JLab LDRD2410 Quarterly Report FY24Q3

Proposal Name:
Streaming Readout Real-Time Development and Testing Platform

Report Date:
Jul. 17, 2024

Principal Investigator:
David Lawrence PhD

1 Work-scope Highlights

Work continues with progress towards multiple milestones. Some issues encountered that slowed progress in some areas. This results in some schedule lag. The milestone schedule for the 2nd year proposal were adjusted slightly to account for that. We will continue to monitor this closely and will communicate with the program managers if we foresee any reduction in scope is necessary.

The major highlights are:

- Data Capture exercise on 05/16/2024 with CODA configuration consisting of 12 crates that read out the PCAL, ECAL, FTOF, and DC in sectors 2 and 5 with average payload rate as shown in Figure 1.
- Continued graph visualization and configuration
- Streaming of podio ePIC data with PoP test from CERN to US
- GlueX online has been containerized and data processing is working.

2 Data Capture

Data capture exercise performed on 05/16/2024 had the bandwidths for the different ports as represented in figure 1. This exercise utilized 24 streams from 12 crates, achieving bandwidths in excess of 1 GB/sec. This event marked the first large-scale data capture of this magnitude at Jefferson Lab. During the exercise, we collected data using beam with the following currents: 5 nA, 70 nA, 40 nA, 60 nA, 25 nA, 80 nA, and 50 nA. The data collected was stored on NVMe disks at ejfat-fs-daq using *tcpdump*.

During the data capture exercise, it was observed that there were 3 instances of the tcpdump program running on ejfat-fs-daq. Two of these were lingering from earlier tests and were not noticed as they appeared dormant until the high speed NIC began receiving data for this test. At that point, three different files were being written to the NVME disk thereby limiting the bandwidth to storage to about 1/3rd of the measured 1.6GB/s rate.

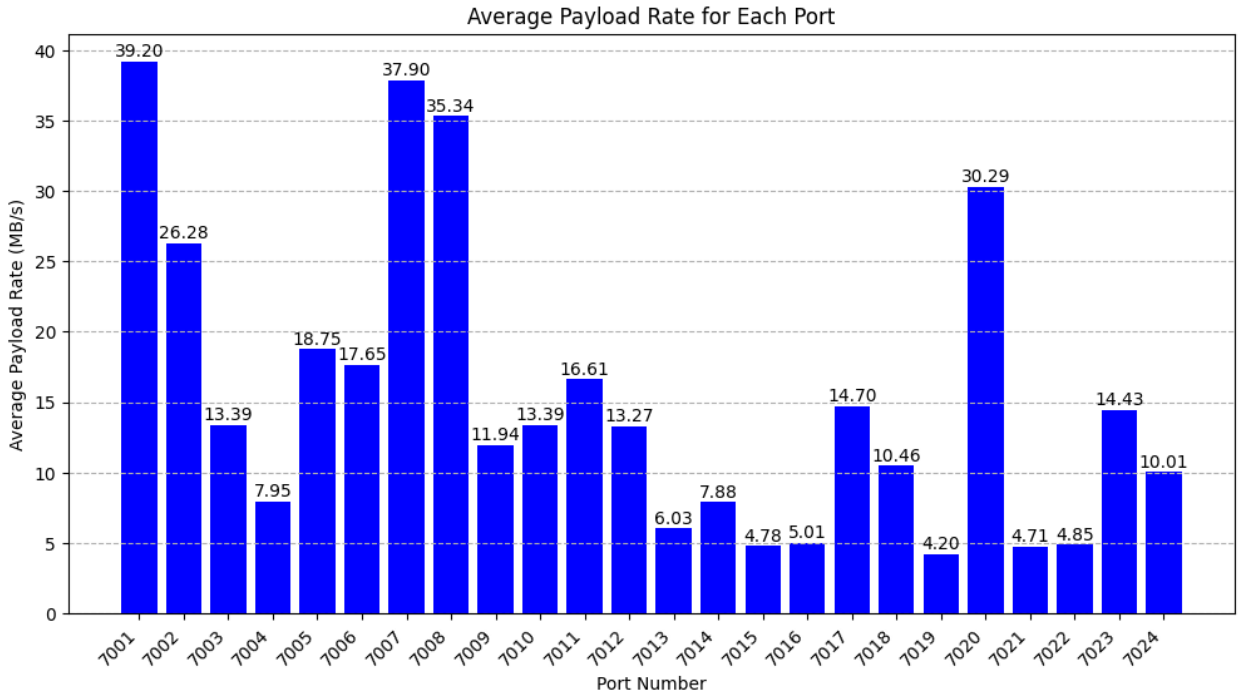


Figure 1: Average payload rate from each port for data capture

2.1 Limitations

The following limitations were observed during the experimentation :

- Bandwidth: The total amount of bandwidth to storage that was available during the data capture exercise was 1.6 GB/s. It was anticipated that the bandwidth would suffice based on the data capture exercise conducted on December 17, 2023. In that exercise only 4 streams were available, each sending 4-5 MB/s when running at 100nA. This was slightly higher than the production beam current for the running experiment at the time. Scaling this up to 24 streams would estimate the May 2024 test to generate an aggregate of only 120MB/s or less than 8% of the available 1.6GB/s bandwidth to storage. For most beam currents, this turned out to be insufficient due to multiple reasons descibed below.

- Additional instances of *tcpdump*: It was also observed during the exercise that there were two (unintended) additional process of *tcpdump* capturing data from the detectors. It went unnoticed as there was no active process indication of the running *tcpdump* when no data was being sent to the NIC. These instances were only discovered when we started capturing the data during the exercise. As a result, the additional instances shared a portion of the available bandwidth during most of the data capture exercise. This essentially means that we were trying to write to the NVMe disk at a bandwidth three times higher than the available bandwidth. It should be noted that using *iftop* on the high-speed NIC will not catch the instance as iftop indicates the aggregate rate at which data is received. However, the *top* command would have revealed the number of active processes, which could have helped us catch the additional *tcpdump* instances.

- Different experimental conditions and detector readout: The data capture exercise conducted in December 2023 had a different target configuration compared to May 2024. The detectors being read out were also different. In December, there were a total of four streams read from 2 crates reading out only ECAL and PCAL channels in a single sector. This data had a combined rate of around 16 MB/s. During the May 2024 data capture exercise, a separate target consisting of liquid deuterium and lead was used with a nominal running current of 70 nA. Compare this to the LH_2 target used in the Dec. exercise where the target was later determined to be empty or emptying over the course of the captures

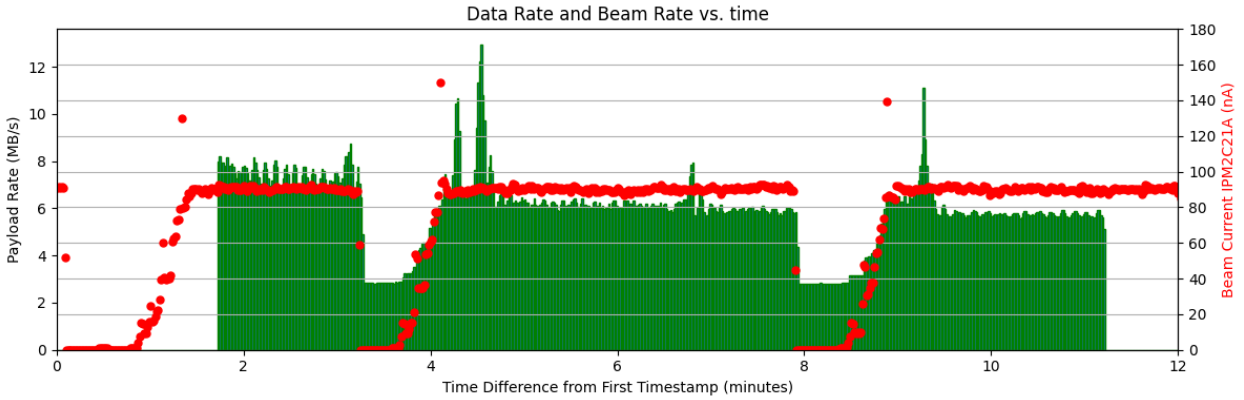


Figure 2: Data rate of captured packets (payload only) vs. time in minutes relative to start run time. The red points indicate the beam current during the same time period, plotted against the right hand axis. The slow reduction in data rate is due to the liquid hydrogen target emptying over this time period. The exact origin of the spikes in the data rate is unknown. This illustrates that even during periods when the beam is steady, the network traffic may have structure.

(see figure 2). The May configuration, along with the new crates which included Drift Chamber (DC) readout, generated data at a much higher bandwidth ($\geq$ 3 GB/s). Once the bandwidth to storage limit was reached, the VTP modules, by design, would start dropping frames. The files themselves though indicated corrupted EVIO format which indicates TCP packets were being lost. Likely at the ejfat-fs node where the actual backup occurred. Only the 5nA data set was completely intact.

3 Milestone Progress

Table 1 shows the status of the FY24Q1-2 milestones. Five are completed, two are partially completed, and one has not been completed as originally envisioned.

Milestone M09 focuses on implementing the LDRD project JIRIAF (JLAB Integrated Research Infrastructure Across Facilities) to send packets from JLab and process them at NERSC. As of June 30th, the following tasks has been completed:

1. Built Docker images containing the sender and receiver from the GitHub repository: `https://github.com/JeffersonLab/SRO-RTDP/tree/main/src/utilities/cpp/podio2tcp`. The built Docker image is located at: `https://hub.docker.com/repository/docker/jlabtsai/eic/general`.

2. Ran the EIC reconstruction on Perlmutter nodes at NERSC and on EJFAT nodes at JLab, indicating that RTDP can provide various resources for processing. Explored monitoring systems, including the Prometheus Process Exporter (`https://github.com/ncabatoff/process-exporter`) for I/O processes.

3. Combined metrics from processing at NERSC with a Kubernetes (K8s) cluster set up at JLab under the JIRIAF framework. These metrics were collected and preserved in Prometheus and Grafana instances in our local cluster, demonstrating a potential scalable monitoring solution for the RTDP platform.

4. Created a demonstration of the Node Graph that involves retrieving real-time metrics from a Prometheus database, transforming them to fit the Node Graph data API, serving them through a Python Flask backend, and finally visualizing them in Grafana.

Milestone M11 represents a fully integrated monitoring system (Hydra) along with RTDP that monitors the data replayed through the SRO. The purpose of the milestone is to generate ".png" format images from

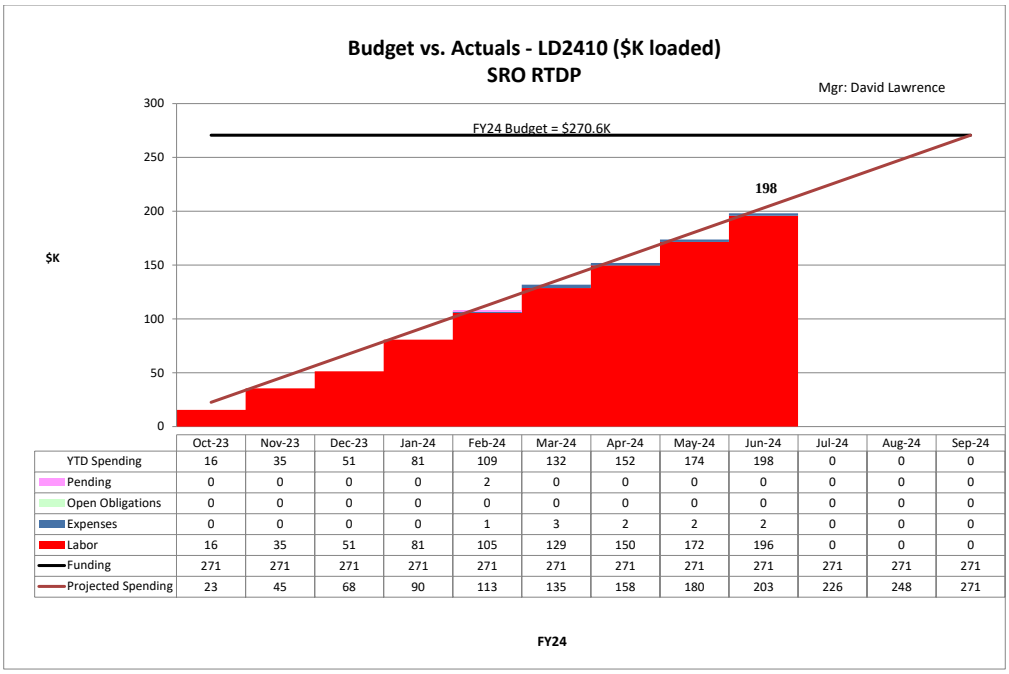
the occupancy online and feed it as input to Hydra for monitoring. The development of this integrated framework is partially complete. GlueX online has been containerized and data processing is working. Issues still being worked out with the monitoring system.

ID	Task	status	Comments
M01	Create prototype ERSAP configurations for INDRA and CLAS12 test systems	■	A CLAS12 example and "Hello World" example have been placed in Github. INDRA has not been done yet.
M02	Identify or capture SRO formatted data from CLAS12 and INDRA test systems with data tag/filtering capability (output data ready for further offline processing)	✓	Data was captured at various beam currents from CLAS12 on Dec. 17. INDRA data capture done using pulser inputs to SAMPA setup.
M03	Evaluate existing solutions for configuring and launching remote distributed processes	✓	see evaluations in document on EPSCI wiki.
M04	Establish code repository(s), project site, and method of documentation	✓	This has been done here: https://github.com/JeffersonLab/SRO-RTDP
M05	Create stream splitter program for EVIO or HIPO data formatted files	✓	Created for GlueX. (See text for details on HIPO)
M06	Create stream splitter program for simulated data in PODIO for ePIC	✓	Prototype tested using FABRIC testbed. Simulated ePIC data sent from CERN to 8 different US sites.
M07	Create VTP emulator using files produced by stream splitter	■	Mostly done for raw data. Not started for simulated data.
M08	Create controller program to synchronize multiple VTP emulators	✓	Satisfied through alternate design using synchronized system clocks.
M09	Determine appropriate schema for all aspects of monitoring	■	Monitoring info. extracted as JSON records from both docker and /proc sources on Linux. Display in Grafana prototyped, but not yet complete.
M10	Establish databases for monitoring system using existing JLab servers.	×	This work has not begun
M11	Integrate Hydra as monitoring component.	■	Work done to containerize GlueX online monitoring in order to allow full test with Hydra. The Hydra is nearly complete containerizing Hydra (for off project purposes) which we will use.
M12	Integrate off-line data analysis framework into platform for CLAS12 data	-	planned for FY24Q4
M13	Integrate off-line data analysis framework into platform for ePIC or GlueX simulated data	-	planned for FY24Q4
M14	Integrate example JANA2 analysis into platform	-	planned for FY24Q4

Table 1: FY24 Milestones

4 Budget

Figure 3 shows the project spending as of the end of FY24Q3. This is considered on track with the project expectations.



	Oct-23	Nov-23	Dec-23	Jan-24	Feb-24	Mar-24	Apr-24	May-24	Jun-24	Jul-24	Aug-24	Sep-24
YTD Spending	16	35	51	81	109	132	152	174	198	0	0	0
Pending	0	0	0	0	2	0	0	0	0	0	0	0
Open Obligations	0	0	0	0	0	0	0	0	0	0	0	0
Expenses	0	0	0	0	1	3	2	2	2	0	0	0
Labor	16	35	51	81	105	129	150	172	196	0	0	0
Funding	271	271	271	271	271	271	271	271	271	271	271	271
Projected Spending	23	45	68	90	113	135	158	180	203	226	248	271

M:\budget\FY2024\FY24_Level 2\CST\Monthly Reports\FY24 WBS 1.10 CST Spending Report Master Worksheet - TS, 7/3/2024

Figure 3: LD2410 Project Spending through FY24Q2. Values are in $K.

5 Concerns

Still some schedule slippage from original plan. These are software issues that are being worked through and the schedule for FY25 has been adjusted accordingly.

Acknowledgements

The research described in this report was conducted under the Laboratory Directed Research and Development Program at Thomas Jefferson National Accelerator Facility for the U.S. Department of Energy.

Appendix: Full Project Milestones

- **Y1Q1**
  - M01: Create prototype ERSAP configurations for INDRA and CLAS12 test systems
  - M02: Identify or capture SRO formatted data from CLAS12 and INDRA test systems with data tag/filtering capability (output data ready for further offline processing)
  - M03: Evaluate existing solutions for configuring and launching remote distributed processes
  - M04: Establish code repository(s), project site, and method of documentation
- **Y1Q2**
  - M05: Create stream splitter program for EVIO or HIPO data formatted files
  - M06: Create stream splitter program for simulated data in PODIO for ePIC
  - M07: Create VTP emulator using files produced by stream splitter
  - M08: Create controller program to synchronize multiple VTP emulators
- **Y1Q3**
  - M09: Determine appropriate schema for all aspects of monitoring system.
  - M10: Establish databases for monitoring system using existing JLab servers.
  - M11: Integrate Hydra as monitoring component.
- **Y1Q4**
  - M12: Integrate off-line data analysis framework into platform for CLAS12 data
  - M13: Integrate off-line data analysis framework into platform for ePIC or GlueX simulated data
  - M14: Integrate example JANA2 analysis into platform
- **Y2Q1**
  - M15: Create configurable CPU proxy component
  - M16: Create configurable GPU proxy component (hardware and software)
  - M17: Create configurable FPGA proxy component (hardware and software)
  - M18: Create functioning hardware GPU component (e.g., CLAS12 L3)
  - M19: Create functioning hardware FPGA component (e.g., ML4FPGA)
- **Y2Q2**
  - M20: Impose artificial time structure on stream sources to mimic beam-like conditions
  - M21: Configure simulation of full SRO system using existing JLab hardware resources
- **Y2Q3**
  - M22: Establish working test of system that transfers ¿=100Gbps from CH to compute center
  - M23: Establish working test of system that includes GPU component for portion of the stream
  - M24: Establish working test of system that includes FPGA component for portion of the stream
  - M25: Test system with remote compute facility (e.g., BNL or NERSC) at limits of available resources
- **Y2Q4**
  - M26: Configure system that results in stream(s) being received by JLab from external source
  - M27: Collaborate with HPDF group to evaluate processing SRO data at JLab for external experiments
  - M28: Complete documentation for platform to be used by non-experts