A Brief Survey of Data Streaming Technologies

Ezra Kissel, Jason Zurawski
Energy Sciences Network (ESnet)
Lawrence Berkeley National Laboratory (LBNL)
Berkeley, CA, USA
{kissel, zurawski}es.net

***Abstract*—Streaming data is data that is emitted at variable volumes in a continuous, incremental manner with the goal of low-latency processing often at a different physical location. Network infrastructure is used to facilitate the connection between data sources and sinks, and must be robust to handle the requirements of the workflow. The U.S. Department of Energy Office of Science (DOE SC) a federal agency supporting fundamental scientific research for energy and the Nation's largest supporter of basic research in the physical sciences. DOE SC has the responsibility for operating 10 National Laboratories, and 28 scientific user facilities supporting advanced supercomputers, particle accelerators, large x-ray light sources, neutron scattering sources, and other specialized facilities for nanoscience and genomics. This paper investigates the state of streaming data workfows, and details some of the approaches to this challenging problem.**

***Index Terms*—Data communication, Network protocols, Data management systems**

I. INTRODUCTION

Data has been compared to the lifeblood of organizations, as well as being as valuable as oil was to early industrial society. The ability to rapidly consume and perform analysis on both pre-exiting data, as well as those that are actively being generated, defines the success of a scientific workflow. Certain types of data pipelines may in fact rely on access to online producers of data whose output must be processed when received, and where results may similarly be quickly acted upon to meet timeliness constraints. Within this realm of online analysis, distributed data paradigms have emerged that enable computation on a critical set of time-sensitive data generated by one or more sources in a deadline-driven, deterministic manner. This type of time-constrained, low-latency processing is often referred to as data streaming [1].

Within the DOE SC user facilities complex, data streaming requirements have been driven in large part by scientific instruments that are generating such high volumes of raw observations that they exceed local processing and storage capacities. Consequently, scientific workflows have begun to concurrently make use of remote High Performance Computing (HPC) facilities to process streams of experimental events in their analysis pipelines. The distributed, time-constrained nature of this processing necessitates that event data be transferred over a high-performance network towards awaiting compute resources while the experiment is ongoing, and where online analysis has the potential to modify, or steer, the active experiment using embedded control mechanisms. Supporting data streaming applications in terms of network access and services, security policy, dynamic compute allocation, and end-to-end orchestration are all areas that will require ongoing evolution and support from facilities before any general model may be adopted.

Early efforts to design and implement new data streaming services and architectures have begun to seize this open opportunity. The Energy Sciences Network (ESnet), the network provider for DOE SC Laboratories and user facilities, has both initiated and integrated with data streaming projects. The goal in this work is to survey a subset of these efforts and provide a better understanding of the current characteristics, requirements, and technological solutions that are driving this shift in data delivery and processing. which in turn may help inform network infrastructure operators and the data producers and consumers alike.

II. BACKGROUND

In contrast to data streaming, the DOE Advanced Scientific Computing Research (ASCR) program has been successful in addressing many of the challenges around file-based data movement. The ScienceDMZ model [2], including the adoption of Data Transfer Nodes (DTNs), has enabled the large-scale distribution of data sets between science and HPC facilities and provides a pattern for reproducible deployment. There has been a similar convergence around patterns for high-performance storage [3]–[5] and wide-area network provisioning systems [6], [7] that integrate tightly with local compute and DTN resources. Unfortunately, file transfer is not a panacea and there are shortcomings for certain use cases that work with time-sensitive data.

More broadly, the focus on HPC facilities to scale in terms of processing and storage has increasingly had to share consideration with the inclusion of enhanced data delivery architectures, which allow for the direct, end-to-end transfer of data over remotely connected, and often heterogeneous, compute and network infrastructure. The distributed nature of such architectures points to the network taking a more active role in realizing any deployed approach. However, what additional capabilities (e.g., in-network data services) the network should provide, and where (Core vs Edge or some combination), in supporting data streaming applications remains an open question. Compounding both technical and sociological issues is the fact that unlike the de facto standards and models around file-based data delivery, the solution space, and even definition, of data streaming for DOE-supported experimental science is not well-defined and remains in a prototyping phase.

A DOE report [8] defines a data stream as ”an unbounded sequence of events that needs reactive real-time or near real-time processing and can possibly have multiple consumers and producers.” Streams do not support random access, i.e., arbitrary forward or backward access to data, and stream data must be processed as it arrives, thus cannot benefit from global meta-data awareness. Steering was defined as “the ability to dynamically control the progression of a computational process to enable decision support”. Steering refers to the real-time adjustment of simulations, sensors, or autonomous vehicles, often in conjunction with streaming data. However, streaming and steering don’t always occur together, as seen in cases where two coupled codes, like fluid dynamics and structural solvers, interact without simultaneous streaming [9].

The need to develop capabilities around data streaming and steering has been captured in the recent Integrated Research Infrastructure Advanced Blueprint Activity (IRI ABA). In particular, the IRI ABA final report [10] highlights a Time-sensitive architectural pattern that motivates development in this area. The Time-sensitive pattern calls out urgency as a defining characteristic ”requiring real-time or end-to-end performance with high reliability, e.g., for timely decision-making, experiment steering, and virtual proximity.” Given the momentum behind IRI activities, HPC facilities have begun evolving their own architectural designs that include capabilities for supporting streaming workflows [11]. Additional work in [12] was used to categorize a number of existing scientific workflows that spanned multiple facilities, highlighting how all three IRI architecture patterns are being used in practice across the DOE SC user facility complex to accomplish modern team science.

III. RELATED WORK

The authors acknowledge that reviewing the current state of topic well explored. In [13] Isah et al. explored a comparative study of distributed data stream processing and analytics frameworks. In [14] Khan et al. performed an empirical study on widely used data streaming technologies and serialization protocols. This work will explore a number of general-purpose streaming frameworks developed in both the open source and commercial spaces.

These frameworks include Kafka , Nanomsg, ZeroMQ, MQTT, RabbitMQ, and NATS, to name only a few [15]–[20]. Such frameworks support various communication patterns like publication/submission (pub/sub), pipeline, and request/reply that form the basis for many data streaming workflows. Additionally, they offer proven, scalable implementations that offer fault tolerance and a number of integration opportunities through robust programming language and library support. One major challenge is in the complexity of setup and maintenance of such frameworks, which often requires a deep understanding of their architectures and operations to effectively deploy and integrate.

There are also a number of frameworks supporting data streaming associated with a number of DOE facilities that are in various stages of development. These include EPICS, SciStream, EJFAT, and DELERIA [21]–[24]. Such streaming frameworks are typically not as generalized as the examples given above; rather they are more focused on supporting a specific type of workflow with well-defined orchestration needs or particular applications within a given science domain that have performance or access requirements. Additionally, such frameworks may incorporate existing streaming framework solutions such as Kafka or Nanomsg to provide network communication infrastructure or they may implement their own. Purpose-built streaming efforts often provide a more tailored set of interfaces designed to meet the needs of experimentalists, and hide the underlying network and orchestration complexities.

Finally, at the organizational level, there are a number of DOE facility and DOE-collaborative projects that are invested in some combination of developing and employing streaming workflow capabilities. These include the Advanced Photon Source (APS) at Argonne National Laboratory (ANL), the Advanced Light Source (ALS) and National Center for Electron Microscopy (NCEM) at Lawrence Berkeley National Laboratory (LBNL), the DIII-D National Fusion Facility, the Facility for Rare Isotope Beams (FRIB) GRETA project at Michigan State University, the ATLAS and CMS collaborations from the Large Hardron Collider (LHC), the Electron-Ion Collider (EIC) at Brookhaven National Laboratory (BNL), and the Linac Coherent Light Source (LCLS) at SLAC. Ongoing nuclear fusion projects such as International Thermonuclear Experimental Reactor (ITER) are also anticipated to depend heavily on distributed data analysis made possible by high-performance data streaming.

IV. CHARACTERISTICS OF DATA STREAMING OVER A NETWORK

Given the variety of data streaming use cases and possible technology solutions, we offer a definition of data streaming for the reader that attempts to narrow the context for the following archetype definitions. This definition is scoped to data streams that traverse a network, as opposed to data streams that may occur within a particular compute system. As a disclaimer, the following characterization should not be considered applicable to all data streaming use cases, but rather used as a helpful guide when considering data streaming in relation to other data movement paradigms such as file transfer. The key characteristics of data streaming are:

- C1: Data delivery between a producer and consumer across a network
- C2: There can be multiple producers and consumers with different endpoints
- C3: At least one endpoint must involve volatile memory
- C4: A control component to create and manage a data pipeline
- C5: Active streaming has a predetermined duration over which the data pipeline is active

C1 necessitates the involvement of a networking component and does not consider purely local inter-process or intra-node use cases. The network itself may be from the data center to the wide area in scale. C1 also considers bi-directional flow of information between producers and consumers and does not make any assumptions as to the initiator of data transport connections.

C2 expresses that data streaming workflows often involve more than single, point-to-point connections. Parallel, incast (many-to-one), and out-cast (one-to-many) connectivity patterns across multiple producer and consumer endpoints are common and must be considered in the supporting network design.

C3 ensures there is a timeliness component to the data stream, and implies some active processing at either the producer or consumer. This characteristic separates data streaming from variations on file-based, storage-to-storage data movement. C3 also highlights the potential need of the network to ensure some level of quality-of-service to ensure the streaming solution can deliver data with reliability guarantees and minimal buffering, reducing latency of the processing pipeline.

C4 necessitates some level of configuration to enable network connectivity between producer and consumer endpoints. Producers and consumers are not assumed to be persistently reachable over a dataplane connection and thus a level of dynamic control is required. Composing services between the endpoints forms a ”data pipeline” that supports the eventual data stream. Such control may also involve changing the network's service level expectations (SLAs) and steering of the data stream connections over the course of a workflow lifetime.

C5 specifies that data streams are not unbounded under this definition and often have a timeliness quality, which may be driven by experimentation or application constraints.The deadline-driven nature of many data streaming use cases require active network connectivity that will have a specific termination date, although that date may be extended as needed.

V. DATA STREAMING SOLUTION ARCHETYPES

There are several data streaming systems under development and undergoing active prototyping that fall within the data streaming definition described above. It has also become apparent that there are a number of different architectural approaches being investigated. Understanding the specific problems being tackled, and the technical solutions being developed within each system, may aid infrastructure operators when providing services and guidance when data streaming applications are deployed. We identify three main archetypes below.

A. End-to-end

The real-time nature of stream processing requires acquiring control of instruments, compute and network; however, a stream of data on its own lacks the mechanisms to decide how to orchestrate these resources on behalf of the application. End-to-end architectures take an active role in orchestrating the entire data streaming workflow, from management of data sources, compute and network allocation, application execution, streaming data transport, and steering feedback support.

End-to-end streaming solutions may be more tightly integrated with a particular instrument or application(s) as the orchestration mechanisms would require a closer coupling with the specifics of the local site interfaces. However, generalized components and methodologies developed as part of the end-to-end solution may find use in other contexts, particularly if the end-to-end design is modular and composable.

B. Network transport

Network transport solutions focus on streaming data delivery over the network, and the control and management of data sources are out of scope. Orchestration in this archetype is limited to the network-based services that provide connectivity and SLAs (e.g., quality-of-service) between endpoints. Such orchestration may involve data-plane setup and active steering of the streaming flows over time.

Network transport approaches may provide companion software and tooling that allows applications to marshall, transpose, and direct source data into the transport data plane. The transport implementation itself may also provide in-network capabilities such as caching for redundancy and failover, load-balancing, and protocol translation/boosting. Providing such features typically necessitates some application-specific knowledge be encoded into the data stream via protocol encapsulation, or otherwise specified using a dedicated control plane.

C. Middlebox

One of the key challenges in supporting online streaming applications is enabling connectivity between endpoints across potentially multiple unique network domains. The middlebox archetype is a solution that makes use of network forwarding components (e.g., proxies) to assist in navigating different administrative and technological network segments, and in particular, between public-private address spaces by providing a network-address (NAT) translation capability.

Middlebox proxies may be software-router based (running on standard servers and network interfaces) or may be hardware-assisted (relying on SmartNICs or other programmable network devices). Beyond providing NAT functions, they may allow flexibility in the forwarding layer (L3/L4/L7) with support for various application-specific interfaces. Middlebox implementation will typically include a control plane that supports the instantiation of authenticated configuration at the per-flow or end-to-end path level.

The middlebox archetype has the potential to offer the most flexibility given its realization as a solution that runs on standard server hardware. This flexibility also presents some potential downsides in terms of achievable forwarding performance and consistency, complexity in securing deployed proxies, and maintaining application expectations around the end-to-end principle.

VI. DATA STREAMING PROJECTS IN USE AT ESNET

Several data streaming solutions have been under investigation at ESnet, each having unique capabilities and targeted use cases. Table I summarizes each project by its key features and categorizes them by the previously defined data streaming archetypes. A brief description of each project follows.

The ESnet-JLab FPGA Accelerated Transport, or EJFAT (pronounced "Edge-Fat") [23] is a collaboration between ESnet and the Thomas Jefferson National Accelerator Facility (Jefferson Lab). The EJFAT prototype is designed to seamlessly integrate edge and cluster computing in order to allow data from multiple types of scientific instruments to be streamed and processed in near real time by multiple HPC facilities — and, if needed, to redirect those data streams dynamically.

DELERIA [24] is a software suite designed to generalize and scale-up the FRIB GRETA instrument's data streaming pipeline. The DELERIA software supports the end-to-end orchestration of the data pipeline, from the instruments, compute, and networking across multiple facilities. The effort represents co-development between ESnet, Nuclear Physics at LBNL, and ORNL for network testbed deployment.

SciStream [22], [25] is a project from ANL that bills itself as an architecture and toolkit for data streaming between federated science instruments. It provides a middlebox solution for memory-to-memory streams and includes a control protocol with Globus Authentication support.

The Extensible Internet Block Streaming Service (EI: BSS) is the outcome of an ongoing research effort within University of California Berkeley ICSI called Extensible Internet [26], [27] to develop a block-oriented streaming solution for time-intensive projects.

Apollo Streams is derived from the Phoebus project [28], a middlebox architecture for WAN acceleration with network service on-ramp capabilities. Apollo enables memory-to-memory streaming applications to proxy their transport flows through Apollo Gateways where per-segment protocol optimization or translation may take place.

VII. CONCLUSIONS

This work can be summarized as follows:

- Data streaming does not fundamentally require that the network be managed other than having reasonable expectations around packet delivery. Best-effort packet forwarding, at whatever connection rates are available, may be suitable for a significant subset of streaming workflows.
- Beyond best-effort delivery, the level of networking support may be dependent on the complexity of the streaming solution. Is network visibility and control required for accurate steering? Are there performance and access requirements? Where are the network boundaries between producer and consumer? Is there an expectation for a particular QoS SLA?
- A key challenge in data streaming between HPC facilities is how to navigate network connectivity across active compute elements. Ethernet convergence has simplified part of the equation, but enabling access to local interconnects, and providing a variety of consistent, end-to-end network service types (e.g. L2VPN, L3VPN) are active areas of development.
- What should the availability and role of networking configurability be in any end-to-end orchestration streaming solution? Is it possible to define a set of network primitives that would form the basis of a network facility API?
- Network transport solutions have an opportunity to introduce bespoke capabilities, often relying on in-network hardware/service deployments. Will such capabilities also be generalizable or are they best served through tightly coupled integration with data producers and consumers?
- Proxy solutions offer a path to streaming adoption using existing, well-understood, ScienceDMZ buildouts. AuthN/AuthZ integration and performance scaling might slow deployment opportunities in the future. Middlebox gateways may also be hosted in-network where they could provide services at the customer edge.
- There is a common pattern that emerges in each data streaming solution: establish a data plane for the workflow given some desired SLA, make use of an existing control plane to establish the data plane, and manage its behavior for the duration of the data streaming workflow

REFERENCES

[1] A. Margara and T. Rabl, *Definition of Data Streams*. Cham: Springer International Publishing, 2019, pp. 648–652. [Online]. Available: https://doi.org/10.1007/978-3-319-77525-8_188

[2] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski, "The science dmz: a network design pattern for data-intensive science," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '13. New York, NY, USA: Association for Computing Machinery, 2013. [Online]. Available: https://doi.org/10.1145/2503210.2503245

[3] A. Bashyal, P. V. Gemmeren, S. Sehrish, K. Knoepfel, S. Byna, and Q. Kang, "Data storage for hep experiments in the era of high-performance computing," 2022. [Online]. Available: https://arxiv.org/abs/2203.07885

[4] A. Dorigo, P. Elmer, F. Furano, and A. Hanushevsky, "Xrootd/txnetfile: a highly scalable architecture for data access in the root environment," in *Proceedings of the 4th WSEAS International Conference on Telecommunications and Informatics*, ser. TELE-INFO'05. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2005.

[5] P. Braam, "The lustre storage architecture," *CoRR*, vol. abs/1903.01955, 2019. [Online]. Available: http://arxiv.org/abs/1903.01955

[6] M. Balman, E. Chaniotakis, A. Shoshani, and A. Sim, "Advance network reservation and provisioning for science." Lawrence Berkeley National Lab. (LBNL), Berkeley, CA (United States), 07 2009. [Online]. Available: https://www.osti.gov/biblio/985919

[7] I. Monga, C. Guok, J. MacAuley, A. Sim, H. Newman, J. Balcas, P. DeMar, L. Winkler, T. Lehman, and X. Yang, "Software-defined network for end-to-end networked science at the exascale," *Future Generation Computer Systems*, vol. 110, p. 181–201, Sep. 2020. [Online]. Available: http://dx.doi.org/10.1016/j.future.2020.04.018

[8] G. Fox, S. Jha, and L. Ramakrishnan, "Stream2016: Streaming requirements, experience, applications and middleware workshop," 10 2016. [Online]. Available: https://www.osti.gov/biblio/1344785

[9] E. Young, X. He, R. King, and D. Corbus, "A fluid-structure interaction solver for investigating torsional galloping in solar-tracking photovoltaic panel arrays," *Journal of Renewable and Sustainable Energy*, vol. 12, no. 6, 11 2020.

TABLE I
ACTIVE DATA STREAMING PROJECTS UNDER INVESTIGATION AT ESNET.

Name	Archetype	Key features
EJFAT	Network Transport	• Real time load balancing of DAQ streams into compute • Stateless load balancing, enabling horizontal scale through multiple parallel FPGA cards • Virtualized load balancers to run multiple workflows on a single physical load balance cluster • Multi facility support for parallel streaming to more than one HPC center • Simple shim library (E2SAR) for adapting user's data streams to EJFAT • WAN QoS baked into the protocol, transparent to the user. • No Proxy or software buffers, enabling multi terabit streaming over the WAN
DELERIA	End-to-end	• Software solution supporting the entire data processing pipeline of instruments streaming data. • Allows for online data analysis and monitoring • Support high latency networks • Accepts ingest of data using UDP • Composable, extensible
SciStream	Middlebox	• Middleware solution for supporting memory-to-memory data streaming • Uses an out-of-band control plane to manage deployment of forwarding proxies • Control and data plane may be authenticated across multiple security domains, including using Globus Auth • Control plane remains agnostic to the underlying data plane proxy deployed
Extnsible Internet: Block Streaming Service	Network Transport	• Managed Publish/Subscribe data distribution paradigm using application-defined ”blocks” • Supports in-network caching of blocks at designated Service Nodes (SNs) within topology • Data consumers explicitly ACK blocks when processed while caches perform accounting to apply eviction strategy • Robust under data consumer failure conditions
Apollo Streams	Middlebox	• A deployable software router (gateway) for proxying transport connections • Uses a session protocol to control gateway and forwarding behavior for one or multiple flows between endpoints • Session protocol is both in-band and out-of-band while forwarding backends are developed within the middlebox framework. • Supports protocol translation and network circuit onramp with buffering • Control and data authentication with X.509 or ssh certificates

[10] W. L. Miller, D. Bard, A. Boehnlein, K. Fagnan, C. Guok, E. Lançon, S. Ramprakash, M. Shankar, N. Schwarz, and B. L. Brown, “Integrated research infrastructure architecture blueprint activity (final report 2023),” US Department of Energy (USDOE), Washington, DC (United States). Office of Science; Lawrence Berkeley National Laboratory (LBNL), Berkeley, CA (United States), Tech. Rep., 07 2023. [Online]. Available: https://www.osti.gov/biblio/1984466

[11] M. J. Brim, A. George, A. Shehata, C. Lester, D. Rogers, P. Widener, R. Miller, G. Jansen, R. Ferreira Da Silva, and S. Oral, “A high-level design for bidirectional data streaming to high-performance computing systems from external science facilities,” Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), Tech. Rep., 03 2024. [Online]. Available: https://www.osti.gov/biblio/2338264

[12] 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,” Lawrence Berkeley National Laboratory (LBNL), Berkeley, CA (United States), Tech. Rep., 11 2023. [Online]. Available: https://www.osti.gov/biblio/2205078

[13] H. Isah, T. Abughofa, S. Mahfuz, D. Ajerla, F. Zulkernine, and S. Khan, “A survey of distributed data stream processing frameworks,” *IEEE Access*, vol. 7, pp. 154 300–154 316, 2019.

[14] S. Jackson, N. Cummings, and S. Khan, “Streaming technologies and serialization protocols: Empirical performance analysis,” 2024. [Online]. Available: https://arxiv.org/abs/2407.13494

[15] J. Kreps, “Kafka : a distributed messaging system for log processing,” 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:18534081

[16] “nanomsg.” [Online]. Available: https://nanomsg.org/

[17] “Zeromq.” [Online]. Available: https://zeromq.org/

[18] “Mqtt.” [Online]. Available: https://mqtt.org/

[19] “Rabbitmq.” [Online]. Available: https://www.rabbitmq.com/

[20] “Nats.” [Online]. Available: https://nats.io/

[21] L. R. Dalesio, A. J. Kozubal, and M. R. Kraimer, “Epics architecture.” Los Alamos National Lab., NM (United States), 01 1991. [Online]. Available: https://www.osti.gov/biblio/6110347

[22] J. Chung, W. Zacherek, A. Wisniewski, Z. Liu, T. Bicer, R. Kettimuthu, and I. Foster, “SciStream: Architecture and Toolkit for Data Streaming between Federated Science Instruments,” in *Proceedings of the 31st International Symposium on High-Performance Parallel and Distributed Computing*, ser. HPDC ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 185–198. [Online]. Available: https://doi.org/10.1145/3502181.3531475

[23] S. Sheldon, Y. Kumar, M. Goodrich, and G. Heyes, “EJ-FAT Joint ESnet JLab FPGA Accelerated Transport Load Balancer,” in *2022 IEEE/ACM International Workshop on Innovating the Network for Data-Intensive Science (INDIS)*, 2022, pp. 32–40.

[24] Cromaz, Mario, Dart, Eli, Pouyoul, Eric, and Jansen, Gustav R., “Simple and Scalable Streaming: The GRETA Data Pipeline,” *EPJ Web Conf.*, vol. 251, p. 04018, 2021. [Online]. Available: https://doi.org/10.1051/epjconf/202125104018

[25] C. Qu, J. Chung, Z. Liu, T. Bicer, and R. Kettimuthu, “Evaluating scistream (federated scientific data streaming architecture) on fabric,” in *2022 IEEE/ACM International Workshop on Innovating the Network for Data-Intensive Science (INDIS)*, 2022, pp. 25–31.

[26] H. Balakrishnan, S. Banerjee, I. Cidon, D. Culler, D. Estrin, E. Katz-Bassett, A. Krishnamurthy, M. McCauley, N. McKeown, A. Panda, S. Ratnasamy, J. Rexford, M. Schapira, S. Shenker, I. Stoica, D. Tennenhouse, A. Vahdat, and E. Zegura, “Revitalizing the public internet by making it extensible,” *SIGCOMM Comput. Commun. Rev.*, vol. 51, no. 2, p. 18–24, May 2021. [Online]. Available: https://doi.org/10.1145/3464994.3464998

[27] L. Brown, E. Marx, D. Bali, E. Amaro, D. Sur, E. Kissel, I. Monga, E. Katz-Bassett, A. Krishnamurthy, J. McCauley, T. Narechania, A. Panda, and S. Shenker, “An architecture for edge networking services,” in *Proceedings of the ACM SIGCOMM 2024 Conference*, ser. ACM SIGCOMM ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 645–660. [Online]. Available: https://doi.org/10.1145/3651890.3672261

[28] E. Kissel, M. Swany, and A. Brown, “Phoebus: A system for high throughput data movement,” *Journal of Parallel and Distributed Computing*, vol. 71, no. 2, pp. 266–279, 2011.