# ESnet / JLab FPGA Accelerated Transport

Michael Goodrich, Carl Timmer, Vardan Gyurjyan, David Lawrence, Graham Heyes, 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 compute cluster 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 a 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 load-balancing decisions for destination compute host redirection / load balancing. The LB provides for three-tier horizontal scaling across LB suites, cluster compute hosts, and CPUs within a host. The LB effectively provides seamless integration of edge / cluster computing to support direct experimental data processing for immediate use by JLab science programs and others such as the Electron-Ion Collider (EIC) as well as data centers of the future requiring high throughput and low latency for both time-critical (e.g., data acquisition systems) and data-driven (data center) workflows.**

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

## 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 ( [1], [2], [3], [4], [5], [6]) 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 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.

Manuscript received August 22, 2022; revised December 9, 2022; accepted January 25, 2023. 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, C. A. Timmer, V. H. Gyurjyan, D. Lawrence, and G. Heyes (e-mail: goodrich@jlab.org, timmer@jlab.org, gurjyan@jlab.org, davidl@jlab.org, heyes@jlab.org) are with the Thomas Jefferson National Accelerator Facility, Newport News, VA 23606 USA ; 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).

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 upwards of 30Gbs. The EIC [5] 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 [8] 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 cluster 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 [7] 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 [6] 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 ESnet6 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*?

- Time-critical workflows
- 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

The EJFAT - phase I effort is currently underway and is chiefly concerned with *time-critical workflows* and *data-driven workflows*.

### *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.

EJFATs design is fully compatible with both *triggered* and *streaming* (non-triggered) data-flows.

### *D. Technical Challenge - Streaming 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

The EJFAT architecture is being developed to answer this question. A primary architectural feature is FPGA based *acceleration* to

- Compress, segment and prepare the data at the edge (*Edge Traffic Shaping FPGA* - Figure 1)

- Dynamically load balance incoming streams of data into a cluster via in-flight destination redirection (*Load Balancing FPGA* - Figure 1)
- Decompress, reassemble, and post process data near the cluster using cluster node or cluster node NIC resident FPGAs (*Cluster* - Figure 1)

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 that are using streaming readout and investigating how to put these into production in the 12 GeV program as well as application to the EIC [7].

Electronics attached to scientific instruments digitize measurements. This digitized data could be packaged with meta-data 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. This supervisory agent is external to EJFAT but is expected to coordinate with it. Since the destination network addresses in the data center are necessarily opaque to the sender, they must be brokered by a proxy device - e.g., a suitably programmed FPGA (*Load Balancing FPGA* - Figure 1) - that functions as a single point of contact for decoupling the data generation network from the data consumption network.

Use of the highest possible bandwidth indicates a handshake-less protocol, e.g., UDP that is however susceptible to data loss. In *time-critical workflows* the delivery must be guaranteed and without back-pressure since it is unacceptable to tell an instrument to "slow down".

EJFAT architectural principles for mitigation of UDP packet loss are:

- Dedicated optical fibre within and connecting DOE labs extending from data source to data sink
- Adequate sizing/availability of cluster resources
- Data Plane Development Kit (DPDK) [9]
- Other Acceleration Opportunities:
  - FPGA accelerated network fabric (e.g., switches)
  - Host resident FPGAs for packet capture and Reassembly
  - NIC resident FPGAs for packet capture and Reassembly

While these principles are to eliminate packet loss entirely, actual packet loss tolerances are use-case defined; e.g., in some use-cases (e.g., *data-driven workflows*) it may be feasible to simply omit incomplete data and in others coordinate resending all or parts of the data.

### A. EJFAT Equipment

Equipment depicted in Figure (1):

- Xilinx Alveo U280 Data Center FPGAs
- Rackmount chassis
  - Dual Socket AMD EPYC 7763, 64c 2.4GHz
  - NVIDIA dual-port ConnectX-6 Dx Ethernet NIC

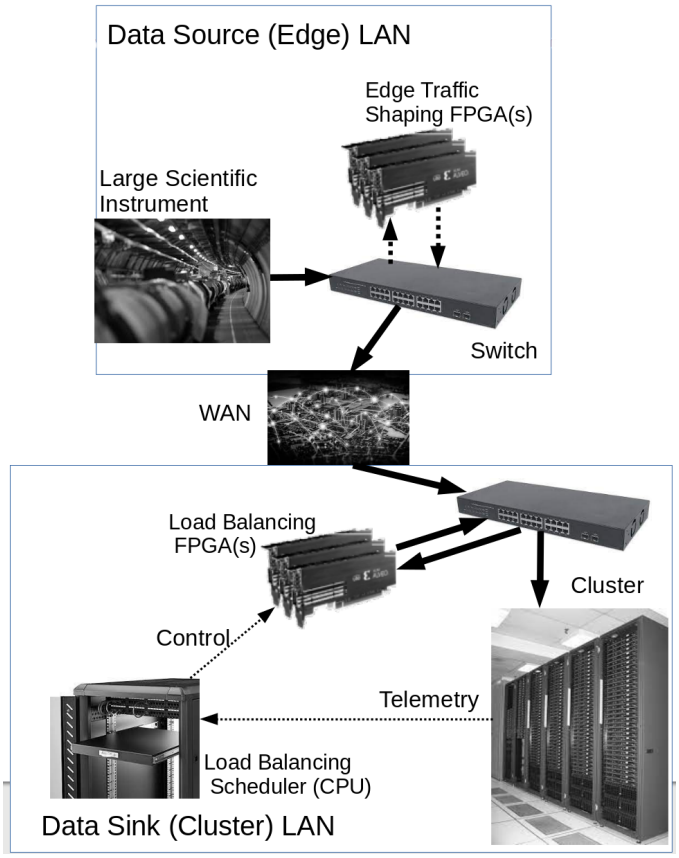


Fig. 1. EJFAT Architectural Concept

- Arista 7280R3, 32x100GbE QSFP and 4x400GbE QSFP-DD switch router

### B. Approach

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 (*Load Balancing FPGA* - Figure 1) that does not *process or buffer* data but redirects it in flight in real-time with minimal (microsecond) latency. It does this by modifying network packet headers in-situ such that it operates transparently at network bandwidth with no bottlenecks and also decouples the networking infrastructures of edge and cluster.

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 hardware or software, a better solution might be to use an additional edge data-shaping FPGA (*Edge Traffic Shaping FPGA* block - Figure 1).

### C. EJFAT Data Flows

Referring to Figure 1, EJFAT data originates at the edge data source and must be comprised of a stream of UDP packets with extra meta-data prepended to the UDP payload. This *load*

*balancing meta-data* is the first part of the UDP payload and identifies each packet for data event, data channel.

Meta-data tagged packets typically stream from the source into a local switch targeting the well known IP address/port of the Load Balancing FPGA(s) resident in the local campus LAN environment or on the other side of a WAN link. In some cases it may be desirable to off load the UDP fragmentation of the stream, meta-data tagging of each packet, and targeting of the Load Balancing FPGA functions to an Edge Traffic Shaping FPGA(s) typically near the data source. In such cases, the data streams into and back out of the Edge Traffic Shaping FPGA(s) located in the local LAN environment of the data source with microsecond level latency and are redirected to the Load Balancing FPGA(s).

After transiting the intervening network/WAN the packets stream into and back out of the Cluster Load Balancing FPGA(s) with microsecond level latency and are redirected to the particular compute node in the cluster that the Load Balancing FPGA has been configured to send to based on the data event ID and associated host port based on the data channel ID.

Addition *reassembly meta-data* in each packet following the load-balancing meta-data then provides adequate packet information for reassembly and typically consists of data event, data channel, and sequence number within each channel. This repetition of some data elements is necessary since in typical network device fashion, the load balancing FPGA removes the load balancing meta-data from the packet as it redirects to the destination cluster node.

Telemetry flows from the cluster into the Load Balancing Scheduler (*Load Balancing Scheduler* - Figure 1) that are typically host processes running on the load balancing FPGA host computer to determine any schedule adjustment (mappings) of data event IDs to cluster nodes based on available computational/buffering capacities of nodes

Schedule adjustments (mappings) are subsequently updated in the Load Balancing FPGA(s) from the Load Balancing Scheduler using the Load Balancing FPGA's Epoch Advance capabilities.

### D. Load Balancing Operations

*1) Edge Processing:* The following describes the requirements for front-end data preparation:

*a) LB Meta-Data:* To interface to the EJFAT architecture and prepare data to be processed by the LB, the data stream coming out of the Readout hardware (*Readout hardware* block - Figure 1) must eventually be sent as a sequence of data *events*. A data *event* is data that is to be reassembled, reconstituted, and analyzed as a meaningful whole, therefore the LB sends commonly tagged packets for an *event* to a single receiving host computer. Further, data from a scientific instrument is typically sent on different physical channels (e.g., read-out controllers) thus *events* are composed of data channels that are redirected to different ports on the same host facilitating parallel reassembly of channels within the encompassing *event*. To accomplish this, the data source must designate data as particular events, then UDP *segment* into packets with both load-balancing and reassembly meta-data for eventual reassembly at individual destination hosts per *event* for further processing.

In traditional triggered systems, *event discretization* is typically based on timestamps. With a new additional requirement for UDP fragmentation, and both load-balancing and reassembly (see section II-D2a) metadata tagging for each packet, these traditional triggered systems may utilize the capabilities of EJFAT.

In non-triggered *streaming* systems, the event discretization logic must define event boundaries as well as attend to UDP fragmentation, and load-balancing and reassembly metadata tagging for each packet, where this logic is use-case (workflow) defined.

For time-critical workflows (e.g., DAQ applications), the event discretization functions will likely be absorbed into modified existing hardware/firmware or be placed into additional FPGAs (e.g., *Edge Traffic Shaping FPGA* block - Figure 1), while in data-driven workflows may possibly be in software.

**It should be noted that LB operations are only dependent on proper event/channel tagging of packets and therefore EJFAT LB operations may be used for both streamed and triggered data event sources.**

*b) LB EventId:* To maintain coherence of the LBs *Event Epoch* advance mechanism (see section II-D3a), the *EventId* is an unsigned 64 bit quantity (e.g., timestamp or event counter/designator) that for the duration of a data transfer session (potentially indefinite):

- 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.

*2) Packet Redirection:* The key component of the EJFAT architecture is the LB FPGA/Host suite (*Load balancing FPGA* and *Load Balancing Scheduler* blocks Figure 1) that is the bridging component between the edge and the cluster computing facilities. The LB FPGA is currently a Xilinx U280 FPGA PCIe card with $2 \times 100$ Gb/s optical ports using firmware developed by ESnet from specifications developed jointly by JLab and ESnet. The EJFAT Architecture and in particular the LB's design were primarily driven by requirements to support both streaming time-critical workflows with real-time inline event packet redirection, reassembly, reconstruction, and analysis but also to accommodate data-driven workflows.

The LB is composed of a *data-plane* processing component implemented in the Programming Protocol-independent Packet Processors [10] (P4) code of the FPGA firmware and the *control-plane* processing component 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. 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.

*a) 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 workflow dependent cooperative agreement between sender and receiver.

*3) Data Redirection:*

*a) Epoch Advance:* The Epoch advance mechanism is the LBs sole technique for adjusting the mapping of data events to cluster 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 cluster host redirection mapping is effective for the specified future EventId range.
- For both IPv4/IPv6, provide mappings for [Epoch, EventId, DataChannel] to Cluster [MAC, IP, Port]
- Monitor downstream cluster telemetry.
- Provide upstream feedback to data generators.
- Provide downstream feedback to Cluster

*c) Data Plane Operations:* The LB P4 code processes the LB meta-data using the mappings specified in section II-D3b in the following manner for each received packet (see Figure (2):

- Map EventId to Epoch
- Map [Epoch, EventId] to cluster [MAC, IP, Port]
- Rewrite UDP packet header for mapped [MAC, IP, Port], remove LB meta data from packet, egress packet to network

*d) Data Consumption:* Data consumers (cluster nodes) configured in the mappings specified in section II-D1a, listen for UDP packets for data channels implied by the port mapping algorithm given in section II-D3b, where in general the cluster nodes will listen on multiple ports for different channels concurrently. It is then up to the node for the final disposition of the 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-D2a and how data received on the various channels is to be understood to constitute a meaningful data event.

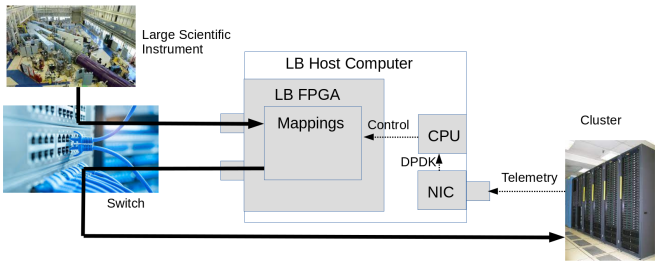


Fig. 2. LB Data Flows

## III. EJFAT Benefits Summary

As elaborated, deployment of the EJFAT Architecture has many potential benefits to consider:

### *A. Designed to Support Both Time-Critical (Streaming/Triggered) and Data-Driven Workflows*

The EJFAT Architecture is capable of supporting both traditional triggered event streams and also streaming (non-triggered) time-critical workflows. Streaming (non-triggered) workflows appear particularly indicated for data-driven workflows. 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 designed to ensure their available capacity for reuse after they have complete their previous assigned event processing.

### *B. Simplifies Front Edge Electronics*

DAQ system front edge electronics could benefit from significantly reduced complexity and processing by obviating the need for complex trigger processing and associated read out of data, and taking advantage of EJFAT's support for real-time packet redirection of continuously streamed 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 and introduce new forms of acceleration by moving this processing to hardware in the form of flexible FPGAs, and FPGA accelerated network and cluster node NICs and other devices.

### D. Network Decoupling/Indirection

Edge to Cluster 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 Cluster compute facilities to be re-mapped, evolve and be located independently of the edge.

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

Rotation of work among a pool of worker Cluster 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-cluster 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 cluster nodes,
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 cluster facilities to participate in the same experiment or data center processing workflow, 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. Accessed August 22, 2022

[2] "FES Exascale Requirements Review," [Online]. Available: https://science.osti.gov/-/media/ascr/pdf/programdocuments/docs/2017/DOE-ExascaleReport-FES-Final.pdf. Accessed August 22, 2022

[3] "HEP Exascale Requirements Review," [Online]. Available: https://science.osti.gov/-/media/ascr/pdf/programdocuments/docs/2017/DOE-ExascaleReport-HEP.pdf. Accessed August 22, 2022

[4] "Nuclear Physics Exascale Requirements Review," [Online]. Available: https://science.osti.gov/-/media/ascr/pdf/programdocuments/docs/2017/DOE-ExascaleReport-NP-Final.pdf. Accessed August 22, 2022

[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. Accessed August 22, 2022

[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 Accessed August 22, 2022

[9] Data Plane Development Kit [Online]. Available: https://www.dpdk.org Accessed August 22, 2022

[10] Programming Protocol-independent Packet Processors [Online]. Available: https://opennetworking.org/p4/ Accessed August 22, 2022