Supporting Data-driven Workflows Enabled by Large Scale Observatories

Ali Reza Zamani, Moustafa AbdelBaky, Daniel Balouek-Thomert, Ivan Rodero, and Manish Parashar
Rutgers Discovery Informatics Institute (RDI^2), Rutgers University, Piscataway, NJ, 08854, USA
Email: {alireza.zamani, moustafa.a, daniel.balouek, irodero, parashar}@rutgers.edu

***Abstract*—Large scale observatories are shared-use resources that provide open access to data from geographically distributed sensors and instruments. This data has the potential to accelerate scientific discovery. However, seamlessly integrating the data into scientific workflows remains a challenge. In this paper, we summarize our ongoing work in supporting data-driven and data-intensive workflows and outline our vision for how these observatories can improve large-scale science. Specifically, we present programming abstractions and runtime management services to enable the automatic integration of data in scientific workflows. Further, we show how approximation techniques can be used to address network and processing variations by studying constraint limitations and their associated latencies. We use the Ocean Observatories Initiative (OOI) as a driving use case for this work.**

Keywords*—*Large scale observatories; Data-driven workflows; Wide-area data analytics; Large-scale science

I. Introduction

Large scale observatories are nationally (and sometimes internationally) funded shared-use resources designed to provide the scientific community with open access to data and data products from geographically distributed sensors and instruments. Examples of such observatories include the National Ecological Observatory Network (NEON), the Large Synoptic Survey Telescope (LSST), and the Ocean Observatories Initiative (OOI). These observatories generate vast and diverse volumes of data and data products, which can be processed by scientists globally using application workflows to accelerate scientific discovery and obtain new insights. However, seamlessly integrating this data and data products into scientific workflows presents several challenges. For example, processing large data volumes requires a lot of resources, which are typically not co-located with the observatories, and as a result, have to be transported for further processing. Moving large volumes of data over wide area networks and shared network links can be expensive, and providing end-to-end Quality of Service (QoS) guarantees can be nontrivial, often preventing timely data processing.

This paper summarizes our ongoing work in supporting data-driven and data-intensive workflows in the case of large-scale scientific observatories. Our overall vision for this work is to develop a scalable framework that meets the needs of diverse scientific workflows while satisfying performance and quality of service constraints. We use the Ocean Observatories Initiative (OOI) [1, 2] as a driving use case to design a framework that can support automated data-driven scientific workflows. We present the design of our framework, and we describe the abstractions and runtime management services provided by it. We also show how approximation techniques can be used to address network limitations and associated latencies.

The rest of the paper is organized as follows. Section II outlines the challenges associated with supporting workflows enabled by large-scale observatories. Section III introduces our framework and describes its key components. Section IV presents a case study using data processing workflows from the Oceans Observatories Initiative. Section V presents related work. Section VI concludes the paper and discusses future work.

II. Requirements and Challenges

Scientific workflows that process data from large scale observatories are typically composed of multiple steps such as data calibration, data transformation, computational modeling, analytics, visualization, and result collection. Furthermore, data dissemination from the observatories involves the end-to-end delivery of processed data from the data source(s) to one or more data consumer(s) using these workflows across a wide area environment, which is challenging for several reasons. First, transferring massive amounts of data over limited wide area network resources within prescribed time constraints can be difficult. Second, workflows involve non-trivial processing requiring significant resources, which may not be co-located with the data producer or consumer. Finally, variability in the resource availability and performance require runtime adaptation as well the use of approximations to meet time and quality constraints.

The emerging cyber-infrastructure ecosystem is becoming increasingly pervasive and integrates non-trivial resources and services along the data path, which can be leveraged to address the challenges in executing data processing workflows. For example, as illustrated in Figure 1, one can consider three classes of resources/services:

1) Edge/Fog resources: These resources are located in proximity to the data production sites. In general, edge resources can be expensive due to limited storage and processing capabilities. However, the latency between data producers and edges resources is very low.
2) In-transit resources: We consider Internet Service Provider(ISP) data centers, content distribution servers, or any resources that are located between edge and core

978-1-5386-2686-3/17 $31.00 © 2017 IEEE
DOI 10.1109/eScience.2017.95

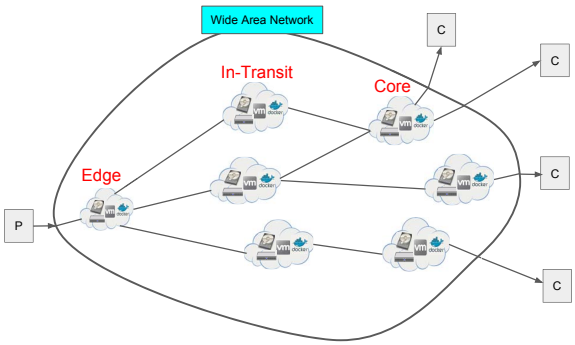

Fig. 1: Emerging cyber-infrastructure.

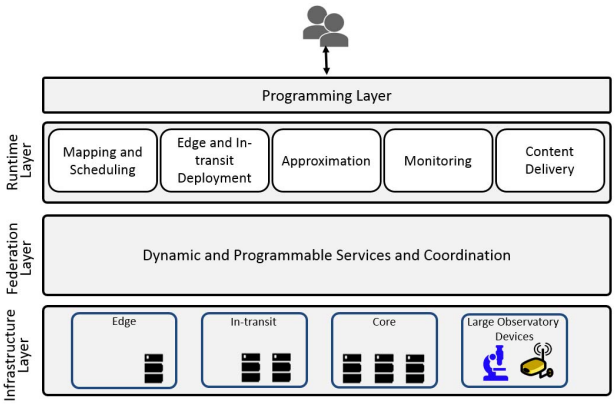

Fig. 2: The overall system architecture.

as in-transit resources. These resources are becoming increasingly available to applications and are characterized by latencies that are in between edge and core resources. Given their distributed nature along the data path, in-transit resources are suitable for intermediate storage and processing.
3) Core resources: These are the primary resources for computation and storage. They are located within well-provisioned data centers. While these resources are relatively inexpensive, they have higher latencies and moving data to/from these resources is expensive.

Although there exist powerful workflow management systems [3] such as Kepler [4], Pegasus [5], and Askalon [6], which can execute and orchestrate complex workflows across distributed resources; these systems do not leverage all the resources classes presented above. Further, these systems cannot exploit resources along the data path for in-transit data processing or create dissemination networks for processed data from sources to destinations. The framework presented in this paper complements these systems as it explores how to process data while taking resources capabilities, network conditions, and constraints from users and resource providers into consideration. Further, the presented framework considers where and how to deploy the components of a workflow according to their associated trade-offs. We also explore adaptive runtime management techniques that continuously monitor changes in the execution environment and resources and adapt accordingly. Finally, we also explore approximation techniques such as reducing data and computation or determining the best data resolution that can be delivered to the users, to ensure user QoS requirements are satisfied.

III. A Framework for Data-Driven Workflows

In this section, we present an overview of our framework for supporting data-driven workflows using large-scale observatories. The framework essentially realizes a data dissemination network that leverages heterogeneous and distributed resources at the edge, along the data path, and at the core based on their availability, location, and characteristics, to process data as required by applications workflows. The framework takes user requirements, constraints, and priorities, such as quality of result (QoR), deadline, and budget to determine when and where to deploy components of a workflow. Furthermore, the framework's adaptive runtime continuously monitors the workflow execution and adapts it to maintain the desired QoS. The runtime also uses approximate computing techniques to find trade-offs, for example, between the cost/time for computing and the quality/optimality of the solution.

A schematic overview of the architecture of our framework is presented in Figure 2. The *Infrastructure layer* consists of edge, in-transit, core resources as well as data sources such as sensors and instruments that are part of the observatories. The *Federation layer* is responsible for coordinating resources, in particular, allowing them to join and leave the federation as needed. The *Programming layer* provides an interface for the programmers and end-users to interact with the framework. It translates high-level instructions from the users to low-level instructions used by the *Runtime layer*. Constraints, priorities, workload description, and features/contents of interest are provided using the *Programming layer*.

The focus of this paper is on the *Runtime layer*, which provides the following capabilities:

1) Mapping and scheduling: Mapping workflow components onto distributed edge, in-transit, and core resources while satisfying user requirements and provider priorities and constraints (e.g., reducing energy consumption or maximizing utilization) is challenging. In our approach, we map the workflow to the distributed resources while considering the location of resources, data sources, and destinations. Further, we leverage shared in-transit resources wherever possible to optimize processing requests across multiple workflows and reduce redundant executions.
2) Edge and in-transit deployment: Edge and in-transit resources can be leveraged to execute part or all of the workflow tasks or to process data as it moves toward the destinations. The framework exploits these resources by opportunistically using them as computational and storage units along the data path.
3) Monitoring: The monitoring service provides the scheduler with information on the state of resources as well as the workflow execution status. It enables control loops to

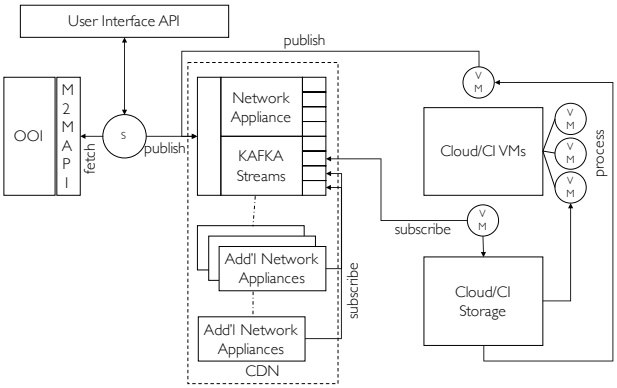

Fig. 3: An overview of the streaming-based publish/subscribe data delivery architecture.

improve performance and avoid potential execution bottlenecks (e.g., links with high packet loss or low bandwidth, computing resources with high load, etc.).
4) Approximation: Many scientific applications and workflows can tolerate errors or imprecisions, such as the energy simulation application presented in [7]. Such applications can benefit from approximation techniques that can reduce computational requirements and time-to-solution without significantly reducing the quality of results (QoR). We are exploring similar approximation techniques for data processing workflows. For example, it may be acceptable to use lossy data compression to address bandwidth limitation. As approximation techniques tend to be application-specific, we are also developing abstractions to enable users to specify approximation parameters for their applications.
5) Content delivery: We also leverage concepts from Content Distribution Networks (CDN) to disseminate data and data products from shared observatories to multiple consumers more efficiently. For example, we are developing publish/subscribe abstractions that allow users to subscribe to "features" in the data, which are extracted using in-transit data processing.

IV. Case study: OOI

The Ocean Observatories Initiative (OOI) [1, 2] currently serves data from 57 stable platforms and 31 mobile assets, carrying 1,227 instruments (∼850 deployed), providing over 25,000 science data sets and over 100,000 scientific and engineering data products. OOI raw data and data products, such as high-definition video and hydrophone data, are rapidly growing in size and even modest queries can result in significant latencies for end users and can overwhelm their local storage and computing capabilities.

Further, real-time access to such data products by the seismic and submarine volcano communities is critical for detecting volcanic eruptions, monitoring pre- and post-eruption processes, and for planning rapid responses, i.e., research cruises after event detection. This data can also be used for tsunami early detection/warning, and so it is critical that it be made available for easy access by organizations such as the Pacific Tsunami Warning Center. Therefore, the ability to store and process this type of data in real-time and push these data products to multiple users on a subscription basis quickly and efficiently is a key requirement. However, the current OOI CI infrastructure is not optimal for real-time processing, quality control/evaluation, event detection, or the distribution of this high sample rate data to interested scientists and organizations.

To better support these requirements, we developed a subscription-based data streaming service for OOI data delivery and integrated it with public cyber-infrastructure (CI) services for automated data processing. Specifically, we enabled users to create data streams based on queries, subscribe to these streams, and associate workflows with stream and stream-related events that when triggered can seamlessly orchestrate the entire data-to-discovery pipeline. Such pipelines involves (1) executing queries on the OOI CI, (2) streaming data to appropriate CI services possibly using high-bandwidth interconnects (such as Internet2), (3) staging the data close to the computing/analytics resources (e.g., XSEDE JetStream [8]), (4) launching the modeling and analysis processes to transform such data into insights, and (5) publishing the results to the user. The developed solution leverages the Apache Kafka [9] data streaming platform. The overall architecture of the solution is depicted in Figure 3.

V. Related Work

The state-of-the-art related to the presented work can be categorized into the following areas:

Wide Area Analytics: Representative solutions in this area such as WANalytics [10], Pixida [11] and Iridium [12], primarily focus on processing large amounts of data over wide area distributed resources and are complementary to our work.

Stream Processing/Analytics: Research in this area focuses on processing streams of data and events. For example, Jetstream [13] addresses analytics on wide area streams with latency bounds, whereas Heintz et al. [14] considers accuracy versus timeliness trade-off in approximate windowed group aggregation at the edge in wide area streaming applications. Our previous work [15] explored wide area data streaming for scientific workflows.

Edge Processing: This research explores the use of edge resources for all or part of the processing. For example, Nebula [16] enables voluntary resources to participate in processing workloads along with dedicated resources. Similarly, the CloneCloud system [17] enables mobile applications to take advantage of cloud environments.

In-transit Processing: This research explores how intermediate nodes can be used to partially process the data before it reaches the destination, for example, research presented in [18, 19].

Approximation: Research in this area takes advantage of approximation techniques to reduce data size and computational requirements. Example research efforts include work by Krishnan et al. [20] that introduced the notion of incremental execution where the output gets updated

for successive runs of a job, and the work by Vassiliadis et al. [21] that explored the execution of different parts of an application and their impact on accuracy and energy consumption.

The framework presented in this paper comprehensively addresses these issues to support end-to-end data-driven workflows using data from large-scale observatories. It creates a data dissemination network that leverages resources along the data path to effectively process data in-transit. It also manages data processing adaptively and leverages approximation techniques to maintain application quality of service.

VI. Conclusion

In this paper, we presented an overview of a framework for supporting data-driven and data-intensive workflows that can take advantage of data from large-scale observatories, and specifically the Ocean Observatories Initiative (OOI). The framework leverages resources/services across the data path (at the edge, in-transit, and in the core) to create a data dissemination network for end-to-end delivery of processed data from data source(s) to one or more data consumer(s) using data processing application workflows. The framework provides programming abstractions as well as adaptive runtime mechanisms that support monitoring and also dynamic mapping and scheduling across edge, in-transit, and core resources. We also discussed how approximation techniques could be used to address network and processing variations, constraints limitations, and the associated latency trade-offs. We have prototyped, deployed and evaluated some of capabilities described in Section III as presented in the following publications [7, 15, 18, 19, 22]. We are currently integrating these components into an overall framework as well as exploring appropriate programming layer abstractions.

Acknowledgements: This work is supported in part by NSF via grants numbers OCI1339036, OCI1441376, ACI1464317, ACI1640834, and by an award from Ericsson.

References

[1] Ivan Rodero and Manish Parashar. Architecting the cyberinfrastructure for National Science Foundation Ocean Observatories Initiative (OOI). In *7th International Workshop on Marine Technology: MARTECH 2016*, pages 99–101, 2016.

[2] Ocean Observatories Initiative Web Site. http://oceanobservatories.org.

[3] Ewa Deelman, Tom Peterka, Ilkay Altintas, Christopher D Carothers, Kerstin Kleese van Dam, Kenneth Moreland, Manish Parashar, Lavanya Ramakrishnan, Michela Taufer, and Jeffrey Vetter. The future of scientific workflows. *The International Journal of High Performance Computing Applications*, 2017.

[4] Ilkay Altintas, Chad Berkley, Efrat Jaeger, Matthew Jones, Bertram Ludascher, and Steve Mock. Kepler: an extensible system for design and execution of scientific workflows. In *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, pages 423–424. IEEE, 2004.

[5] Ewa Deelman, Gurmeet Singh, Mei-Hui Su, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Karan Vahi, G Bruce Berriman, John Good, et al. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, 13(3):219–237, 2005.

[6] Thomas Fahringer, Radu Prodan, Rubing Duan, Francesco Nerieri, Stefan Podlipnig, Jun Qin, Mumtaz Siddiqui, Hong-Linh Truong, Alex Villazon, and Marek Wieczorek. Askalon: A grid application development and computing environment. In *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, pages 122–131. IEEE Computer Society, 2005.

[7] Ali Reza Zamani, Ioan Petri, Javier Diaz-Montes, Omer Rana, and Manish Parashar. Edge-supported approximate analysis for long running computations. In *IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2017 – to appear.

[8] XSEDE JetStream Cloud. http://jetstream-cloud.org.

[9] Jay Kreps, Linkedin Corp, Neha Narkhede, Jun Rao, and Linkedin Corp. Kafka: a distributed messaging system for log processing. NetDB'11, 2011.

[10] Ashish Vulimiri, Carlo Curino, Brighten Godfrey, Konstantinos Karanasos, and George Varghese. Wanalytics: Analytics for a geo-distributed data-intensive world. In *CIDR*, 2015.

[11] Konstantinos Kloudas, Margarida Mamede, Nuno Preguiça, and Rodrigo Rodrigues. Pixida: optimizing data parallel jobs in wide-area data analytics. *Proceedings of the VLDB Endowment*, 9(2):72–83, 2015.

[12] Qifan Pu, Ganesh Ananthanarayanan, Peter Bodik, Srikanth Kandula, Aditya Akella, Paramvir Bahl, and Ion Stoica. Low latency geo-distributed data analytics. *ACM SIGCOMM Computer Communication Review*, 45(4):421–434, 2015.

[13] Ariel Rabkin, Matvey Arye, Siddhartha Sen, Vivek S Pai, and Michael J Freedman. Aggregation and degradation in jetstream: Streaming analytics in the wide area. In *NSDI*, volume 14, pages 275–288, 2014.

[14] Benjamin Heintz, Abhishek Chandra, and Ramesh K Sitaraman. Trading timeliness and accuracy in geo-distributed streaming analytics. In *SoCC*, pages 361–373, 2016.

[15] Viraj Bhat, Scott Klasky, Scott Atchley, Micah Beck, Douglas McCune, and Manish Parashar. High performance threaded data streaming for large scale simulations. In *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*, pages 243–250. IEEE, 2004.

[16] Albert Jonathan, Mathew Ryden, Kwangsung Oh, Abhishek Chandra, and Jon Weissman. Nebula: Distributed edge cloud for data intensive computing. *IEEE Transactions on Parallel and Distributed Systems*, 2017.

[17] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. Clonecloud: elastic execution between mobile device and cloud. In *Proceedings of the sixth conference on Computer systems*, pages 301–314. ACM, 2011.

[18] Viraj Bhat, Manish Parashar, Hua Liu, Mohit Khandekar, Nagarajan Kandasamy, and Sherif Abdelwahed. Enabling self-managing applications using model-based online control strategies. In *Autonomic Computing, 2006. ICAC'06. IEEE International Conference on*, pages 15–24. IEEE, 2006.

[19] Ali Reza Zamani, Mengsong Zou, Javier Diaz-Montes, Ioan Petri, Omer Rana, and Manish Parashar. A computational model to support in-network data analysis in federated ecosystems. *Future Generation Computer Systems*, 2017.

[20] Dhanya R Krishnan, Do Le Quoc, Pramod Bhatotia, Christof Fetzer, and Rodrigo Rodrigues. Incapprox: A data analytics system for incremental approximate computing. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1133–1144. International World Wide Web Conferences Steering Committee, 2016.

[21] Vassilis Vassiliadis, Konstantinos Parasyris, Charalambos Chalios, Christos D Antonopoulos, Spyros Lalis, Nikolaos Bellas, Hans Vandierendonck, and Dimitrios S Nikolopoulos. A programming model and runtime system for significance-aware energy-efficient computing. In *ACM SIGPLAN Notices*, volume 50, pages 275–276. ACM, 2015.

[22] Ali Reza Zamani, Mengsong Zou, Javier Diaz-Montes, Ioan Petri, Omer Rana, Ashiq Anjum, and Manish Parashar. Deadline constrained video analysis via in-transit computational environments. *IEEE Transactions on Services Computing*, 2017.