ESnet / JLab FPGA Accelerated Transport

Michael Goodrich
Carl Timmer, Vardan Gyurjyan
David Lawrence , Graham Hayes
Yatish Kumar , Stacey Sheldon

***Abstract*—To increase the science rate for high data rates/volumes, Thomas Jefferson National Accelerator Facility (JLab) has partnered with Energy Sciences Network (ESnet) to define an edge to data center traffic shaping / steering transport capability featuring data event aware network shaping and forwarding. The keystone of this ESnet+JLab FPGA Accelerated Transport (EJFAT) is the joint development of an AI/ML directed dynamic compute work Load Balancer (LB) of UDP streamed data. The LB is a suite consisting of a Field Programmable Gate Array (FPGA) executing the dynamically configurable, low fixed latency LB *data plane* featuring real-time packet redirection and high throughput, and a *control plane* running on the FPGA host computer that monitors network and compute farm telemetry in order to make dynamic AI/ML guided decisions for destination compute host redirection / load balancing and destination resource provisioning. The LB provides for three-tier horizontal scaling across LB suites, core compute hosts, and CPUs within a host. The LB effectively provides seamless integration of edge / core computing to support direct experimental data processing for immediate use by JLab science programs and others such as the EIC as well as data centers of the future requiring high throughput and low latency for both hot and cooled data for both running experiment data acquisition systems and data center use cases.**

***Index Terms*—Data Acquisition Systems, Streaming Readout, FPGA, Network Acceleration, Load Balancing, AI/ML**

I. Introduction

IN operation since about 1995, the Thomas Jefferson National Accelerator Facility's (JLab) primary mission is to study the internal structure of the atomic nucleus using a beam of electrons from the Continuous Electron Beam Accelerator Facility (CEBAF) located on the same campus. In the first 10 or so years the beam energy was 6 GeV and was updated to 12 GeV in the early 2010's.

Nuclear Physics (NP) experiments take place in one of four areas using the same beam concurrently. Currently, JLab's GLUEX experiment is the most demanding in terms of data rate at 3 GB/s, where data is sorted into files limited to 20GB and analyzable in approximately 3 to 4 hours. Typically a new file is opened every 7 seconds in round the clock 8 hr shifts with one day of maintenance each week. Data is taken in "runs" lasting from minutes up to a whole shift yielding 50 thousand files per week. These files are currently staged close to the detector then transferred to the JLab data center to be archived before processing. Within 5 years JLab expects all experiments to be taking data at similar rates.

Over the last year the cluster has grown rapidly to support 12 GeV where resources are "impedance matched" to the accelerator schedule and detector needs. In 2017, JLab adopted a design criteria to support the 12 GeV experimental program that greatly facilitated reduced time to science, improved the service delivery of the onsite resources, and embraced a distributed computing model via the Open Science Grid (OSG). The intent was to integrate existing resources in preference to creating new ones, thereby creating a standard environment aligned with the larger Large Hadron Collider (LHC) community.

A. Challenges being addressed

Detectors becoming more complex and increased beam intensities lead to higher event rates and more complex events that are harder to process. Data driven workflows can be bursty in nature and are typically provisioned for according to the average not peak data rates. Complex detectors are harder to monitor for anomalous behavior. Detector calibration needs to be timely otherwise slow response times for anomaly detection and experiment steering slow down the time between data taking and science results.

In 2018, JLab articulated a 'Grand Challenge' in readout and analysis with focus areas in

- Streaming readout
- Calibration/ML
- Distributed Computing
- Heterogeneous Computing
- Statistical Methods

Several workshops initiated by Advanced Scientific Computing Research (ASCR) and the Office of Science have highlighted a community need for edge computing, close to the experiment, that is also indirectly coupled to a large, centralized compute resource. The benefits of such an arrangement are many but include the ability to process data from an experiment as it is acquired. This would reduce the

Submitted for review August 22, 2022. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research Program under Grant DE-FOA-0002414. Notice: This manuscript has been authored by Jefferson Science Associates, LLC under Contract No. DE-AC05-06OR23177 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. Authors: M. S. Goodrich is with the Thomas Jefferson National Accelerator Facility, Newport News, VA 23606 USA (e-mail: goodrich@jlab.org).

C. A. Timmer, V. H. Gyurjyan, D. Lawrence, and G. Heyes are also with Thomas Jefferson National Accelerator Facility (e-mail: timmer@jlab.org, gurjyan@jlab.org, davidl@jlab.org, heyes@jlab.org).

Y. Kumar and S. Sheldon are both with the U.S. Department of Energy's Office of Science Energy Sciences Network (e-mail: yak@es.net, stac@es.net).

data volume that is archived and provide feedback to the experimenter for experiment steering. Use of a larger compute resource than would typically be available to an individual experiment would allow implementation of digital twinning, where a real and simulated experiment run in tandem in real time.

At Jefferson Lab there is an existing Nuclear Physics (NP) program using the 12 GeV CEBAF electron accelerator. Two of the detectors, CLAS12 and GLUEX, already generate data at rates close to or above 1 GB/s. In the continuation of the program, the MOLLER and SoLID detectors are expected to operate at similar or higher rates. The Electron Ion Collider (EIC) is a joint project between Brookhaven National Laboratory (BNL) and JLab to construct a new nuclear physics facility on the BNL site. As well as the EIC science aspects, JLab will contribute to the EIC project in several other areas such as detector development, electronics, data acquisition, and computing.

There is already much interest in the NP data acquisition community in transitioning from traditional triggered readout to a streaming mode where data is continuously read from the detector without a complex hardware trigger. In this mode the data rate off the detector can be significantly higher than in a traditional triggered system but the data is read out in parallel data streams each of which is manageable by contemporary hardware. A streaming system relies on edge computing to reduce the data volume after it is acquired. The dataset is then transported to a larger computing resource for further processing. The challenge is to integrate edge computing with this larger computing resource in a seamless way that will allow real or near-real time processing to provide feedback for experiment steering.

If this can be achieved, rather than associate significant edge resources to each experiment, more modest edge systems can be deployed that are backed by the larger compute resource. The availability of this central resource would allow the support of analysis methods that are not practical within the computing resource constraints of systems deployed at the edge. Reduction of data volumes by real time processing, and digital twinning have already been mentioned but the availability of a heterogeneous core computing platform could also support novel data processing methodologies, for example examination of datasets using AI/ML and comparison with existing datasets. This would benefit the JLab 12 GeV program and answer questions about how the EIC computing model will operate when the data source at BNL is geographically separate from JLab. Such technologies are also highly applicable to other science domains.

The 2019 ESnet NP requirements review discusses and analyzes current and planned use cases by science programs to inform ESnet's strategic planning. From the findings of the review it is clear that coupling of edge to supporting computational resources, as described earlier in this document, requires integration with the networking infrastructure between them. Since the computational resource will already support AI/ML as a means of simulation and data processing it is a natural extension to apply the same technology to enhance computational, networking, and storage workflows. This could be achieved by combining ESnet 6 telemetry data with telemetry data from the edge and central data centers as inputs to AI/ML to steer data and computational workloads efficiently. A finding of the ESnet report is that there is interest from scientific communities for work in this area. The recommendations and action items from the report point to a closer collaboration between ASCR, ESnet, and experiments to develop these capabilities.

B. A DOE integrated research ecosystem

The DOE has a vision for an infrastructure that transforms science via seamless interoperability. A key component will be facilities that are better suited to data driven workflows. *What are the drivers for a new type of facility*?

- Support for time critical use patterns
- Experiment steering
- Data driven workflows to support filtering, calibration, analysis, and other computational processing of data from experiments across a broad range of science programs
- Well defined quality of service that researchers can rely on while running
- Provide a range of heterogeneous computing technologies
- Be a key component of a distributed data storage infrastructure
- Allowing cross cutting research that accesses data from several sources

C. A paradigm shift to Streaming Readout

Data acquisition is currently based on a legacy readout model. A subset of signals form a trigger using custom hardware and firmware that read out only signals in the trigger window. This model is breaking down as rates and detector complexity increase. Alternatively, *Streaming Readout* continuously reads all channels that have data and relies on availability of transient data storage and high throughput processing to filter data in software as it is taken.

D. Technical Challenge - Data transport

Success of streaming readout relies on availability of a large compute resource with additional challenges in that the resource is some distance from the data source. This is a much more complex environment than a counting house and must deal with contention for resources, maintenance, and detection of anomalies. Also required is reliable, high bandwidth, data transport that can adapt to changing conditions in the data center. A key question emerges:

How do we migrate a workflow from small compute systems close to the detector to a data center when 24/7 reliability is required?

II. EJFAT Architecture - A way forward

An architecture - the ESnet JLab FPGA Accelerated Transport (EJFAT) - is being developed to answer this question. This features FPGA based *acceleration* to

- Compress, segment and prepare the data at the source

- Dynamically load balance incoming streams of data into a cluster via in-flight destination redirection
- Decompress, reassemble, and post process data near the cluster

FPGA accelerated network switches and Network Interface Cards (NIC) are possible solutions for some of these accelerations. This architecture also leverages previously successful streaming readout setups from detectors at JLab and DESY and is using streaming readout and is investigating how to put this into production in the 12 GeV program as well as application to the EIC.

Electronics attached to scientific instruments digitize measurements. This digitized data could be packaged with metadata headers e.g., where and when the measurement was made and other markers to specify its down-stream disposition, then transmitted over the network. Data processing in the data center must then keep up with the flow requiring dynamic monitoring and allocation of resources by a supervisory agent. Additionally, the destination network addresses in the data center are necessarily opaque to the sender and could be brokered by a proxy device - e.g., a suitably programmed FPGA - that functions as a single point of contact decoupling the data generation network from the data consumption network.

Further, bandwidth bottleneck challenges indicate dynamic allocation and utilization of resources in the data center. Use of the highest possible bandwidth indicates a handshake-less protocol, e.g., UDP that is however susceptible to data loss. Delivery must be guaranteed and without backpressure since it is unacceptable to tell an instrument to “slow down”.

A solution of this type would be interposed between generator and consumer networks in the manner illustrated in Figure 1, where instead of a gateway server, which could introduce a bottleneck, a better solution is a hardware device such as an FPGA based load balancer that does not process or buffer data but redirects it in flight. It does this by modifying network packet headers in-situ with minimal latency and interval such that it operates transparently at network bandwidth with no bottlenecks and also decouples the networking infrastructures of edge and core.

The primary technique for this in-flight redirection is to add metadata at the sending end for the load balancer for intelligent and ecosystem aware disposition. While this could be done in software a better solution would be to use an additional data-shaping FPGA at the edge.

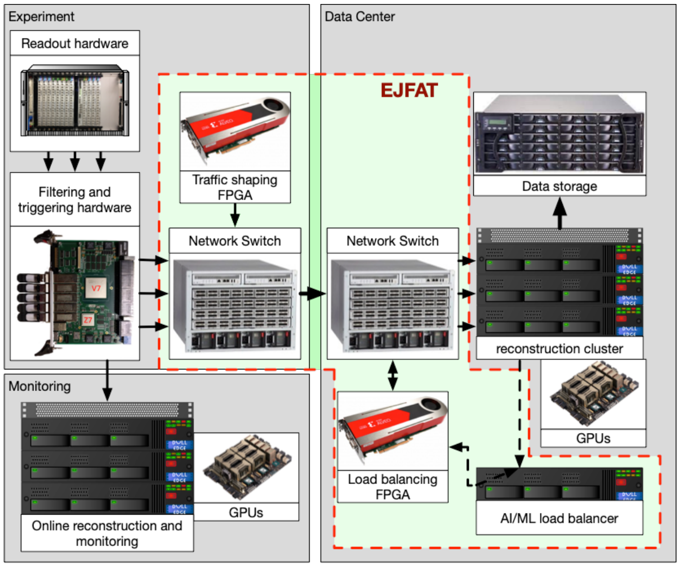

Fig. 1. EJFAT Architectural Concept

A. Approach

What is sought is an adaptive generic hardware design that is data centric and can dynamically create a computing solution tailored to the use case. The concept is a mix of CPU, GPU, FPGA, and other hardware accelerators that can be disaggregated and reassembled in the desired configuration using a high bandwidth low latency network. This is aligned with the roadmaps of several vendors, for example NVIDIA’s DGX SuperPOD. The key to this design is integration of EJFAT technology at the facility boundary and within the “machine”. Remote sites will have instruments and local “edge computing” that perform tasks that are very tightly coupled with the hardware. Via the integrated infrastructure they will have access to a much richer set of resources and services at a remote facility. JLab has worked closely with ESnet on the EJFAT project which they see as a “game changer” for high throughput data transport. JLab will continue that partnership with EJFAT-II focusing on robust, lossless, high bandwidth data transport between facilities. JLab’s success with distributed computing as well as the commitment of ESnet to EJFAT has accelerated the ESnet6 install at JLab that will soon address future high performance data center bandwidth needs. JLab has a pilot project to use EJFAT onsite to transfer data from one of the detectors to the local data center, dynamically matching resource allocation to need.

B. Load Balancing Operations

A key component of the EJFAT architecture is the Load Balancer (LB) FPGA/Host suite that is the bridging element between the edge and the compute core computing facilities. The LB FPGA is currently a Xilinx U280 FPGA PCIe card with 2 × 100 Gb/s optical ports using firmware developed by ESnet from specifications developed jointly by JLab and ESnet. Its design was primarily driven by requirements to support streaming Data Acquisition (DAQ) systems with real-time inline event packet redirection, reassembly, reconstruction, and analysis as the operational theme, but also to accommodate cooled (or hot) data WAN oriented inter-DOE Lab use cases.

The LB is composed of a *data-plane* processing element implemented in the P4 code of the FPGA firmware and the *control-plane* processing element implemented on the FPGA host computer using conventional host programming resources. The LB firmware that executes on the LB FPGA is a combination of P4 code and Register Transfer Logic (RTL) that is *synthesized* together to build the necessary firmware. Programming Protocol-independent Packet Processors (P4) is a domain-specific language for network devices, specifying how data plane devices (switches, NICs, routers, filters, etc.)

process packets. A baseline capability to function as a network device is supplied by the supporting RTL code. The following describes the details of how the data generation and redirection take place:

1) Data Generation:

a) LB Meta-Data: To interface to the EJFAT architecture and prepare data to be processed by the LB, the data stream must be sent as a sequence of data *events*, where an event is composed of any number of *data channels*. An *event* is data that is to be reassembled and reconstituted as a meaningful whole, and that the LB will aggregate and send to a single receiving host computer, using data channel id to direct to different ports on the same host facilitating parallel reassembly of channels. To accomplished this, the data is *segmented* into UDP packets of an acceptable size based on the underlying network fabric. Each UDP packet contains the following 128 bits (two 64 bit words) *LB meta-data* in network byte order at the start of the UDP payload:

- bits 0-7 the 8 bit ASCII character 'L'
- bits 8-15 the 8 bit ASCII character 'B'
- bits 16-23 the 8 bit LB version number starting at 1 (constant for run duration)
- bits 24-31 the 8 bit Protocol Number (very useful for protocol decoders e.g., wireshark/tshark)
- bits 32-47 or 16 bits Reserved, MBZ
- bits 48-63 an unsigned 16 bit *Entropy* value for destination port selection (see section II-B2b)
- bits 64-127 an unsigned 64 bit *EventId* for destination host selection

b) LB EventId: To maintain coherence of the LBs Epoch advance mechanism (see section II-B2a), the EventId is an unsigned 64 bit quantity (often a timestamp) that for the duration of a data transfer session (potentially indefinitely)

- Only increases
- Unique
- Never rolls over
- Never resets
- Serves as the top level aggregation tag across packets that should be sent to a single specific destination.

In DAQ applications, EventId is typically a timestamp.

c) RE Meta-Data: In addition, a *Reassembly Engine (RE) meta-data* section should follow the LB meta-data and provide information for downstream disposition at the receiving host. The RE meta-data is not defined by EJFAT and is a use case dependent cooperative agreement between sender and receiver. A typical example might be 128 bits as follows:

- bits 0-3 the 4 bit Version number
- bits 4-13 a 10 bit Reserved field
- bit 14 indicates first packet
- bit 15 indicates last packet
- bits 16-31 an unsigned 16 bit Data Channel Id
- bits 32-63 an unsigned 32 bit packet sequence number or optionally data offset byte number from beginning of file (BOF) for reassembly
- bits 64-127 an unsigned 64 bit *EventId*

2) Data Redirection:

a) Epoch Advance: The Epoch advance mechanism is the LBs technique for adjusting the mapping of data events to compute core hosts dynamically. An *Epoch* is defined as a sequential block of EventIds. The LB control-plane defines independent mappings for each of a configurable number of future Epochs which become effectively when the EventIds sent by data generators advance into each waiting Epoch. The LB then switches to the mapping defined for the newly current Epoch.

b) Control Plane Responsibilities: The LB *control plane* has the following responsibilities:

- Populate the LB Network ID Tables: Note that with this and the following mappings, the LB is capable of processing IPv4 and IPv6 streams concurrently with completely independent sets of mappings:
  - IPv4 Unicast MAC address
  - IPv4 Broadcast MAC address
  - IPv4 Unicast IP Address
  - ARP Target Protocol Address
  - IPv6 Solicited Node Multicast MAC Address
  - IPv6 All Nodes Link-Local Multicast MAC Address
  - IPv6 Unicast IP Address
  - IPv6 Solicited Node Multicast IP Address
  - IPv6 All Nodes Link-Local Multicast IP Address
- For both IPv4/IPv6, map each LB meta-data EventId to an *Epoch*; typically each of the available number of Epochs is defined as some subset (as opposed to *proper* subset) of the EventId sequence space. This specifies which Epoch is active for each EventId and is the primary technique that the CP uses to respond to changing conditions. New Epochs are defined (just-in-time if desired) to be come effective at some designated (upcoming) EventId and selects which core host redirection profile is effective for the specified future EventId range.
- Populate the [Epoch, LSB(EventId,n)] to Compute Core Member mapping; here the *n* least significant EventId bits are the Member Number for the Epoch.
- For each Epoch, for both IPv4/IPv6, map each Member Number to the tuple [MAC, IP, BasePort, PrtBits]; *PrtBits* is the number of *entropy* bits used to select the destination Port associated with the mapped MAC/IP, such that the mapped port is effectively ((LB meta-data *Entropy*) modulo (PrtBits)) + BasePort.
- Monitor downstream Core Compute telemetry.
- Allocate resources as conditions demand or preferably using AI/ML predictions/inferences.
- Provide upstream feedback to data generators.
- provide downstream feedback to Core Compute

c) Data Plane Operations: The LB P4 code processes the LB meta-data using the mappings specified in section II-B2b and as depicted in Figure 2, in the following manner for each received packet:

- Drop if LB meta-data does not match 'L','B' in first 16 bits
- Use EventId to determine Epoch
- Use [Epoch, LSB(EventId,n)] to determine Compute Core Member Number

- Use Compute Core Member Number to determine destination [MAC, IP, Port]
- Rewrite UDP packet header, remove LB meta data from packet, egress packet to network

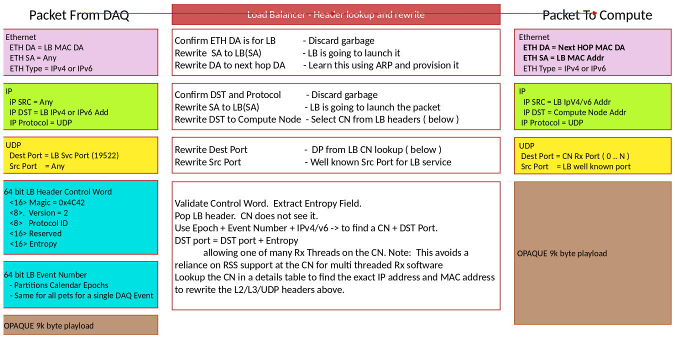
Packet From DAQ
Load Balancer - Header lookup and rewrite
Packet To Compute
Ethernet
ETH DA = LB MAC DA
ETH SA = Any
ETH Type = IPv4 or IPv6
Confirm ETH DA is for LB - Discard garbage
Rewrite SA to LB(SA) - LB is going to launch it
Rewrite DA to next hop DA - Learn this using ARP and provision it
Ethernet
ETH DA = Next HOP MAC DA
ETH SA = LB MAC Addr
ETH Type = IPv4 or IPv6
IP
iP SRC = Any
IP DST = LB IPv4 or IPv6 Add
IP Protocol = UDP
Confirm DST and Protocol - Discard garbage
Rewrite SA to LB(SA) - LB is going to launch the packet
Rewrite DST to Compute Node - Select CN from LB headers (below)
IP
IP SRC = LB IpV4/v6 Addr
IP DST = Compute Node Addr
IP Protocol = UDP
UDP
Dest Port = LB Svc Port (19522)
Src Port = Any
Rewrite Dest Port - DP from LB CN lookup (below)
Rewrite Src Port - Well known Src Port for LB service
UDP
Dest Port = CN Rx Port (0 .. N)
Src Port = LB well known port
64 bit LB Header Control Word
<16> Magic = 0x4C42
<8>. Version = 2
<8> Protocol ID
<16> Reserved
<16> Entropy
64 bit LB Event Number
- Partitions Calendar Epochs
- Same for all pets for a single DAQ Event
OPAQUE 9k byte playload
Validate Control Word. Extract Entropy Field.
Pop LB header. CN does not see it.
Use Epoch + Event Number + IPv4/v6 -> to find a CN + DST Port.
DST port = DST port + Entropy
allowing one of many Rx Threads on the CN. Note: This avoids a
reliance on RSS support at the CN for multi threaded Rx software
Lookup the CN in a details table to find the exact IP address and MAC address
to rewrite the L2/L3/UDP headers above.
OPAQUE 9k byte playload

Fig. 2. Load Balancer Packet Rewrite Function

d) Data Consumption: Data Consumers, Core Compute Hosts configured in the mappings specified in section II-B1a, listen for UDP packets for data channels implied by the port mapping algorithm given in section II-B2b, where in general the Host will listen on multiple ports for different channels concurrently. It is then up to the Host what the final disposition of packet stream for each channel associated with a common event. The typical disposition is to reassemble the event according to the packet order sequence for each channel implied by section II-B1c and how data received on the various channels is to be understood to constitute a meaningful data event.

III. EJFAT Benefits

Deployment of the EJFAT Architecture has many potential benefits to consider:

A. *Designed to Support Streaming Readout*

While the EJFAT Architecture is capable of supporting a traditional triggered event stream, it is primarily designed to support the future of DAQs with a streaming (non-triggered) readout. EJFAT anticipates and is designed to process a continuous stream of packets of indefinite duration by real-time indirection of work to resources determined to be available for work on a schedule of rotation among such resources, with scheduling provisions to designed to ensure their reuse after they have complete their previous assigned event processing.

B. *Simplifies Front Edge Electronics*

DAQ system front edge electronics should benefit from significantly reduced complexity and processing by obviating the need for complex trigger processing and associated read out of data.

C. *Leverages Advances in Network Acceleration*

As accelerated network devices continue to mature, EJFAT is well architected to continue to off-load processing by software by moving this processing to hardware in the form of flexible FPGAs.

D. *Network Decoupling/Indirection*

Edge to Core network decoupling is a major feature of EJFAT and provides the edge data generation layer with a single point or few points of indirection allowing the Core compute facilities to evolve and be located independently of the edge.

E. *Facilitates Near R/T Experiment Data Processing*

Rotation of work among a pool of worker Core computing resources should facilitate as near real-time data processing responses as is feasible especially as more work can be migrated to accelerated FPGA equipped resources, potentially even *in-network*.

F. *Reduce Archived Data Volume*

The more processing that can be accelerated and load balanced via host rotation, the more opportunities there will be for reduced archival requirements.

G. *Facilitates Data Centers Supporting Multiple Labs and Experiments (Reduced Power, Cost)*

Indirection via strategically placed LBs in an overarching EJFAT architecture with its decoupling of edge-to-core and therefore geographic independence of associated resources facilitates centralized high performance data centers of the future, and associated reduction of energy utilization, management, and other direct and indirect costs.

H. *Three Tier Horizontal Scaling*

EJFAT significantly increases the ability to leverage decoupled horizontal scaling:

1) First in the form of the rotation schedule among Core Compute hosts,
2) Second by facilitating parallel data processing of data channels for reassembly and also event reconstruction and post processing,
3) Third across separate LB deployments within the same EJFAT deployment enabling, for example, several Core Compute facilities to participate in the same experiment or data center processing use case, and it's converse centralization of back-end computing to serve geographically disperse and independent experiments.

:

References

[1] ”BES Computing and Data Requirements in the Exascale Age,” [Online]. Available: https://science.osti.gov/-/media/ascr/pdf/programdocuments/docs/2017/DOE-ExascaleReport_BES_final.pdf.

[2] ”FES Exascale Requirements Review,” [Online]. Available: https://science.osti.gov/-/media/ascr/pdf/programdocuments/docs/2017/DOE-ExascaleReport-FES-Final.pdf.

[3] ”HEP Exascale Requirements Review,” [Online]. Available: https://science.osti.gov/-/media/ascr/pdf/programdocuments/docs/2017/DOE-ExascaleReport-HEP.pdf.

[4] ”Nuclear Physics Exascale Requirements Review,” [Online]. Available: https://science.osti.gov/-/media/ascr/pdf/programdocuments/docs/2017/DOE-ExascaleReport-NP-Final.pdf.

[5] National Academies of Sciences, ”An Assessment of U.S.-Based Electron-Ion Collider Science.,” The National Academies Press., [Online]. Available: https://doi.org/10.17226/25171.

[6] ESnet, Energy Sciences Network, ”Nuclear Physics Network Requirements Review,” University of California, Publication Management System report number LBNL-2001281, May 8–9, 2019.

[7] Scientific Computing Plan for the ECCE Detector at the Electron Ion Collider [Online]. Available: https://arxiv.org/abs/2205.08607

[8] Streaming readout for next generation electron scattering experiment [Online]. Available: https://arxiv.org/abs/2202.03085