Enabling Data Streaming-based Science Gateways through Federated Cyberinfrastructure

Ivan Rodero* Yubo Qin* Jesus Valls* Anthony Simonet* J. J. Villalobos* Manish Parashar* Choohan Youn†
Cong Wang‡, Komal Thareja‡ Paul Ruth‡, George Papadimitriou§, Eric Lyons¶, Michael Zink¶

*Rutgers Discovery Informatics Institute, NJ, USA
†San Diego Supercomputing Center, CA, USA
‡RENCI, University of North Carolina at Chapel Hill, NC, USA
§Information Sciences Institute, University of Southern California, CA, USA
¶Department of Electrical and Computer Engineering, University of Massachusetts at Amherst, MA, USA

***Abstract*—Large scientific facilities are unique and complex infrastructures that have become fundamental instruments for enabling high quality, world-leading research to tackle scientific problems at unprecedented scales. Cyberinfrastructure (CI) is an essential component of these facilities, providing the user community with access to data, data products, and services with the potential to transform data into knowledge. However, the timely evolution of the CI available at large facilities is challenging and can result in science communities requirements not being fully satisfied. Furthermore, integrating CI across multiple facilities as part of a scientific workflow is hard, resulting in data silos.**

In this paper, we explore how science gateways can provide improved user experiences and services that may not be offered at large facility datacenters. Using a science gateway supported by the Science Gateway Community Institute, which provides subscription-based delivery of streamed data and data products from the NSF Ocean Observatories Initiative (OOI), we propose a system that enables streaming-based capabilities and workflows using data from large facilities, such as the OOI, in a scalable manner. We leverage data infrastructure building blocks, such as the Virtual Data Collaboratory, which provides data and computing capabilities in the continuum to efficiently and collaboratively integrate multiple data-centric CIs, build data-driven workflows, and connect large facilities data sources with NSF-funded CI, such as XSEDE. We also introduce architectural solutions for running these workflows using dynamically provisioned federated CI.

***Index Terms*—Cyberinfrastructure, science gateways, data streaming, virtual data collaboratory, scientific workflows, federated infrastructure**

I. Introduction

Open, large-scale facilities are already an essential part of scientific observation systems; they are openly accessible and deliver a broad range of data and data products. For example, current and future experimental and observation facilities, such as the Large Hadron Collider (LHC) [1], the Laser Interferometer Gravitational-Wave Observatory (LIGO) [2], the Large Synoptic Survey Telescope (LSST) [3], and the Square Kilometre Array (SKA) [4], provide or will provide massive amounts of data and data products that have the potential to revolutionize the science and engineering domains. However, despite providing reliable and pervasive access to the data and data products, users typically have to download the data of interest and process them using local resources. Consequently, transforming these data and data products into insights requires local access to powerful computing, storage, and networking resources. These requirements can significantly limit the impact of the data, especially for researchers, educators, and students who do not have access to such capabilities. Furthermore, large scientific facilities and advanced cyberinfrastructure (CI) remain largely disconnected [5], and, as a result, users are forced to actively be part of the process that moves data from large facilities to local computational resources or NSF-funded CI. The size of the data makes this process very challenging as it involves making use of wide area networks, which often connect users with limited bandwidth. Moreover, users might not have enough storage at their disposal to archive all the required data before executing their scientific simulations. This data delivery mode thus becomes inefficient and limits the potential utility that the data would have if processed in an automatic manner.

The goal of this effort is to bring together science gateways with data infrastructure building blocks (i.e., the Virtual Data Collaboratory [VDC] platform) to provide access to observatory data and data products using effective delivery mechanisms (e.g., push-based data delivery not offered by large facilities CIs) in a scalable manner. We also aim to integrate large facilities, such as the OOI, with existing NSF-funded CI to support end-to-end workflows. The proposed science gateway provides ways to deliver OOI streamed data using pub/sub mechanisms via an Apache Kafka-based resource broker system; however, it can be used with other observatories and can seamlessly combine data and data products from multiple sources collaboratively provided through the VDC. For example, the dashboard application described in this paper delivers automatically generated scientifically rich data products through a streaming-based workflow engine, which avoids downloading complete data and can significantly reduce the time to scientific insights. Existing scientific workflow engines [6] include general systems, such as Pegasus [7], Kepler [8], and Triana [9], and domain-specific workflow systems, such as Taverna [10]. Some of these systems incorporate streaming workflow capabilities that have been applied in science applications [11] and provide processing and querying mechanisms on data streams, such as Spark-streaming, STREAM [12], and Aurora [13]. While a Python-based scientific workflow engine that natively supports Kafka data streams has been designed and deployed to seamlessly integrate the dashboard functionalities with the proposed science gateway, we integrate it with the DyNamo framework to automatically provision workflows with network and computing resources across federated NSF-funded CI.

II. Towards Optimizing Data Delivery for Large Scientific Facilities

Data and services provided by shared facilities, such as large-scale instruments and observatories, have become important enablers of scientific insights and discoveries across many science and engineering disciplines. As a result, the quality of service provided by these facilities and the associated CI is an important concern and has direct impact on the ability of researchers to effectively use these facilities. However, ensuring quality of service can be challenging due to the remote locations and distributed nature of these instruments and observatories, combined with the rapidly growing volumes of data produced on the one hand and the size of the user communities on the other.

This paper addresses key issues associated with these challenges by exploring how push-based publish/subscribe data delivery models coupled with emerging CI solutions can be used to improve the performance, usability, and science impact of data and services provided by facilities. The proposed architecture and mechanisms can deliver capabilities that are not offered by the data facility CI and have the potential to improve the user experience and scientific insights. For example, the dashboard use-case application described in section IV delivers data and data products via a streaming-based publish/subscribe science gateway (see section III), which is not currently available in the OOI CI.

A. The Virtual Data Collaboratory

In this work, we leverage data infrastructure building blocks, such as the VDC platform [14], to efficiently and collaboratively integrate multiple data-centric CIs, build data-driven workflows, and connect large facilities data sources with NSF-funded CI, such as XSEDE. The VDC is a regional CI composed of dedicated network and data transfer nodes, storage, computing resources, and data services for enabling open, collaborative, and interdisciplinary science. The VDC accomplishes this goal by providing seamless access to data and tools that are part of the federation and a set of unique features, as shown in figure 1. In addition to the dedicated high-speed network, the VDC provides a set of data services, including indexing, cataloging, sharing, and metadata management, and an execution platform that allows distribution of complex distributed analytics and abstractions for smart data delivery strategies.

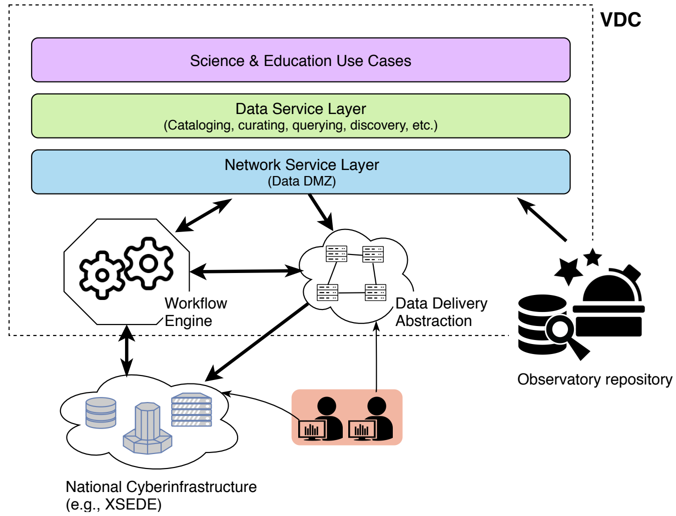

Fig. 1. Overall system architecture within the VDC platform

In this paper, we focus on streaming-based publish/subscribe data delivery models. As shown in figure 1, the VDC provides abstractions to which users can subscribe via the science gateway described in section III, but it also enables content delivery through existing NSF-funded CI, such as XSEDE. Furthermore, the VDC provides a comprehensive framework for implementing scientific workflows, which can combine multiple data sources that are part of the VDC federation.

III. Science Gateway: Message Broker System for Push-based OOI Data Delivery

The NSF OOI [5], [15] is a networked ocean research observatory with arrays of instrumented water column moorings and buoys, profilers, gliders, and autonomous underwater vehicles (AUV) within different open ocean and coastal regions. The OOI infrastructure also includes a cabled array of instrumented seafloor platforms and water column moorings on the Juan de Fuca tectonic plate. This networked system of instruments, moored and mobile platforms, and arrays provides ocean scientists, educators, and the public with the means to collect sustained, time-series datasets to enable the examination of complex, interlinked physical, chemical, biological, and geological processes operating throughout the coastal regions and open ocean. The OOI has been built with the expectation of 25 years of operation.

The OOI implements a geographically distributed, secure, highly available CI that is responsible for data acquisition and collection, data storage and processing, and on-demand delivery of data and data products to scientists and application developers. The core of the OOI CI software ecosystem (uFrame-based OOINet) is based on service-oriented architecture, datasets, instruments, platform drivers, and data product algorithms that plug into the uFrame framework. uFrame is implemented using a combination of scalable and highly available open source distributed data management technologies and custom development.

The main data delivery in the OOI is based on an on-demand product generation in the form of synchronous pull-based data requests or asynchronous downloading of NetCDF datasets. OOI CI also provides other data delivery methods, such as a THREDDS, and a raw data server. OOI data, such as high-volume video and hydrophone data, are rapidly growing in size, and even modest queries can result in significant latencies. As a result, the current data delivery methods are not optimal for real-time processing, quality control/evaluation, event detection, and distribution of this high sample-rate data to interested scientists and organizations. The science community has developed community tools using the OOI machine-to-machine API for providing users with means to access data in real-time. These tools employ the OOI API to continuously make requests for new data with the goal of retrieving data as soon as possible, even if the data is not available in the system, which results in a large number of failed requests. Consequently, these tools require hundreds of thousands of pull-based data requests to be served daily per user, which is not a sustainable approach.

To address this problem, this science gateway leverages the OOI API for developing and deploying, in NSF-funded CI (e.g., XSEDE resources, such as Jetstream [16]), a scalable, reliable, and sustainable subscription-based mechanism for real-time streaming-based data delivery to users, which is currently not available in the OOI CI. With the extended developer support from the Science Gateway Community Institute (SGCI) [17], [18], the prototype implements the resource broker based on Apache Kafka [19], illustrated in figure 2, which delivers data streams from the OOI cabled array in near real-time and provides a Java-based interface. In our implementation, this architecture has three functions: Kafka producer, Kafka consumer, and topic management. To increase the efficiency in the application development process, we designed and implemented a JSON schema that defines the information necessary to specify scientific instruments in a message broker and to perform the data streaming process through these three functions in a straightforward way. It contains multiple instruments metadata, such as the name and code of the array and the platform, instrument, mooring, and stream type and method. The Kafka producer and consumer are used for publishing/subscribing to the targeted sensor instrument for multiple streams. The topic management shows the number of instruments registered and can create/remove the targeted topic in a message broker to a particular instrument. Further, easy-to-use front-end tools enable the end user to subscribe to OOI data and to make data delivery and format choices. This science gateway facilitates the development of workflows based on streaming data, but it requires a workflow engine system, as described in section IV.

IV. Dashboard Use Case Application

We use a dashboard use-case application to illustrate how data infrastructure building blocks, such as the VDC, can

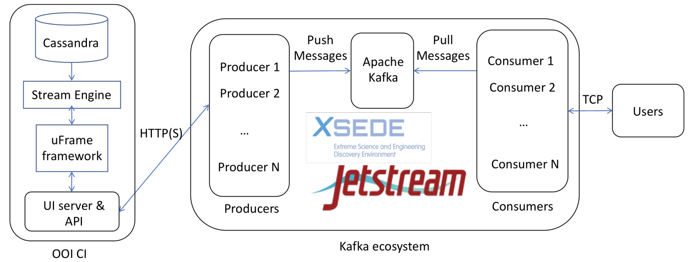

Fig. 2. Resource broker system architecture: push-based on-demand OOI data delivery through NSF-funded CI

provide the capabilities to process data from observatories in the continuum for the users. The implemented dashboard application generates and delivers data products consisting of plots of the latest available data using different granularities (customizable parameters). In this way, the users can access rich information in a timely manner without having to periodically download the data and retain the entire dataset. This use-case application is based on streamed data from the OOI, but the system architecture is designed to support other data sources and potentially combine data from different observatories. This opens the door to generating and delivering to the science community scientifically rich data products and services that are not supported or cannot be supported in a scalable manner by the large facility.

The dashboard application that is deployed at the VDC is built upon three main components: (i) the science gateway, (ii) the scientific workflow system described in section IV-A, and (iii) a graphical user interface. The science gateway is deployed with a set of agents that retrieve data from the observatory and publish them to the message broker system using Kafka topics, as illustrated in figure 2.

The scientific workflow system is subscribed to the Kafka topics and continuously produces the data products (e.g., plots for different parameters) based on the configuration parameters, such as the plot granularity and plotting frequency. When new plots are generated, they are automatically serialized and published to the message broker system using Kafka topics associated with those products. The Kafka topics and the associated data pulled from the observatory are all available for users to subscribe to. The message broker system implements the data delivery abstraction shown in figure 1.

A prototype graphical user interface is structured as a web site running on an Apache HTTP server at the VDC. Auxiliary agents are employed to subscribe to Kafka topics and obtain the latest generated plots, to be displayed on the web site. The web site contents are refreshed regularly so they provide, as a dashboard, an updated view of the plots for the different data streams and parameters. The web site also provides links to the different Kafka topics that can be used with the science gateway client software.

A. *Scientific Workflow System*

The scientific workflow system is a Python-based package focused on facilitating generic mechanisms for handling data streams from different sources and applying operations to these data as they become available. It schedules tasks according to user-defined dependencies, which can be used to implement data flows and analytics.

The workflow system provides an expandable catalog of abstractions for constructing workflows, including (1) input methods, (2) output methods, (3) data analytics, (4) task definitions, and (5) workflow definitions. These abstractions are implemented by the modules associated with them. For example, the $swinput$ module provides implementations for different input methods, such as $SWIFile$ and $SWIKafka$ for files and Kafka streams, respectively. A workflow is defined with a YAML file, which is parsed using the $SWGenerator$ class in the $swgenerator$ module. As the workflow configuration files and catalog use the same format, the workflow catalog is expanded as workflows are defined (i.e., users can deploy their own code).

V. Towards Dynamically Provisioned Federated CI

We are also developing the DyNamo system, which offers new capabilities for (a) downstream computing to process the data streams and generate higher-level science data products by utilizing science workflows and workflow management systems and (b) scalable and efficient data delivery using high-performance layer 2 network overlays. The DyNamo system allows dynamic resource provisioning on an integrated, diverse infrastructure federation, which enables on-demand access to multiple national-scale computational cloud resources (e.g., ChameleonCloud, ExoGENI racks, and Jetstream). It offers end-to-end network management for malleable data movement across infrastructures and external science data repositories and includes a network-centric platform that hides the complexity of resource and network provisioning from the scientists. The dashboard discussed above is a use-case application that can leverage such an elastic model for distributed processing and real-time push of high-resolution seafloor pressure data from OOI instruments.

VI. Conclusion

In this paper, we discuss how a science gateway can be integrated with the VDC data infrastructure building block to expand the data delivery capabilities supported by observatory systems. It provides users with push-based, real-time, and scalable access to streamed data and facilitates building data-driven workflows from single or multiple sources. It makes possible the dynamic creation of data products with rich scientific value that otherwise would require orchestrating the access and downloading of large volumes of data and significant local processing capabilities. The proposed approach also aims to connect large facilities data with NSF-funded CI, which would enable data processing (e.g., generating scientifically rich data products) in the continuum for the user, which could improve the scalability and efficiency of content delivery. Furthermore, it enables science workflows to be built upon dynamically provisioned federated CI with the DyNamo platform.

Acknowledgments

This work is supported in part by the NSF via grants numbers OAC 1640834, OAC 1835692, OCE 1745246, and OAC 1826997. The research at Rutgers was conducted as part of the Rutgers Discovery Informatics Institute (RDI^2). This work is also supported by the NSF Science Gateway Community Institute with extended developer support and used XSEDE, which is supported by the NSF.

References

[1] L. Evans and P. Bryant, "LHC machine," *Journal of Instrumentation*, 2008.

[2] Laser Interferometer Gravitational-wave Observatory (LIGO), "https://www.ligo.caltech.edu," Last accessed on March 2019.

[3] W. Gressler and et al., "LSST Telescope and site status," *Ground-based and Airborne Telescopes V*, 2014.

[4] B. Wang and et al., "Square Kilometre Array telescope - Precision reference frequency synchronisation via 1f-2f dissemination," *Scientific reports*, 2015.

[5] I. Rodero and M. Parashar, "Data cyber-infrastructure for end-to-end science: Experiences from the nsf ocean observatories initiative," *Computing in Science Engineering*, 2019.

[6] E. Deelman, T. Peterka, I. Altintas, C. D. Carothers, K. K. van Dam, K. Moreland, M. Parashar, L. Ramakrishnan, M. Taufer, and J. Vetter, "The future of scientific workflows," *Int. J. High Perform. Comput. Appl.*, vol. 32, no. 1, pp. 159–175, Jan. 2018. [Online]. Available: https://doi.org/10.1177/1094342017704893

[7] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz, "Pegasus: A framework for mapping complex scientific workflows onto distributed systems," *Sci. Program.*, vol. 13, no. 3, pp. 219–237, Jul. 2005. [Online]. Available: http://dx.doi.org/10.1155/2005/128026

[8] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, "Scientific workflow management and the kepler system: Research articles," *Concurr. Comput. : Pract. Exper.*, vol. 18, no. 10, pp. 1039–1065, Aug. 2006. [Online]. Available: http://dx.doi.org/10.1002/cpe.v18:10

[9] D. Churches, G. Gombas, A. Harrison, J. Maassen, C. Robinson, M. Shields, I. Taylor, and I. Wang, "Programming scientific and distributed workflow with triana services: Research articles," *Concurr. Comput. : Pract. Exper.*, vol. 18, no. 10, pp. 1021–1037, Aug. 2006. [Online]. Available: http://dx.doi.org/10.1002/cpe.v18:10

[10] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li, "Taverna: A tool for the composition and enactment of bioinformatics workflows," *Bioinformatics*, vol. 20, no. 17, pp. 3045–3054, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/bth361

[11] R. F. da Silva, E. Deelman, R. Filgueira, K. Vahi, M. Rynge, R. Mayani, and B. Mayer, "Automating environmental computing applications with scientific workflows," in *2016 IEEE 12th International Conference on e-Science (e-Science)*, Oct 2016, pp. 400–406.

[12] A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, M. Datar, K. Ito, R. Motwani, U. Srivastava, and J. Widom, "Stream: The stanford data stream management system," Stanford InfoLab, Technical Report 2004-20, 2004. [Online]. Available: http://ilpubs.stanford.edu:8090/641/

[13] J. Hwang, U. Cetintemel, and S. Zdonik, "Fast and highly-available stream processing over wide area networks," in *2008 IEEE 24th International Conference on Data Engineering*, April 2008, pp. 804–813.

[14] M. Parashar, V. Honavar, A. Simonet, I. Rodero, F. Ghahramani, G. Agnew, and R. Jantz, "The Virtual Data Collaboratory: a Regional Cyberinfrastructure for Collaborative Data-Driven Research," *Computing in Science Engineering*, 2019.

[15] L. M. Smith, J. A. Barth, D. S. Kelley, A. Plueddemann, I. Rodero, G. A. Ulses, M. F. Vardaro, and R. Weller, "The ocean observatories initiative," *Oceanography*, vol. 31, no. 1, pp. 16–35, 2018.

[16] R. Tudoran and et al., "Jetstream: Enabling high performance event streaming across cloud data-centers," *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, 2014.
[17] N. Wilkins-Diehr, M. Zentner, M. Pierce, M. Dahan, K. Lawrence, L. Hayden, and N. Mullinix, "The science gateways community institute at two years," in *Proceedings of the Practice and Experience on Advanced Research Computing*, ser. PEARC '18, 2018, pp. 53:1–53:8. [Online]. Available: http://doi.acm.org/10.1145/3219104.3219142
[18] N. Wilkins-Diehr and T. D. Crawford, "Nsfs inaugural software institutes: The science gateways community institute and the molecular sciences software institute," *Computing in Science Engineering*, vol. 20, no. 5, pp. 26–38, Sep. 2018.
[19] J. Kreps, L. Corp, N. Narkhede, J. Rao, and L. Corp, "Kafka: a distributed messaging system for log processing. NetDB'11," 2011.