FPGA ARCHITECTURES FOR DISTRIBUTED ML SYSTEMS FOR REAL-TIME BEAM LOSS DE-BLENDING

M.A. Ibrahim*, M.R. Austin, J.M. Arnold, J.R. Berlioz, P. Hanlet,
K.J. Hazelwood, J. Mitrevski, V.P. Nagaslaev, D.J. Nicklaus, G. Pradhan, A.L. Saewert,
B.A. Schupbach, K. Seiya, R.M. Thurman-Keup, N.V. Tran, A. Narayanan[1]
Fermi National Accelerator Laboratory†, Batavia, IL USA
J.YC. Hu, C. Xu, J. Jiang, H. Liu, S. Memik, R. Shi, A.M. Shuping, M. Thieme
Northwestern University‡, Evanston, IL USA
[1]also at Northern Illinois University, DeKalb, IL USA

Abstract

The Accelerator Real-time Edge AI for Distributed Systems (READS) project's goal is to create a Artificial Intelligence (AI) system for real-time beam loss de-blending within the accelerator enclosure, which houses two accelerators: the Main Injector (MI) and the Recycler Ring (RR).

In periods of joint operation, when both machines contain high intensity beam, radioactive beam losses from MI and RR overlap on the enclosure's beam loss monitoring Beam Loss Monitor (BLM) system, making it difficult to attribute those losses to a single machine. Incorrect diagnoses result in unnecessary downtime that incurs both financial and experimental cost. The ML system will automatically disentangle each machine's contributions to those measured losses, while not disrupting the existing operations-critical functions of the BLM system.

This paper will focus on the evolution of the architectures, which provided the high-frequency, low-latency collection of synchronized data streams to make real-time inferences. The ML models, used for learning both local and global machine signatures and producing high quality inferences based on raw BLM loss measurements, will only be discussed at a high-level.

INTRODUCTION

Accelerator Complex

After the Collider Physics program [1] ended in 2011, RR was re-purposed as a proton stacker for Main Injector, delivering 8-120 GeV beam to multiple experiments and facilities. The RR is directly installed above MI, such that their beam lines' centers are physically separated by only about 120 cm. As a result, when high intensity beams are in both machines simultaneously, understanding beam losses becomes a significant concern during normal operation of the accelerator complex (Fig. 1).

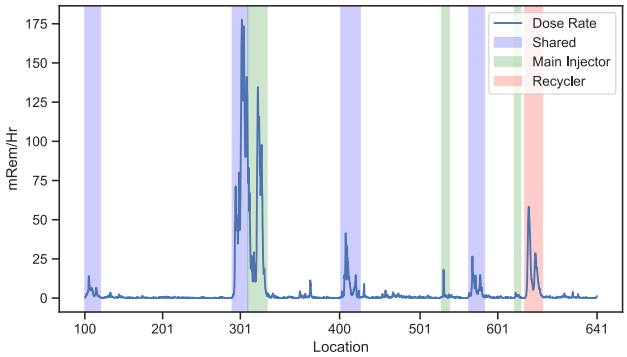

Figure 1: Plot of localized beam losses based on tunnel residual dose rates.

Beam Loss Monitoring

Real-time localized beam losses for both the MI and RR are monitored by over 250 argon gas ionization chamber-type BLM detectors. These signals are received, processed, and instrumented within a Versa Module Eurocard (VME)-based architecture, forming a distributed network of 7 VME "front-end nodes" around the 2.2 mile complex. Together, they capture and report spatially-identifiable and time-correlated integrated beam loss measurements on all BLM detectors within the enclosure for display and analysis [2].

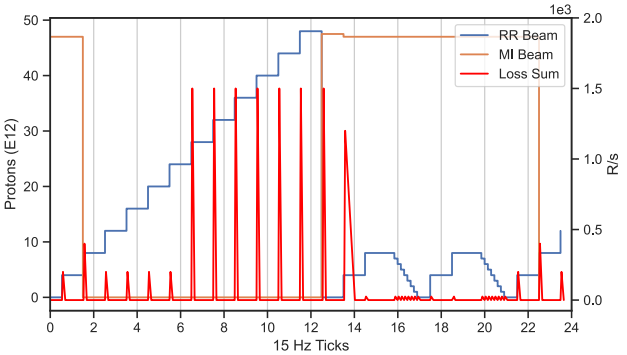

Figure 2: Example to overlay beam events and expected losses during a machine cycle.

Although the origin of radioactive losses measured on any operational BLM can be difficult to attribute to a single machine, experts can often manually decipher and attribute losses to either MI or RR, based on timing, machine state, and physical location within the ring (Fig. 2).

---

* Equal contribution

† Operated by Fermi Research Alliance, LLC under Contract No.De-AC02-07CH11359 with the United States Department of Energy. Additional funding provided by Grant Award No. LAB 20-2261

‡ Performed at Northwestern with support from the Departments of Computer Science and Electrical and Computer Engineering

READS DISTRIBUTED NETWORK

The current BLM VME nodes were not designed to simultaneously output all BLM measurements to ACNET [3], at their VME digitizers' maximum polling frequency of 320 Hz. Instead, through the ACNET Data Pool Managers (DPM), the time resolution of the integrated loss measurements is relatively course (33 Hz, max).

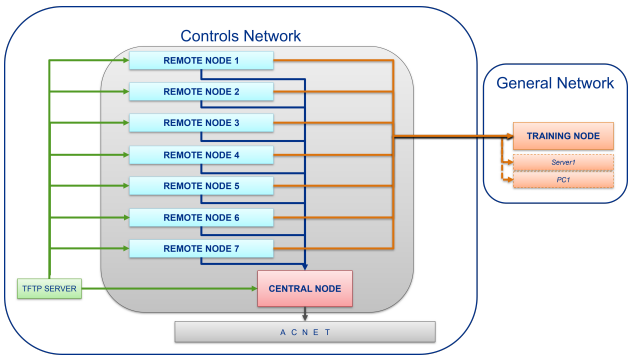

Figure 3: Block diagram of READS distributed network.

Consequently, READS Distributed Network (Fig. 3) provides a separate data path, which does not disturb normal operations of the BLM system. The main components of this network are Remote Data Aquisition (DAQ) Nodes, Central Deblending Node, and Training Nodes.

CLIENT-SERVER MODEL

The Distributed Data Communications Protocol (DDCP) (Fig. 4) is a lightweight User Datagram Protocol (UDP) application layer protocol, used within the READS network to establish client-server relationships between the Remote DAQ Nodes and ML-related nodes (i.e. the Training Node and Central Deblending Node).

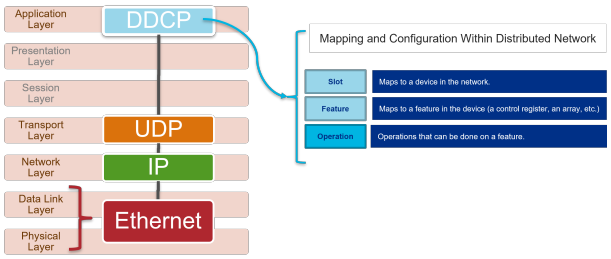

Figure 4: Block diagram of DDCP Ethernet stack.

DAQ nodes act as DDCP servers while ML-related nodes act as DDCP clients. While DDCP had been previously used to read and write parameters between operational nodes of other systems, development was needed to implement a streaming service. This feature pushes data from the server to the client automatically after just one client request. Furthermore, each server can connect to multiple clients; clients can connect to multiple servers. Optimizations were also needed to preserve configuration compliance across the READS network, ensuring packets can be efficiently aggregated and correctly correlated.

REMOTE DAQ NODE

Uniquely registered within the Controls Network, a *VME Reader Card* is integrated into each BLM VME node and serves as a Remote DAQ Node within the READS network. Utilizing a *Critical Link MitySOM-5CSX* System On Module (SOM) (Fig. 5), it passively monitors the VME backplane as well as generates data streams of User Datagram Protocol (UDP) packets with a customized application layer [4].

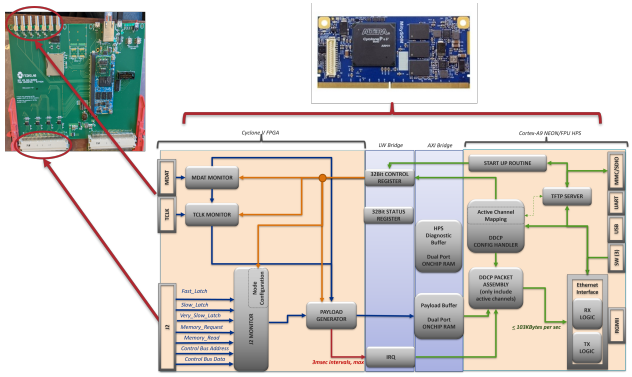

Figure 5: Block diagram of SOC architecture of Remote DAQ Node.

The SOM hosts an *Altera Cyclone V* System On Chip (SOC), which is a highly-integrated infrastructure of both a Field Programmable Gate Array (FPGA) and dual-core *ARM Cortex-A9* Hard Processor Subsystem (HPS). The architecture of the Remote DAQ Node was implemented and optimized to provide consistent Ethernet performance to support the multiple clients, DDCP data streams, and remote troubleshooting.

The FPGA is responsible for the receiving integrated sums from the BLMs as well as encoded signals from the global event system and frame links, in order to generate each packet's payload. Each packet's data payload include epoch second, millisecond time-stamping, event-based counters, decoded global clock events, machine readings and state information for proper synchronization and correlation (Fig. 6). Also, the streams are aligned, with respect to a single, asynchronous global clock event, which regularly occurs at 5 sec intervals. Assembly of the READ data packets, and serialized transmission of packets into data streams, and management of other Ethernet communication falls to the HPS. Other supported services are Trivial File Transfer Protocol (TFTP) for remote programming and booting, Kerberized Secure Shell Protocol (SSH) for expert accessibility, and ping for verifying proper network registration.

TRAINING NODES

The Training Node is a designated server on the general network, that receives and archives data streamed from the Remote DAQ Nodes into a storage device. Functionally, the server mimics the DPM but provides for faster data collection rates at 333 Hz. These Training Data Sets are then used offline to develop, train, and optimize an U-Net-based ML model, which was prototyped using Sample Data Sets

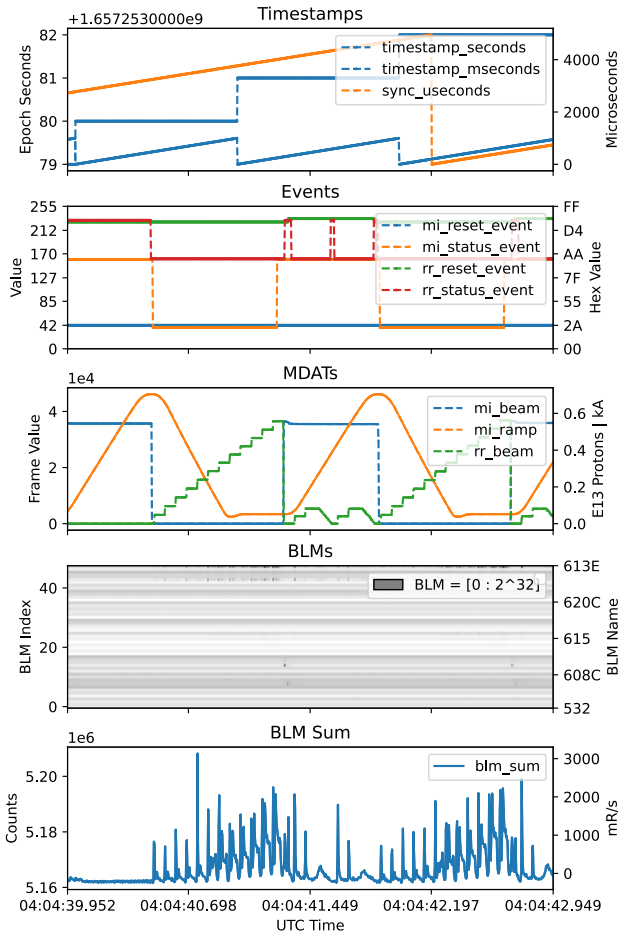

Figure 6: Display of data collected and streamed from a Remote DAQ Node.

collected at 33 Hz from the existing BLM System via the ACNET DPM [5]. Training data sets capture measurements collected during normal operation but also during special study times. During such times, the beam event timeline is manipulated to purposefully keep events for RR and MI from overlapping, thus only having beam in one machine at a time. Moreover, to ensure that a broad range of loss conditions are captured, moderate beam losses were also generated at all locations in both machines using various miss-configurations of the machines. This allowed for verification of the model on data that is possible but not generated during normal operations.

CENTRAL DE-BLENDING NODE

A Central De-blending Node ingests streamed packets from the Remote DAQ nodes to output inferences, which attributes the origin of the loss at each BLM location [6]. The Central Node uses a *REFLEX CES Achilles Arria10SoC SOM*, which is a mezzanine card that mounts onto a carrier board in a rack-wide chassis [7]. As in the Remote DAQ Node, this SOM also provides an embedded systems environment with both a FPGA and a dual-core *ARM Cortex-A9* HPS (Fig. 7). However, the Central Node hosts an *Altera Arria 10* SOC.

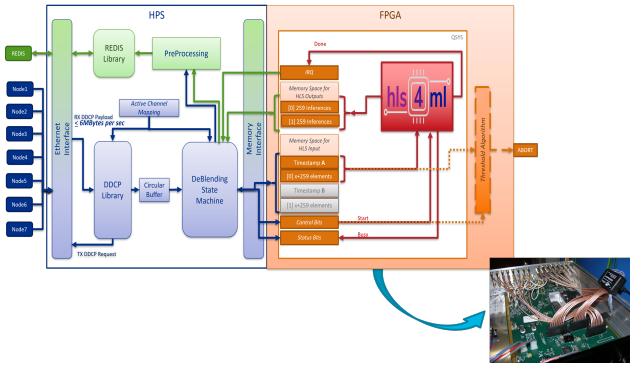

Figure 7: Block diagram of SOC architecture of Central Deblending Node.

The FPGA implements an embedded inference engine based on the trained model. Optimizations at the algorithm level, high-level synthesis level, and communication protocol level improved the overall node's speed, power consumption, and resource usage.

Packets, containing the engine's required inputs, are received, deserialized, correlated in time, and queued for input by the HPS. The engine outputs real-time inferences back to the HPS. These inferences are in percentages, attributing the likelihood of the integrated beam loss at each BLM to either the MI or RR. Then, the HPS provides these results to control system for display within the Main Control Room. Development in the HPS focused on minimizing the latency and optimizing the consistency of the arbitration of state machines handling Ethernet interface and lightweight bridge interfaces.

ML Model Implementation

A workflow was developed to integrate open-source tools, which have been successfully used to implement neural networks on *Xilinx* chips, with *Intel* FPGA compilation and build tools (Fig. 8). *HLS4ML* was used to translate the original U-Net model, written in *Keras*, to High Level Synthesis (HLS) for the Intel HLS compiler.

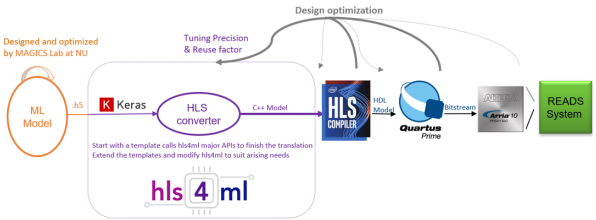

Figure 8: Block diagram of Keras to HLS4ML workflow.

Beam studies have confirmed that the ML model works well for ring-wide loss attribution, but not for narrower localized losses. The implemented neural network had a maximum latency of 1.74 ms and average latency of 1.2 ms meeting the required specification. Transfers between HPS and FPGA were about 200 µs.

Control System Interface

The resulting inferences need to be accessible to Main Control Room operators and experts thru existing tools used

to tune and diagnose the accelerators. In order to do so, the Central Node establishes a connection to a *RE mote DI ctionary S erver* (REDIS) server, which acts a versatile data structure server that can be used as a database, cache, streaming engine, and message broker to ACNET. In this manner, arrayed ACNET devices for the inferences are updated in real time and are available for fast time plotting via a READS *JAVA* Open Access Client (OAC). The fully deployed system satisfies the 3 ms latency requirement for the complete data signal path through the READS network (Fig. 9).

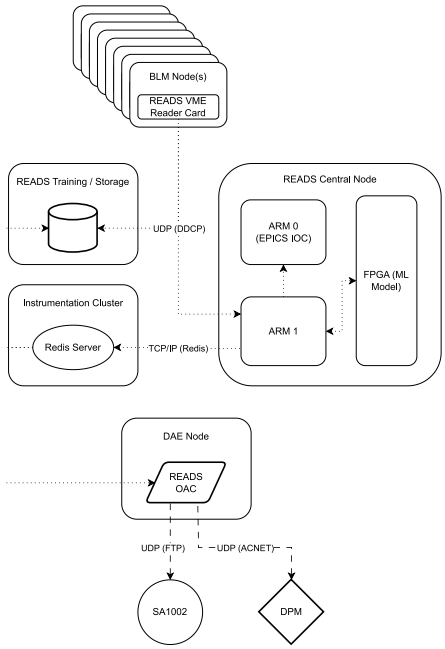

Figure 9: Dath Path to and from READS network to ACNET Control System.

CONCLUSION

READS successfully implemented and deployed the first operational FPGA-based edge-AI system in the Fermilab accelerator complex. Distributed remote nodes create and stream BLM readings from around the accelerator complex to perform near real-time inferences at a centralized node, which feed ACNET devices available to operators. Also, dedicated beam studies confirmed that ML model implementation agrees with the offline predictions from the Keras model. As a result, operators have a new tool to attribute percentages of beam loss, from each BLM, in MI and RR (Fig. 10). The system is expected to reduce pulse inefficiencies during tuning as well as unnecessary downtime due to tripping off the incorrect machine.

ACKNOWLEDGMENTS

This work was produced by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. Publisher acknowledges the U.S. Government license to provide public access under the DOE Public Access Plan [8]. Additional funding provided by Grant Award No. LAB 20-2261 [9].

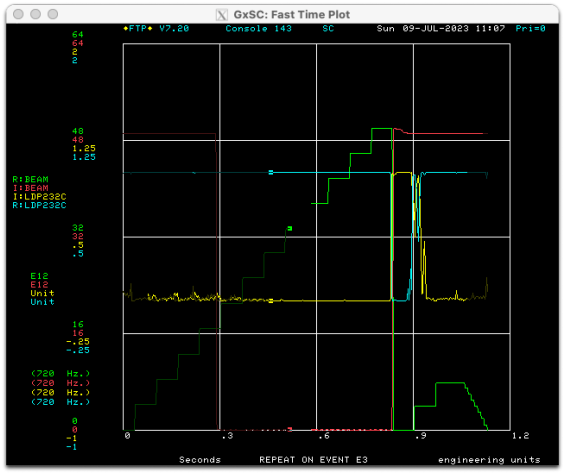

Figure 10: Real-time ML inferences on Fast Time Plot display - I:LDP232C and R:LDP232C are the loss percentage attribution for 232C BLM.

REFERENCES

[1] G. Jackson, "The Fermilab Recycler Ring Technical Design Report. Revision 1.2," Nov. 1996. doi:10.2172/16029

[2] A. Baumbaugh *et al.*, "The upgraded data acquisition system for beam loss monitoring at the fermilab tevatron and main injector," *Journal of Instrumentation*, vol. 6, no. 11, Nov. 2011. doi:10.1088/1748-0221/6/11/T11006

[3] J. F. Patrick, "The Fermilab Accelerator Control System," in *Proc. ICAP'06*, Chamonix, Switzerland, Oct. 2006, pp. 246–249. https://jacow.org/icap06/papers/WEA2IS03.pdf

[4] J. Berlioz *et al.*, "Synchronous High-Frequency Distributed Readout for Edge Processing at the Fermilab Main Injector and Recycler," in *Proc. NAPAC'22*, Albuquerque, NM, USA, Nov. 2022, pp. 79–82. doi:10.18429/JACoW-NAPAC2022-MOPA15

[5] K. Hazelwood *et al.*, "Real-Time Edge AI for Distributed Systems (READS): Progress on Beam Loss De-Blending for the Fermilab Main Injector and Recycler," in *Proc. IPAC'21*, Campinas, SP, Brazil, Aug. 2021, paper MOPAB288, pp. 912–915. doi:10.18429/JACoW-IPAC2021-MOPAB288

[6] M. Thieme *et al.*, "Semantic Regression for Disentangling Beam Losses in the Fermilab Main Injector and Recycler," in *Proc. NAPAC'22*, Albuquerque, NM, USA, Nov. 2022, pp. 79–82. doi:10.18429/JACoW-NAPAC2022-MOPA15

[7] M. Ibrahim *et al.*, "Preliminary Design of Mu2E Spill Regulation System (SRS)," in *Proc. IBIC'19*, Malmö, Sweden, Nov. 2019, pp. 177–180. doi:10.18429/JACoW-IBIC2019-MOPP033

[8] *DOE Public Access Plan*. https://www.energy.gov/doe-public-access-plan

[9] Department of Energy, Office of Science. "Data, Artificial Intelligence, and Machine Learning at DOE Scientific User Facilities, DOE National Laboratory Program Announcement Number: LAB 20-2261." (2020), https://science.osti.gov/-/media/grants/pdf/lab-announcements/2020/LAB_20-2261.pdf