-end workflows for a fusion device in Korea, connecting a remote DOE facility in the USA. The capability demonstrated by this project is the basis for improving the state of the art for near-real-time data federation amongst remote facilities.

Keywords: data federation, fusion, data streams, remote data analysis

Notice: This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

1 Introduction

As scientific experiments have gotten bigger and complex, scientists have had to rely more on concurrent, online, high-performance computing resources to analyze, reduce, and/or visualize the data streams. Such experiments may depend on hundreds or thousands of high-resolution, high-speed sensors to carefully collect the required data. Even further, there has emerged a new class of federated experiments that bring this experimental data together with large-scale simulation and the associated analysis capabilities to gain fast and accurate understanding of complex phenomena. However, the logical, time-scale, and physical discrepancies between the experiment site and high-computing facilities require scientists to explicitly spend time to manage their data, resources, and execution patterns in the distributed setting, while still trying to find time to do their science.

Plasma science, and fusion energy research in particular, have already made the transition into exploring these sorts of federated and hybrid experimental/computational environments. As next generation devices like ITER [17] move toward long-pulse experiments, not only will the amount of data increase, but there will also be an intense focus on this federation of experimental and simulation results. In particular, ITER has announced the intention that no run should occur on the physical facility without first having a full set of simulated results to compare against. Any deviations from predicted plasma behavior should be monitored and such monitoring should trigger additional analysis/simulation to identify causes of deviations. If significant deviations can be identified and assessed rapidly during an experiment, then such information can contribute to the optimization of the experimental plan and ultimately contribute to the timely realization of the ITER mission.

In this context, the workflows needed for timely and autonomous delivery of the analytics become extremely important. There are a host of new, next-generation compute-intensive AI and ML analyses that will need to be run over a wide range of time (e.g., near-real-time, between-shot) and computing (e.g., edge device, GPU accelerator) constraints. However, the data volume, velocity, and variety make it extremely challenging for AI/ML analysis using only local computational resources. Therefore, researchers need the ability to compose and execute workflows spanning edge resources to large-scale high performance computing facilities.

Currently, constructing such a capability is labor-intensive and distracting from the core science investigation. We represent a team of computer science and fusion scientists who are experiencing such data federation challenges. In our research and development of software tools to support current and future workflows aimed at next-generation devices such as ITER, we have identified what we view as core challenges and the capabilities that are needed to address them. We present a framework for aDaptive rEaL Time Analysis, *Delta* for short, as our progress that addresses such data federation challenges for remote fusion experiment analysis. Specifically, we design *Delta* to provide services and runtime support for the following core capabilities: 1) Temporal analysis, 2) Feature detection, 3) Data reduction/compression, and 4) Data movement and

access. All of these are identified as a critical, generic tasks needed to support federated fusion data analysis.

This infrastructure for *Delta* is constructed using our Adaptable I/O System (ADIOS) [12] as a backbone, which offers a high-performance data management solution for HPC storage as well as an online data management based on a publish/subscribe abstraction [8,13]. We take advantage of ADIOS’ publish/subscribe interface in which users can federate data producers (sensors or simulations) and consumers (analysis or visualization application).

In this paper, our contribution is in identifying these core capabilities and in offering experimental validation of this approach for real scientific investigations. In Section 2, we discuss two fusion experiments and their data, which motivate our research. One of which is a multi-national collaboration between the KSTAR facility in South Korea, researchers at Princeton Plasma Physics Laboratory, and contributors from Oak Ridge National Laboratory as well as the NERSC Computing Center. In Section 3, we demonstrate how we use *Delta* to attempt to support the temporal and spatial evolution of the plasma within the device during long-pulse experiments to simulated results and online analytic pipelines.

An additional concern that we address in our *Delta* experiments has to do with the introduction of AI/ML methods into the federated computing pipeline. In many ways, AI/ML methods represent a fourth pillar to the traditional approaches to science of experiment, theory, and simulation. As such, we found that there were unique constraints and considerations that needed to be respected in order to fully integrate AI/ML methods into *Delta*. Although utilizing a previously-trained model may not be very different from other analytics methods, an ML algorithm we use in this paper (the variational auto-encoder, VAE), which allows for efficient compression and generation of surrogate data representations, combines training and model usage in novel ways. The experience of integrating these features into the capability matrix of *Delta* has offered lessons that we believe can be applicable to other areas of science as well. Some of these connections will be discussed in Section 4.

1.1 Related work

We briefly discuss existing work prior to *Delta* in the field of remote data federation and AI/ML applications for data streams.

Remote data federation: Data federation for stream data processing over a wide-area network has been an active research area for the last decades. In our previous work, WASP [5], we explored our initial research on how to use data streams to remotely federate data from KSTAR and fusion simulation with the previous ADIOS version. In this paper, we extend our work with new data transport engines, called DataMan and SST, by leveraging a newer version of ADIOS and focus on demonstrating data federation between KSTAR in Korea and one of DOE HPC sites, NERSC, in the USA.

Amazon Lambda [2] and Apache OpenWhisk [1] utilizes the cloud platform to ingest data collected from distributed sensors or the internet at scale. Our

work focus more on performing mission-critical analysis for science, targeted toward near-real-time support.

While Codor [20] and Globus [7] focused on utilizing multi-institute computing resources, integrating remote computing sites and data facilities has been a new trend as the big data challenge emerges. More recently, Cray announced the next interconnect network, Slingshot [3], which will include the capability to connect remote third-party data centers with HPC centers.

AI/ML applications for streams: As data volume grows at an exceeding rate, several floating-point lossy data compressors, such as ZFP [11], SZ [6], and MGARD [4], have been actively researched and applied in many science applications. Researchers start looking at deep learning methods for data compression as well. While most lossy compression methods are broadly based on numerical solutions for regression, interpolation, and decompositions, data compression with deep learning is mostly based on developing a generative process for the data provided by users. A family of generative deep learning models, such as Variational Autoencoders (VAE) and Generative Adversarial Network (GAN), has been developed to reconstruct missing parts of an image or compress image data. PixelRNN [16], WaveNet [15], and realtime adaptive compression [19] are some of the examples. In our paper we explore vector quantization based VAE, VQ-VAE [14], for our fusion data compression.

2 Remote fusion experiment

ITER [17] is the largest, next-generation international experimental fusion reactor currently being built in France, aiming to begin operation with first plasma in December 2025. ITER will make a huge impact on how fusion scientists process and federate experimental data for online analysis by integrating their workflow using local and remote compute resources. In this section, we describe the current fusion experiments and data challenges by using the cases of two current machines, KSTAR and NSTX-U, to give an overview of fusion data challenges.

2.1 KSTAR fusion experiment and workflows

The Korea Superconducting Tokamak Advanced Research (KSTAR) [10] is a medium-sized tokamak (or a fusion reactor). This fusion device confines a plasma whose core temperature may exceed 50 million degrees for up to 5 minutes. Achieving such long plasma confinement times with available technology requires the use of superconducting magnet coils and KSTAR is one of only a hand full superconducting tokamaks ever built. Operating with such uniquely long pulses, KSTAR data is a relevant target to develop federated data analysis systems for.

Numerous scientific diagnostics interrogate the plasma and its supporting machinery in plasma discharges and generate a variety of data streams. For plasma diagnostics, designed to investigate physical processes occurring in the fusion plasma, read-outs of high-dimensional data on a microsecond time-scale is

required. Electron Cyclotron Emission Imaging (ECEI) diagnostics for example measure the intensity of emissions by free electrons in the plasma. This information allows to recover fluctuations of the electrons temperature, which in turn gives physicists important information on the macro- and micro-scale dynamics of the confined fusion plasma. Modern ECEI systems, as the one installed in KSTAR, measure the emission intensity using hundreds of spatial channels with Megahertz sampling rates. Generating a data stream of about 5 GByte per second, this diagnostic produces a fast, high-dimensional data stream that makes it challenging to analyze in near real-time.

A common analysis workflow for ECEI time series data is spectral analysis routines which compare ECEI channel pair combinations. Since short-time Fourier transformations are used for this workflow, the time series data for this workflow can be divided in millisecond long chunks. Furthermore all channel pair combinations can be treated separate from one another. That is, the input data for this algorithm consists of numerous relatively small data chunks which can all be analyzed independent from one another. Such workflows are typically performed manually by fusion scientists in a batch-wise fashion some time after a given plasma shot. While this workflow involves the calculation of standard spectral quantities, performing it on the large amount of available ECEI data render it a substantial task.

We envision federated data analysis systems that perform such a prototypical workflow with little configuration. The system should allow to offload the data analysis to HPC systems that can handle fast, high-dimensional incoming data stream and perform the analysis in near real-time to facilitate decision making.

2.2 NSTX-U fusion experiment and workflows

National Spherical Torus Experiment - Upgrade (NSTX-U), a magnetic fusion device that is operated by the Princeton Plasma Physics Laboratory (PPPL), provides another exemplary analysis workflow that can be enhanced using federated compute resources.

One of the diagnostics installed in NSTX-U is the so-called gas-puff imaging (GPI) system. This diagnostic measures light emissions from the edge of the confined plasma using a fast framing 2d camera that captures 80-by-64 pixel frames at a rate of about 400,000 frames per second. A particularly interesting phenomena that can be studied by GPI is the motion of plasma filaments with excess pressure that transport a significant amount of hot plasma towards the main chamber walls. As such they may affect confinement performance and erode material surfaces. When observed in the two-dimensional view of the GPI, these filaments present their circular cross-section, usually referred to as blobs.

To better understand this blobby transport, physicists study statistics of the blobs observed in the plasma discharges, including their cross-field sizes, amplitudes and velocities. All of these are readily calculated from GPI data. Modern algorithms rely either on manually selected filters or use machine learning algorithms to detect blobs in single frames and compile statistics on their extent and their dynamics.

Now NSTX-U plasma discharges last only a few seconds and the GPI diagnostic only collects data for a fraction of that time, typically about 500 MBytes per discharge. Blob detection can be performed in individual frames and is therefore parallelizable in a similar way as the ECEI workflow. To compile the motion of plasma blobs in a given discharge however, the positions of blobs in the entire sequence of images is required. In other words, the data needs to be made available in full and in order. This, together with the fact that we are using a neural network based architecture to detect blobs motivates us to treat this workflow as a machine learning toy model for the context of federated data analysis.

The requirements posed by these two workflows allow us to derive capabilities we want from a federated workflow system. While this list should not be seen as exhaustive, the workflows described above capture a broad range of requirements for other workflows.

Capabilities to drive future federated fusion experiments

- Reliably transfer large volumes of data with high velocity from data production site to HPC resources with minimum overhead (e.g., memory-to-memory)
- Facilitate execution of data analysis routines with as few as possible changes to existing codes
- Adaptively adjust data analysis and compression techniques in real-time to be reactive to analysis results and network performance

3 *Delta*: Supporting Federated Data Today

We present the *Delta* framework as a concrete example on how federated data analysis workflows may be implemented and facilitated with ADIOS. We also discuss how recent AI/ML methods can be incorporated in such a framework.

3.1 ECEI analysis with *Delta*

Delta is a python framework that facilitates federated data analysis workflows. In particular, it is used to analyze data from the KSTAR ECEI diagnostic in near real-time by using remote high-performance computing resources at Cori, NERSC. While we aim to facilitate general federated data analysis workflows with *Delta* in the future, it has been designed around the task to perform a suite of spectral analysis on time-series of imaging data from KSTAR’s ECEI diagnostic. We choose this task because it can be seen as a prototype workflow for analysis of other diagnostic data, such as probes or magnetic flux sensors, and because the the ECEI diagnostic produces a high-velocity data stream.

Figure 1 illustrates the network topology that *Delta* targets, KSTAR and NERSC as the experimental and compute site and as well as an additional

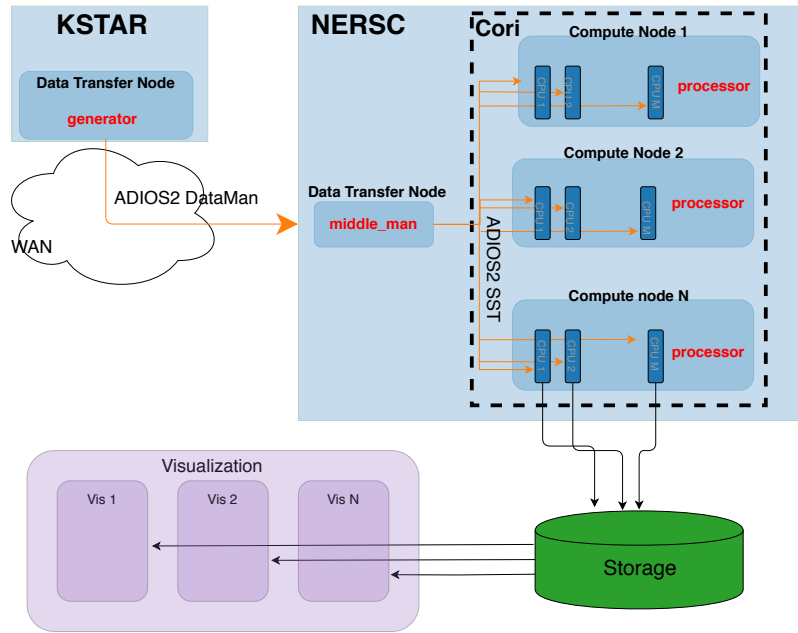


Fig. 1: Where does *Delta* live? *Delta* consists of multiple distributed components (generator, middleman, and processor) and ADIOS provides remote data federation. More details can be found in [9].

storage backend that feeds live-streams of analyzed data into visualizers. As a framework for distributed computing, *Delta* is comprised of multiple components that connect the individual sites with one another. At KSTAR, a **generator** sources and streams sequential tranches of ECEI data to NERSC. This stream is received by a **middleman** running on the NERSC Data Transfer Node (DTN) and forwarded to a **processor** on Cori. This processor uses a **dispatcher**, a thread for asynchronous I/O, and uses MPI to execute a user-defined suite of spectral analysis on compute nodes. Analyzed data is stored in a storage backend, such as a database. Data visualizers can then ingest this data and make it available to researchers in near real-time.

Delta consists of multiple distributed components and more comprehensive discussion of achieved performance on the architecture can be found in [9]. The ECEI workflow consists of calculating spectral quantities, the cross-phase, cross-correlations, coherence and cross-power for 18,336 unique channel pair combinations from the 192 channels of the diagnostic on 500 time chunks consisting of 10,000 data points per channel. Launching this analysis on Cori, where the processor runs on 32 compute nodes, *Delta* performs the ECEI workflow in 339 s. To set this runtime in context, performing this workflow using a naive single-core implementation takes multiple hours.

3.2 Adaptable data transfers using data compression and filtering

One of the challenges in remote data federation for near real-time processing is how to reduce the amount of data to transfer over a wide-area network. Data filtering and compression are two of the most common practices for remote processing. If one can accurately detect specific features of interest in the data stream, we can filter out unnecessary parts and send only the necessary data for remote processing. With lossy compression, if one can compress data for sending and receivers can reconstruct the original data with minimum loss of information, we can save network bandwidth. Filtering and compression can be combined or used separately. While developing algorithms and methods for feature detection and information preserving encoding based on conventional machine learning techniques have been active research areas and numerous techniques have been deployed so far, deep learning-based approaches are getting attention recently for its flexible and customizable design concept.

We explore two DNN-based methods, YOLO and VQ-VAE, with the NSTX (before the upgrade) GPI images, real-world fusion data set, to demonstrate AI/ML-based data filtering and compression workflows.

YOLO filtering We explore data filtering with DNN-based feature detection. YOLO [18] is a state-of-the-art DNN algorithm for real-time object detection. It is based on the idea of using a single-pass neural network regression for detection. In contrast, most object detection systems are based on multi-pass classifiers and compute-intensive localizers, which might not be efficient for fusion data toward near real-time processing.

We train a YOLO model with a set of exemplar blob images extracted from the NSTX GPI data and conducted a set of experiments to test the detection performance against the data set prepared for validation. We compare YOLO with a conventional blob detection method using ellipse curve fitting. In the ellipse curve fitting, we find contours of blob boundaries and perform a fitting to find ellipse parameters to fit each contour into an ellipse shape. In contrast, YOLO divides the whole region into multiple subblocks and performs a one-pass regression to decide whether a chunk contains an object of interest, which is a blob in our case. Fig. 2 demonstrate how YOLO can detect blobs compared with ellipse fitting based blob detection. We extend our test to check whether the model we built can be used for the out-of-sample data. Table 1 shows various accuracy measurements for four different data sets we prepared. Except for Set A, which is the in-sample data, Set B-D are out-of-sample data. The performance result shows that YOLO detection performance remained closely similar even for the out-of-sample datasets.

VAE compression Next, we explore how we can use DNN to build a generative model that can efficiently find a compressed representation of data from which users can reconstruct the data. We use VQ-VAE [14], a variant of Variational autoencoder (VAE) families, as a reduced representation finder. While

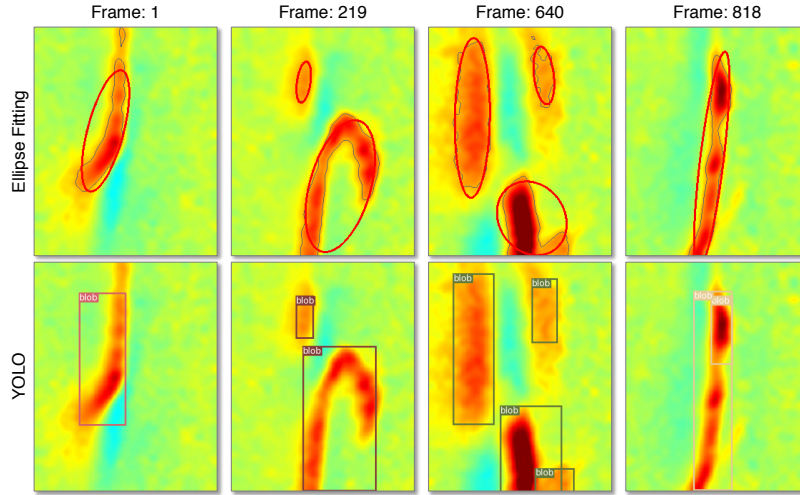


Fig. 2: Blob detection demonstrated with NSTX GPI images, comparing conventional ellipse fitting with DNN-based YOLO.

	Recall(%)	Precision(%)	Accuracy(%)
Set A	96.8	93.2	95.2
Set B	96.0	90.8	97.2
Set C	93.7	87.8	96.6
Set D	91.7	80.8	98.9

Table 1: Performance of YOLO with NSTX GPI data set.

conventional VAEs focusing on finding Gaussian distribution as a latent representation, VQ-VAE employs vector quantification to capture a more discrete nature of features in the data.

The core of the VQ-VAE algorithm is to find an underlying latent representation of encoding data, which can be decoded with a minimum loss of information. In the context of data compression, the latent representation can be considered as a reduced form of data. By sending this reduced representation, a receiver can reconstruct the original data with a certain degree of information loss. Compared with conventional lossy data compressions, such as ZFP and SZ, VQ-VAE is generative as it recovers the process of data generation and is customizable as the network architectures and parameters are tuned for the user’s data set.

Fig. 3 demonstrate how VQ-VAE can learn NSTX GPI images and reconstruct. The errors measured by Root Mean Square Error (RMSE) show about 22% errors without any treatment. With a noise filter applied, we further lowered the errors around 12%. In this experiment, we achieved 8x data compression ratio. We leave more extensive comparison studies for the next work.

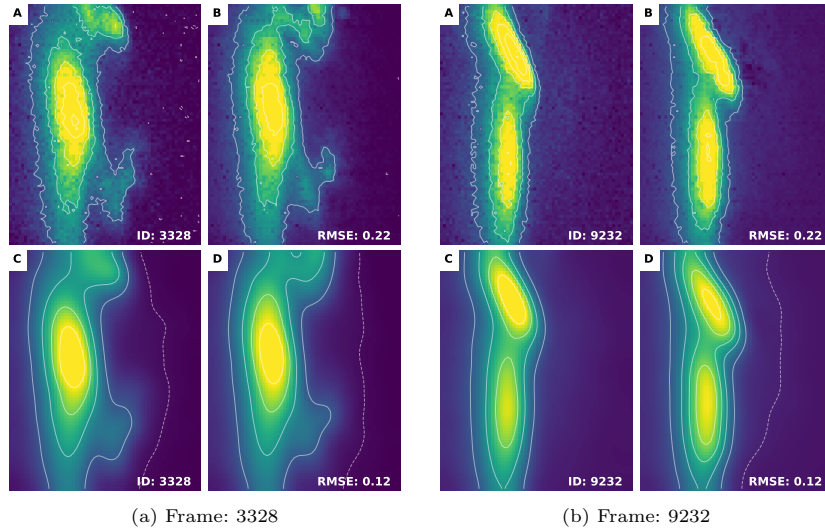


Fig. 3: Demonstration of VQ-VAE reconstruction with NSTX GPI images. (A) original NSTX GPI image, (B) reconstructed image by VQ-VAE, (C) original image with Gaussian denoising treatment, and (D) reconstructed VQ-VAE image followed by Gaussian denoising. Frame numbers and Root Mean Square Error (RMSE) metrics are shown on the bottom-right corner.

3.3 Remote Data Federation Services with ADIOS

ADIOS [12] is an HPC I/O library designed to provide I/O services to users, such as data management, transport, and transforms. We leverage ADIOS to build our framework, *Delta*. ADIOS supports remote data federation with a convenient and unified data interface that allows scientists to handle data and its metadata both as files and data streams.

To transport data as streams, ADIOS provides a plug-in approach; ADIOS provides multiple engines users can choose depending on the need. We use two main ADIOS engines, called SST and DataMan, which can be used for a wide-area network transport. We briefly introduce SST and DataMan engine used in this paper and provide experiment results to demonstrate how we use ADIOS for remote analysis.

Scalable Staging Transport (SST) SST is a flexible data transport engine in ADIOS. It is designed to utilize a wide range of network protocols, such as RDMA, TCP, and UDP, widely used for both high-performance HPC networks and long-haul wide-area network connections. This flexibility is an important feature, especially for remote data processing. With SST, users can switch network protocols easily to adapt to different network environments.

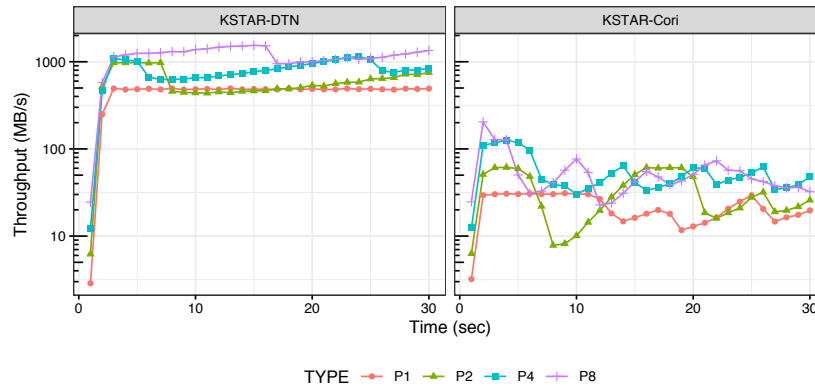


Fig. 4: Network performance between KSTAR and NERSC DTN nodes (left) and between KSTAR and Cori compute node (right), measured by iperf3 with up to 8 parallel streams. The Y axis (throughput) is log scaled.

On the other hand, data flow control is another important aspect in data stream processing. It is common the speed of data generation and processing is different. Without caching or buffering, either the generator or the receiver needs to wait simply by wasting valuable CPU cycles and resources. SST provides a buffering policy for users to control, such as changing buffer size or applying rules when a buffer is full or not.

DataMan While SST is an engine for the best flexibility, DataMan is another ADIOS engine mainly focusing on performant data transport over a wide area network. DataMan is specifically optimized for long-distance low-latency data movement, leveraging the ZeroMQ library for asynchronous messaging.

Experiments First, we measure the bandwidths between KSTAR and NERSC. We have two connection points in NERSC. We can connect to NERSC through the NERSC DTNs, a set of dedicated nodes for wide-area data transfer. NERSC also allows users to connect directly to Cori compute nodes as an experimental feature. Fig. 4 shows the bandwidths, measured by iperf3, a) between KSTAR and NERSC DTN and b) between KSTAR and NERSC Cori compute node, with a varying number of parallel streams, up to 8.

In both cases, we observed variances in the throughput between KSTAR and NERSC due to the congestion. Also, there are large differences in throughput for sending from Korea to NERSC DTN or directly to Cori compute nodes. It implies we need a transport mechanism adaptively adjust throughput developing on the dynamic network conditions. DataMan is currently implemented adaptive performance, and further improvement is under development.

In Fig. 5, we performed the ECEI workflow (Fig. 1) with 2 different scenarios; (a) 2-node scenario where we send ECEI data from KSTAR directly to

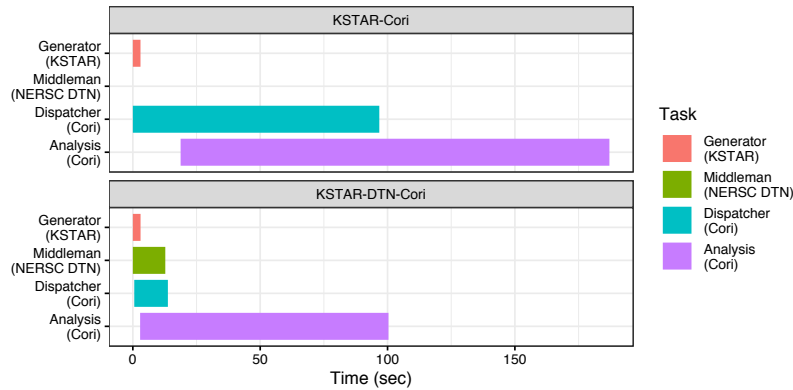


Fig. 5: Timelines of tasks performed in two cases; (a) 2-node scenario (KSTAR-Cori) and (b) 3-node scenario (KSTAR-DTN-Cori). We used ADIOS DataMan engine for both KSTAR-Cori and KSTAR-DTN transport. ADIOS SST engine was used intra NERSC transportation, DTN-Cori.

Cori compute node. We use DataMan between KSTAR and Cori, and (b) 3-node scenario where we send ECEI data to a middle man running on a NERSC DTN node, which receives data from KSTAR and concurrently forwards to the ECEI analysis processes running on Cori compute nodes. We use DataMan for KSTAR-NERSC DTN over wide area network and SST between NERSC DTN and Cori compute node communication. Although we deployed an extra process (middleman) in our 3-node scenarios, we were able to finish ECEI analysis task earlier than using 2-node scenario. ADIOS’s transport engine enabled us to achieve flexible workflow composition and execution.

4 Toward Plasma Science of the Future

We have explored how we can support the next-generation data analysis challenges in fusion science, based on the real-world data and workflows obtained through our collaboration with KSTAR and NSTX-U, which we believe can be applicable to the upcoming ITER challenges. We summarize key capabilities identified for the success of the future fusion analysis workflow.

First, an adaptive workflow system with intelligence is required. This will allow the system to adjust to the state of the network and/or receiver agent responsiveness, by for example sending coarser or reduced data when the system lacks the network fidelity to accomplish the streaming, and returning to complete full dataset transfers when possible. With the current ECEI workflow with *Delta*, we found optimal network transfer could result in sent data chunks lost if the receiver agent was overwhelmed and not able to keep pace. Adaptivity built into the framework will allow intelligent, automated throttling (through data coarsening, or delayed network transfer) depending on the data stream and user

needs, which for some streams will require complete datasets (such as the GPI workflow), and for others can accept data loss (such as the ECEI workflow).

Second, extending on the concept of adaptivity for data streaming, a flexible yet tightly integrated workflow execution will be enormously beneficial in terms of data monitoring and workflow automation. This enables reactive workflow components, where additional, more expensive higher-tiered analyses can be spun up based on an anomaly/novelty detected in lower-tiered analysis on data streams. For example, a sudden change in the correlation between certain ECEI channel pairs could indicate the presence of magnetohydrodynamics (MHD) mode activity, which could automatically trigger simulations into the current MHD stability limits based on magnetic and kinetic plasma profiles diagnostics. Instead of waiting for the scientist to manually comb through the data and run the simulations themselves, the results would be present and the scientists notified to draw their attention to such time points of interest.

Third, incorporating next-generation analysis, including additional AI/ML, which can take advantage of hardware-based acceleration can be a game-changer for how fusion scientists analyze their experiments. Automating and accelerating the tasks fusion scientists must perform in time for more immediate feedback will allow scientists to extract richer insights from their experimental and simulation data. Interfacing humans and AI/ML in this way can be greatly beneficial, but must be researched and explored in the context of fusion science, to determine best practices and needs, for example if techniques found in continual learning are needed to update machine learning models as new tasks and data distributions present themselves.

5 Conclusion

Fusion experiments pose unique data federation challenges. Not only the volume, velocity, variety of data the fusion experiments produce, but also the time constraint for between-shot or near-real-time analysis makes performing data analysis workflows challenging. In addition, utilizing remote high-performance computing resources is being explored to perform compute-intensive, state-of-the-art analysis at scale and/or to couple with impromptu simulation runs on an HPC machine for making timely decisions for the next shot. Indeed, constructing and executing such analysis pipelines combined with multiple remote sources is challenging and efficient data federation for remote data processing is necessary.

We present *Delta* as a framework to facilitate fusion data analysis using remote computing resources, taking advantage of ADIOS' publish/subscribe interface where users can federate data producers (sensors or simulations) and consumers (analysis or visualization application) regardless of their locations. By using real-world fusion data and analysis workflows from KSTAR and NSTX-U as a use case, we demonstrate how we use *Delta* and ADIOS for real-world fusion analysis workflows to federate experiment data and remote resources over wide

area networks and introduce recent AI/ML methods to be applicable for fusion data streams.

Building on this framework, and the ideas and principles underlying it can provide fusion scientists extended remote computing capabilities to accelerate their current workflows, and make possible workflows not currently employed or even imagined today. AI/ML can play a key enabling role in this regard, offering the promise of faster, better analysis to provide scientists deeper insights and guidance.

Acknowledgement

This research was supported by the Department of Energy’s SciDAC RAPIDS Institute and the HBPS SciDAC Partnership, as well as the Exascale Computing Project (17-SC-20-SC), a collaborative effort of U.S. Department of Energy Office of Science and the National Nuclear Security Administration. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, under AC02-09CH11466. This research used resources of the Argonne and Oak Ridge Leadership Computing Facilities, DOE Office of Science User Facilities supported under Contracts DE-AC02-06CH11357 and DE-AC05-00OR22725, respectively, as well as the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231. The research at KSTAR was conducted as part of KSTAR R&D Program of National Fusion Research Institute of Korea (EN2001-11).

References

1. Apache OpenWhisk: Open source serverless cloud platform. <https://openwhisk.apache.org/>
2. AWS Lambda - serverless compute - Amazon Web Services. <https://aws.amazon.com/lambda/>
3. Slingshot: The interconnect for the exascale era. Tech. rep., Cray Inc. (2019)
4. Ainsworth, M., Tugluk, O., Whitney, B., Klasky, S.: Mgard: a multilevel technique for compression of floating-point data. In: DRBSD-2 Workshop at Supercomputing (2017)
5. Choi, J.Y., Kurc, T., Logan, J., Wolf, M., Suchyta, E., Kress, J., Pugmire, D., Podhorszki, N., Byun, E.K., Ainsworth, M., et al.: Stream processing for near real-time scientific data analysis. In: 2016 New York Scientific Data Summit (NYSDDS), pp. 1–8. IEEE (2016)
6. Di, S., Cappello, F.: Fast error-bounded lossy hpc data compression with sz. In: 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 730–739. IEEE (2016)
7. Foster, I., Kesselman, C.: Globus: A metacomputing infrastructure toolkit. The International Journal of Supercomputer Applications and High Performance Computing **11**(2), 115–128 (1997)

8. Klasky, S., Wolf, M., Ainsworth, M., Atkins, C., Choi, J., Eisenhauer, G., Geveci, B., Godoy, W., Kim, M., Kress, J., Kurc, T., Liu, Q., Logan, J., Maccabe, A.B., Mehta, K., Ostrouchov, G., Parashar, M., Podhorszki, N., Pugmire, D., Suchyta, E., Wan, L., Wang, R.: A view from ORNL: Scientific data research opportunities in the big data age. In: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), pp. 1357–1368 (2018). DOI 10.1109/ICDCS.2018.00136
9. Kube, R., Churchill, R.M., Choi, J.Y., Wang, R., Klasky, S., Chang, C., Choi, M.J., Park, J.: Leading magnetic fusion energy science into the big-and-fast data lane (2020). DOI 10.25080/issn.2575-9752
10. Lee, G., Kwon, M., Doh, C., Hong, B., Kim, K., Cho, M., Namkung, W., Chang, C., Kim, Y., Kim, J., et al.: Design and construction of the KSTAR tokamak. *Nuclear Fusion* **41**(10), 1515 (2001)
11. Lindstrom, P., Isenburg, M.: Fast and efficient compression of floating-point data. *IEEE transactions on visualization and computer graphics* **12**(5), 1245–1250 (2006)
12. Liu, Q., Logan, J., Tian, Y., Abbasi, H., Podhorszki, N., Choi, J.Y., Klasky, S., Tchoua, R., Lofstead, J., Oldfield, R., et al.: Hello ADIOS: the challenges and lessons of developing leadership class I/O frameworks. *Concurrency and Computation: Practice and Experience* **26**(7), 1453–1473 (2014)
13. Logan, J., Chen, J., Choi, J.Y., Kress, J., Eisenhauer, G., Godoy, W., Kim, M., Liu, Q., Mehta, K., Ostrouchov, G., Podhorszki, N., Pugmire, D., Suchyta, E., Thompson, N., Wan, L., Wang, R., Whitney, B., Wolf, M., Klasky, S.: Extending the publish/subscribe abstraction for high-performance I/O and data management at extreme scale. *Data Engineering Bulletin* (2020)
14. van den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. In: *Advances in Neural Information Processing Systems*, pp. 6306–6315 (2017)
15. Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016)
16. Oord, A.v.d., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759* (2016)
17. Rebut, P.H., et al.: Iter: the first experimental fusion reactor. *Fusion engineering and design* **30**(1-2), 85–118 (1995)
18. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788 (2016)
19. Rippel, O., Bourdev, L.: Real-time adaptive image compression. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2922–2930. JMLR. org (2017)
20. Thain, D., Tannenbaum, T., Livny, M.: Distributed computing in practice: the condor experience. *Concurrency and computation: practice and experience* **17**(2-4), 323–356 (2005)