This space left intentionally blank
Please do not adjust your projector.

DAQ Group

Hall and group leaders meeting

June 4th 2019

Topics

Streaming readout progress.

Online and offline computing and networking.

Other projects on backup slides.

Streaming ReadOut Workshop IV – Camogli, Italy

- Two day workshop covered:
  - Verification/Validation techniques for Streaming Readout
    - Contributions from PANDA, ALICE, CLAS12, BDX
  - Electronics – Front End, Timing and Synchronization
    - ASIC and Micro-Electronic developments, Ultra Fast Silicon Detectors for Timing/Tracking, DAQ Electronics for Large Detectors[Industry Partners – CAEN], FADC250 in Streaming mode with Ethernet, and Streaming Readout Timing/Synch and TDC
  - Streaming Readout Software
    - Real-Time Analysis at LHC, Software Consortium report, TriDAS for EIC, Prototype [Protocol] results
  - Streaming Readout for EIC Detectors
    - TOPSIDE, JLEIC, eRHIC/FELIX-DAq, eRD23 proposal discussion

https://agenda.infn.it/event/18179/overview

Streaming ReadOut Workshop IV – Camogli, Italy

- Broadly and generally, the presentations were very good and in particular it was encouraging to see that other groups have implemented SRO techniques. PANDA and ALICE are two examples where SRO hardware and software have been implemented.

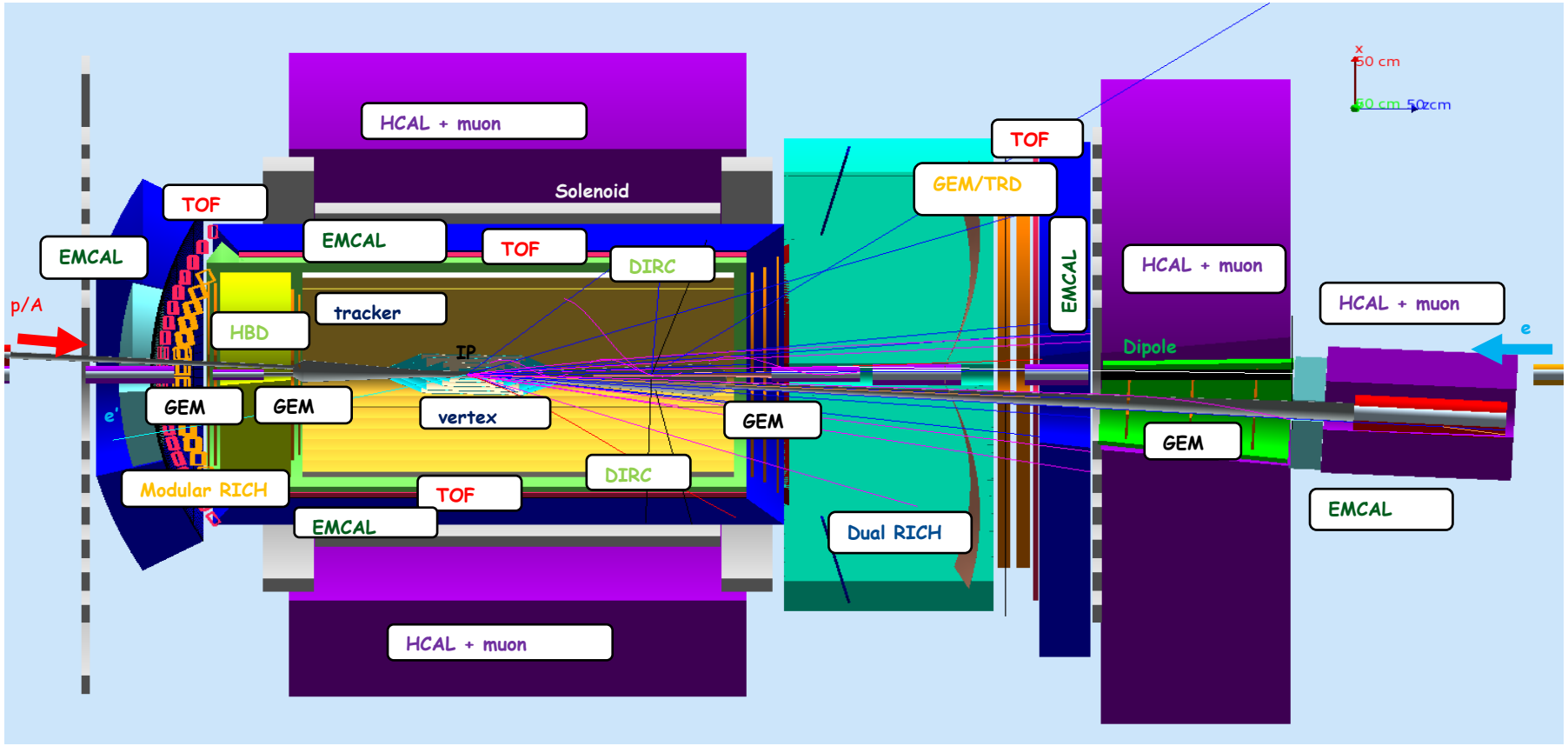
- The software work and report from the consortium were very informative plus we heard about very new micro-electronics [ASIC] and developments with Ultra-Fast Silicon detectors for timing/tracking during the hardware session.

- All of the EIC session included presentations that were similar in terms of channel count, data rates and methods for DAQ. The eRHIC/FELIX solution is advanced because the hardware will be used for sPHENIX.

- Definitions for the SRO data streaming protocol were presented and discussed. This protocol including other specific ‘standards’ for hardware and software will have to be documented and circulated soon. Good progress since 2018-December’s workshop.

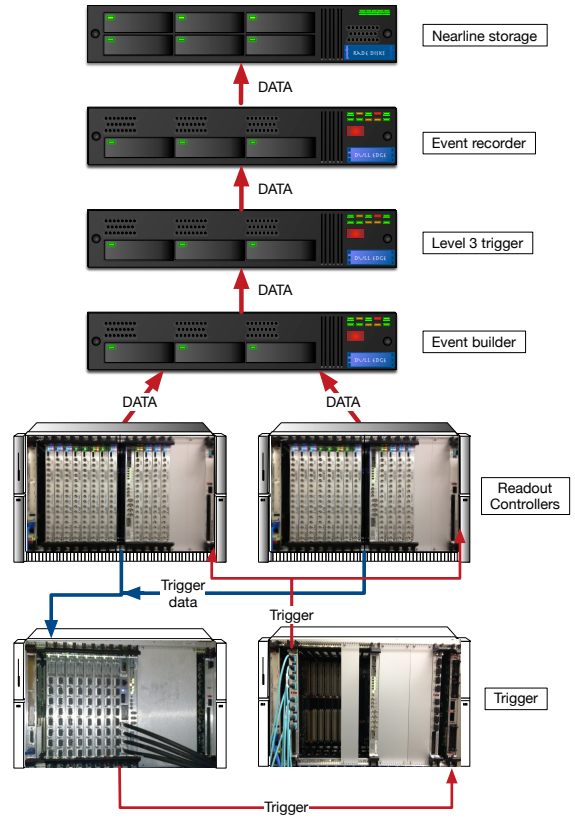
Example EIC detector design

- Just counting labels on the diagram there are ~25 detector packages.
  - Wide range of response times for the detector types.
- The largest single channel count is the Vertex Detector.



What happens if we use traditional DAQ - crates

- Back of the envelope calculation ignoring the Vertex detector (which could be 20-50 M channels).
- The rest of the detector is ~1M channels.
  - CLAS12 : ~90k channels read by 100 ROCs
  - GLUEX : ~40k channels read by 50 ROCs
  - Average ~1 ROC per 1000 channels, seems like a lot of channels per ROC but is dominated by high channel count detectors.
  - EIC detector would be ~1,000 ROCs.
    - We need to distribute triggers to 1000 devices.
    - We could have up to 1000 devices contributing signals to the trigger.
- Assume average 1% occupancy.
  - Vertex detector rate ~240 GB/s. (yes bytes)
  - Rest of the detector ~5 GB/s total.
    - Fair agreement with CLAS12 and GLUEX if we were to scale them up to 100 kHz and 1% of 1M channels.



The Vertex detector

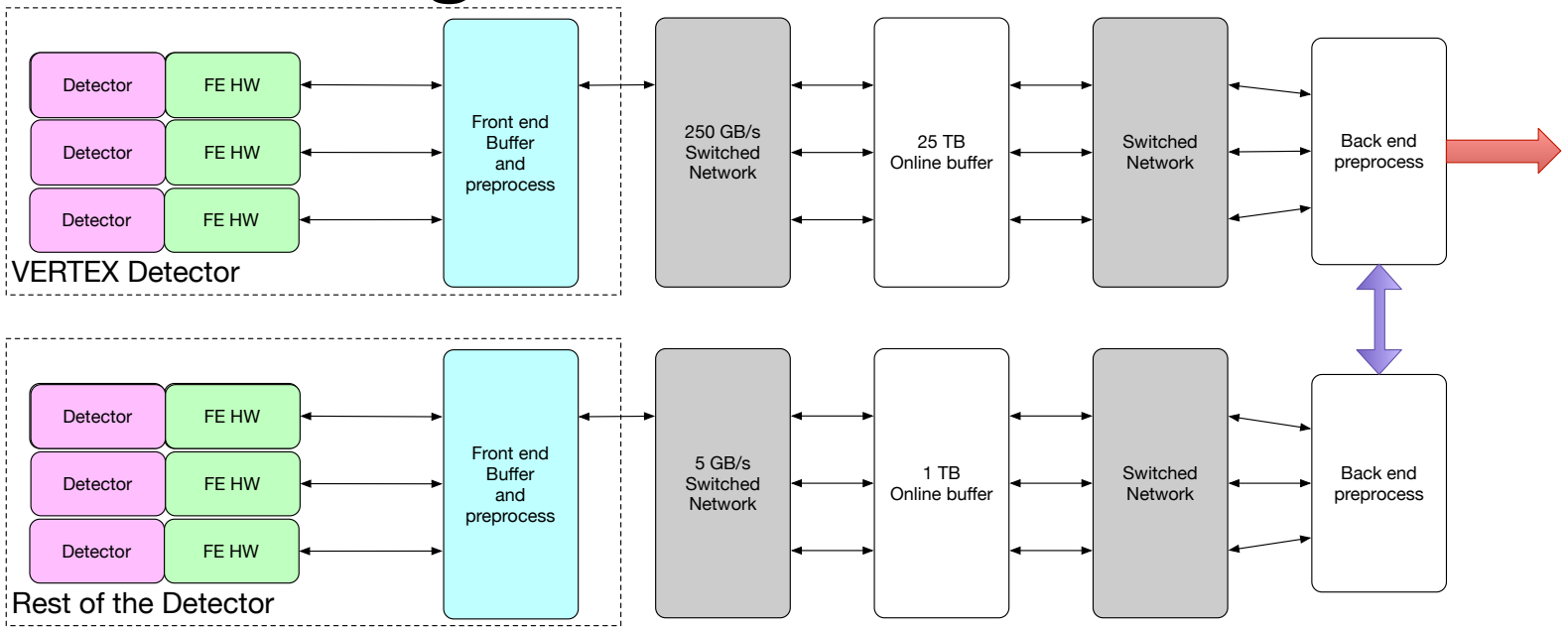
- How would you read out a detector that generates 240 GB/s ?

- The only way that even remotely makes sense is massive parallelism.
  - Split the detector into small regions and read those out in parallel.
  - No sensible way to sync the different regions in real time without spending a lot on electronics.

- Aim to reduce the rate to storage by using data from the rest of the detector to define regions of interest.
  - Have to hold on to the Vertex Tracker data until R.O.I. can be identified.

- Dealing with the Vertex Tracker dominates the design of the DAQ.

Dealing with the Vertex detector

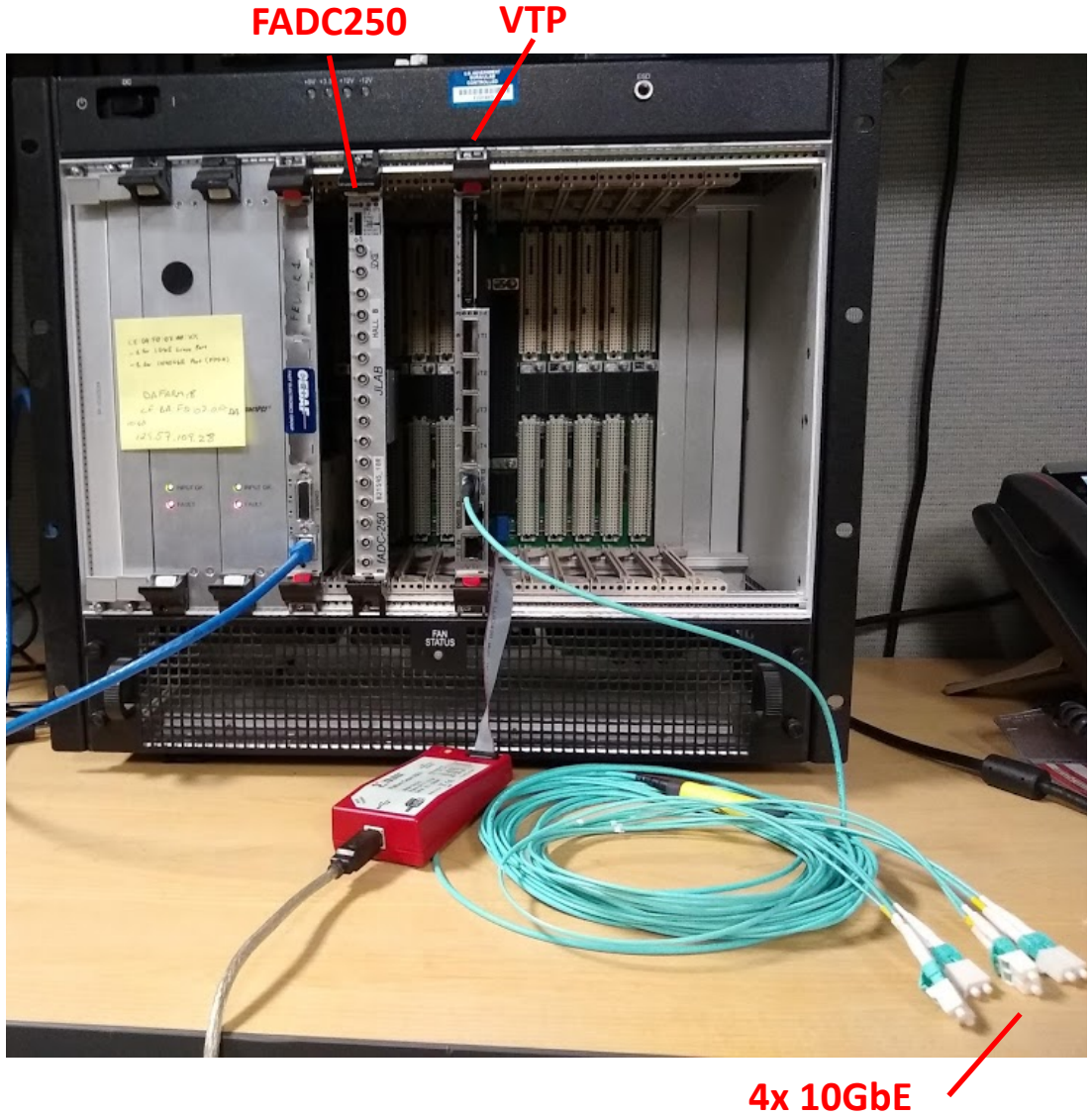


- Vertex Detector is read in parallel streams into online buffers.
  - Say 25 front end buffers at 10 GByte/s (Using today's 100 Gbit/s HW).
  - Main Online buffer is 25 nodes with 1TB of memory each ~100s buffer time.
- Rest of detector streams to a smaller online buffer, 5 GB/s total, single 1TB buffer ~200s buffer time.
  - Actually almost identical to 1/25th of the Vertex system – so 26 main streams in total.
- Identify regions of interest in Vertex Detector : 4D regions = 3D volume in detector + time range.
- Vertex Detector back end processors pull ROI data from online buffer. Unwanted data is discarded

Ben's talk - Read FADC250 via VTP - Current Test Setup

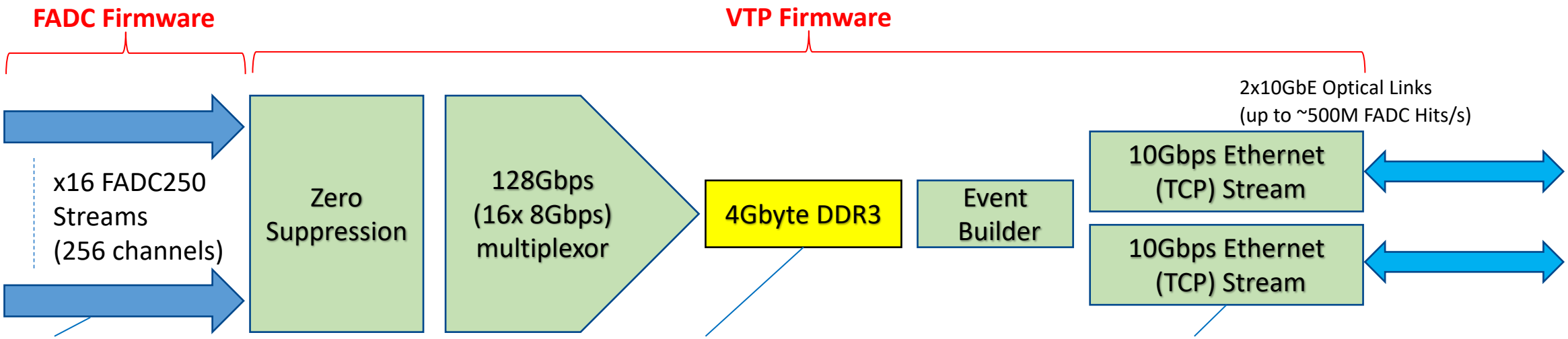
- Current test setup sits in my office:
  - Small VXS Crate
  - 1 FADC250
  - 1 VTP
  - Old PC w/10GbE (Mellanox ConnectX-3)

- Will move to INDRA-ASTRA lab soon
  - Expanding to 16 FADC250 modules
  - High performance servers



Ben - Firmware Development

- FADC250 – no firmware development needed
  - Reusing trigger path, which discriminates and provides pulse time and charge
- VTP – nearly all firmware completed



16x FADC 8Gbps streams, each:
- 16 channels
- 32ns double pulse resolution
- 4ns leading edge timing
- Pulse integral

Large buffer:
- 200Gbps bandwidth
- Allows high burst rates
- (doesn't need to be this big, but it's what the hardware has, so might as well use it)

TCP Stack:
- Hardware accelerated
- Up to 4 links can be used
- Could be a single 40GbE (if we could afford the TCP stack IP – not worth it for R&D)

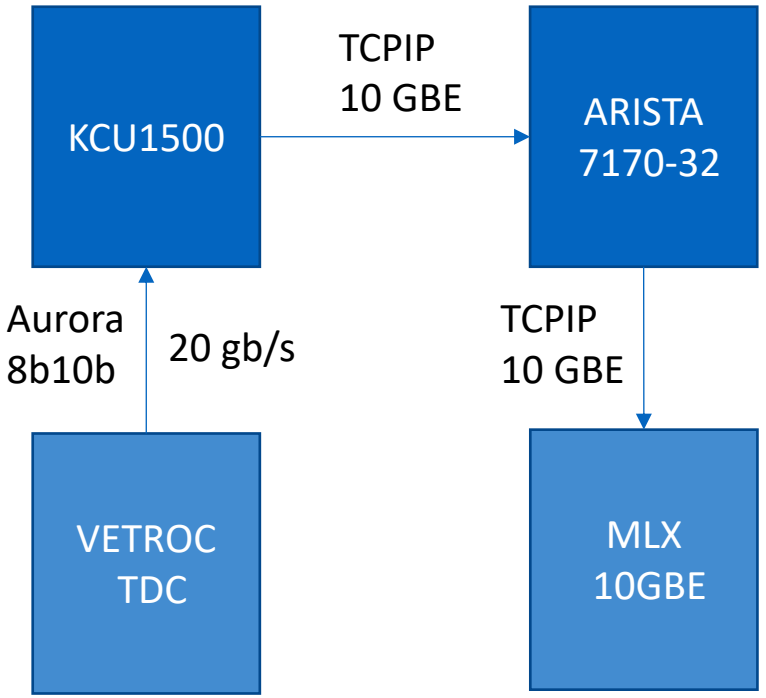
Ben's talk continued

- TCP hardware accelerated stack is probably one of the trickier parts that luckily we have a vendor providing. We have found a number of issues with the IP, but the vendor has been working with us to resolve them – shouldn't prevent us from reach test goals.

- Delays due to CLAS12 & HPS experiment preparations, but these will be complete in the next few weeks so Ben can actually spend good time to wrap up this project!

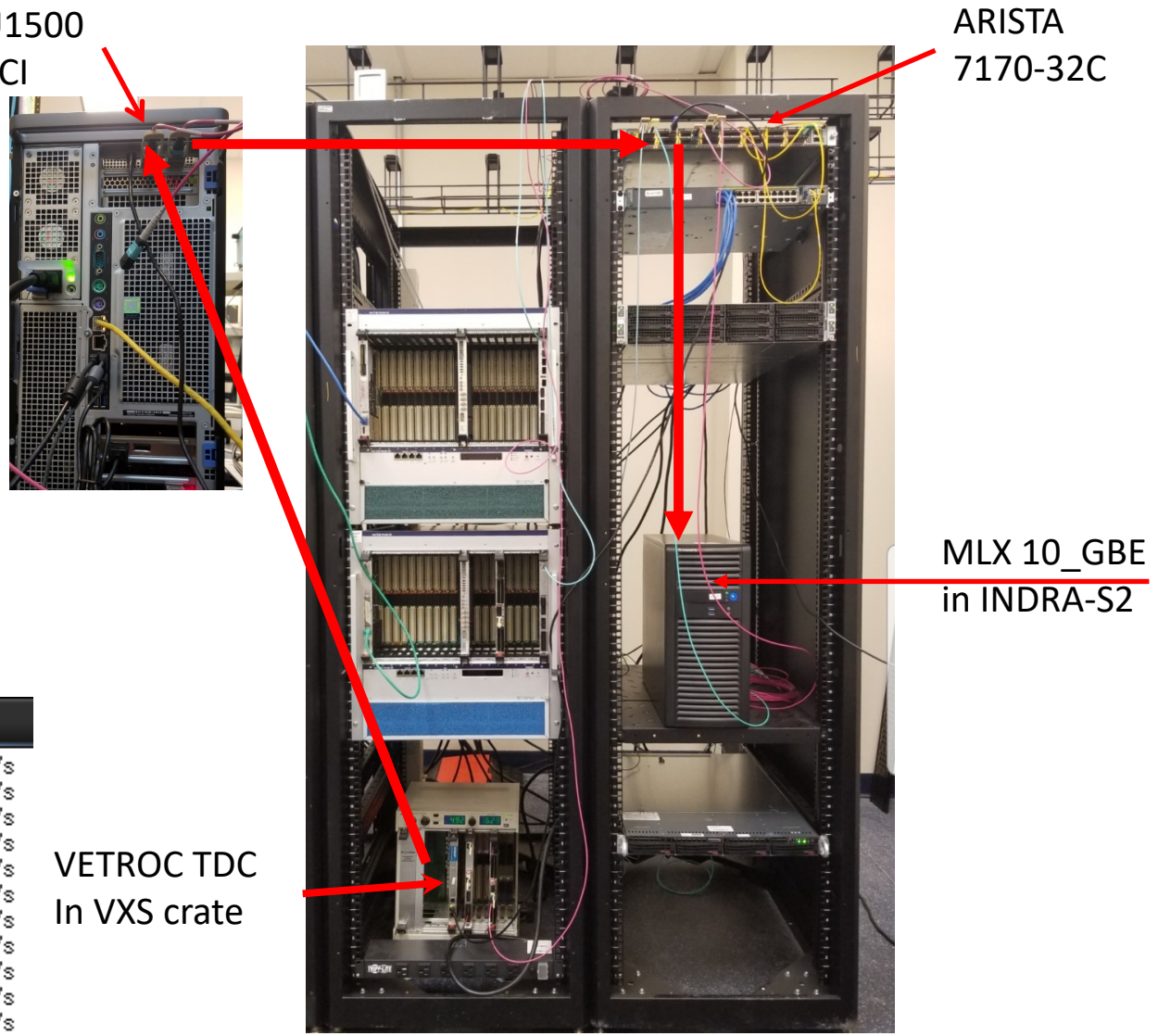
- Making progress towards FADC250 crate streaming over Ethernet using TCP.
  - expected to have a functional demonstration this summer!

William - Streaming Readout of a VETROC 128 chan streaming TDC



Graham's
Stream_Router
Software
~400 Mbyte/s

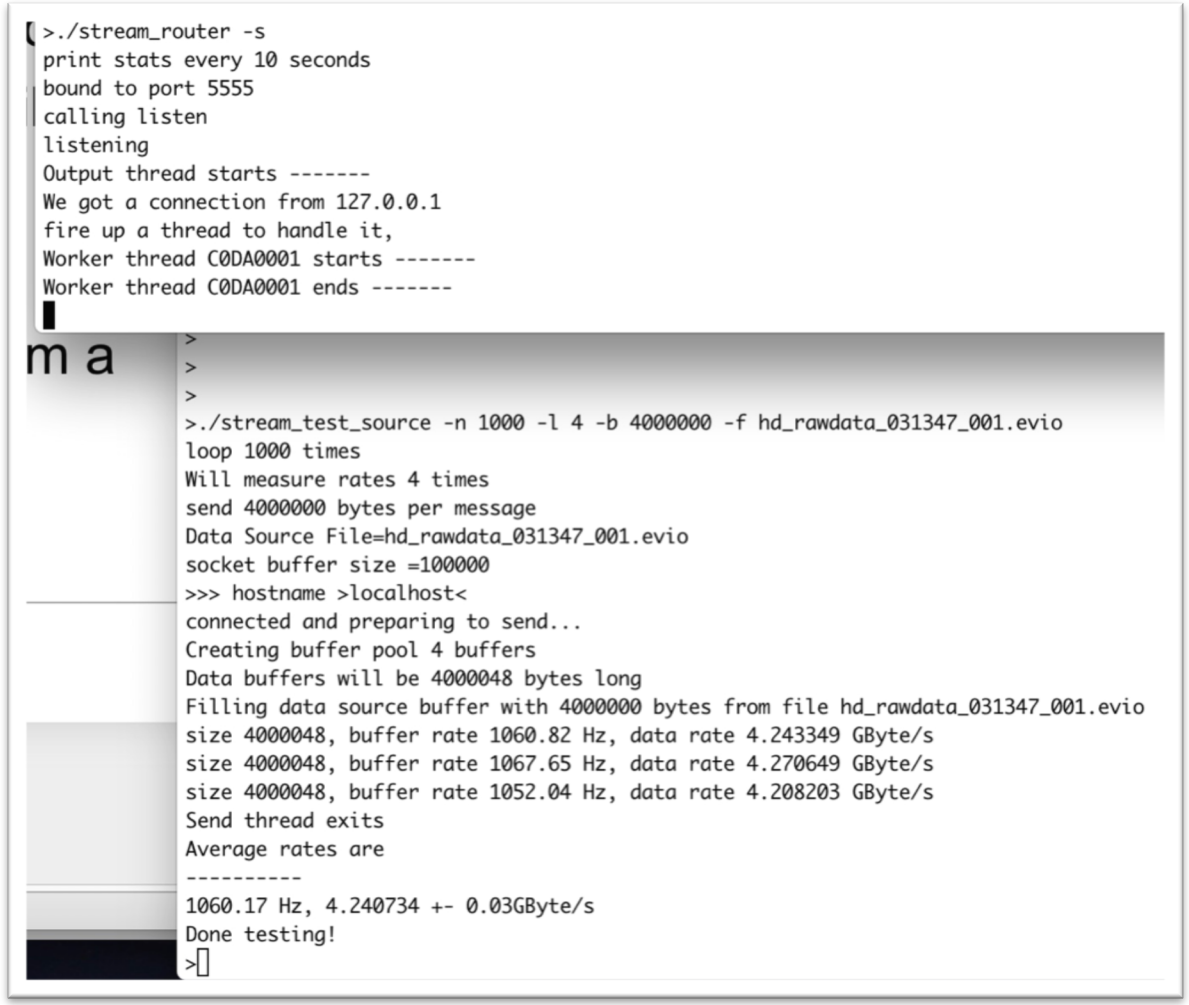
```
xterm (on indra-s2)
ID CODA0048 - buffer rate 379895.25 Hz, data rate 0.407248 GByte/s
ID CODA0048 - buffer rate 391878.30 Hz, data rate 0.420094 GByte/s
ID CODA0048 - buffer rate 381900.90 Hz, data rate 0.409398 GByte/s
ID CODA0048 - buffer rate 389568.43 Hz, data rate 0.417617 GByte/s
ID CODA0048 - buffer rate 382177.78 Hz, data rate 0.409695 GByte/s
ID CODA0048 - buffer rate 386047.66 Hz, data rate 0.413843 GByte/s
ID CODA0048 - buffer rate 369711.18 Hz, data rate 0.396330 GByte/s
ID CODA0048 - buffer rate 357362.87 Hz, data rate 0.383093 GByte/s
ID CODA0048 - buffer rate 376956.18 Hz, data rate 0.404097 GByte/s
ID CODA0048 - buffer rate 366132.41 Hz, data rate 0.392494 GByte/s
ID CODA0048 - buffer rate 398968.96 Hz, data rate 0.427695 GByte/s
ID CODA0048 - buffer rate 390347.00 Hz, data rate 0.418452 GByte/s
ID CODA0048 - buffer rate 393121.97 Hz, data rate 0.421427 GByte/s
ID CODA0048 - buffer rate 359522.01 Hz, data rate 0.385408 GByte/s
```



Streaming test code (not shown at workshop)

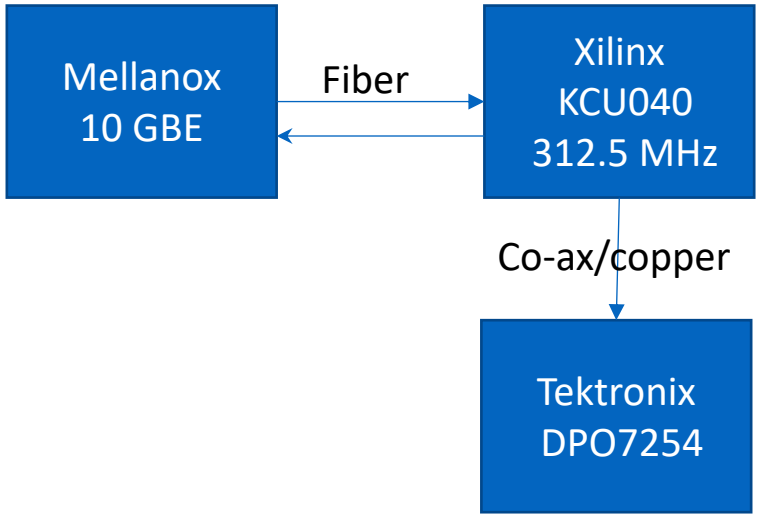
Three pieces of test code:

- stream_test_client
  - Simulated data source.
  - Sends fixed length data blocks.
  - Data from file or random numbers.
  - Sends multiple blocks to allow rate tests.
- stream_router
  - Receives TCP packets from data source.
  - Optionally route to ZeroMQ subscriber.
  - Measures throughput.
- stream_test_subscriber
  - Example of how to receive data from a stream_router.



William's talk – Clock Recovery study

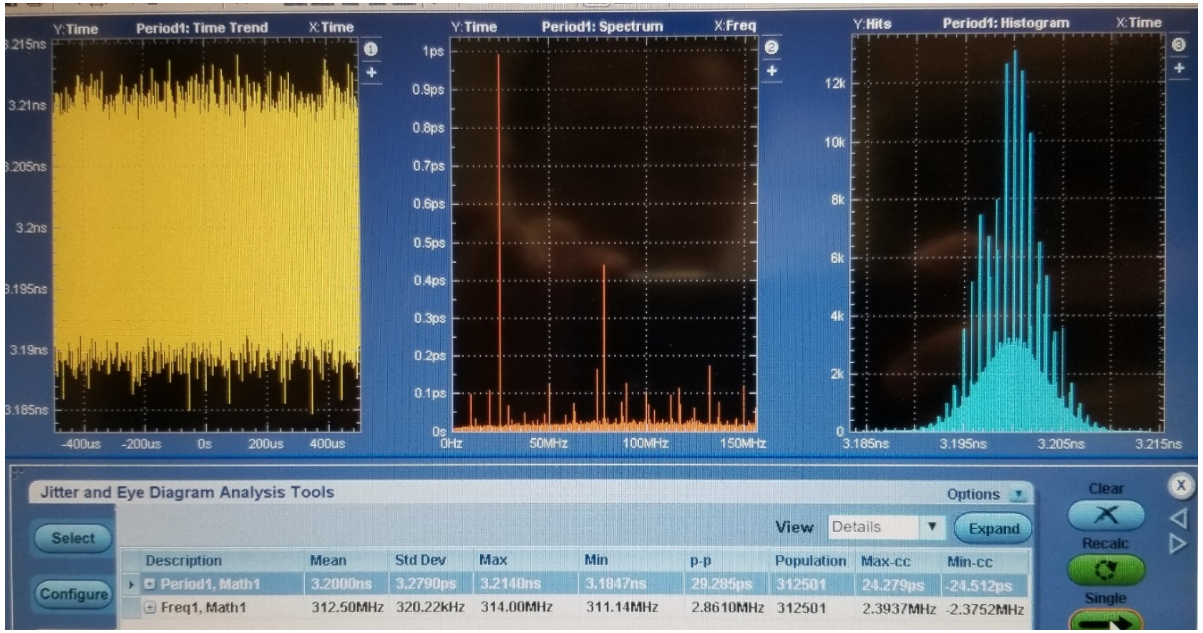
How to do the timing?
Can a low jitter clock can be recovered from the fiber data links?



Mellanox
(10 GBE)

Xilinx KCU105
(UltraScale Kintex)

Recovered clock to Tektronix DPO7254



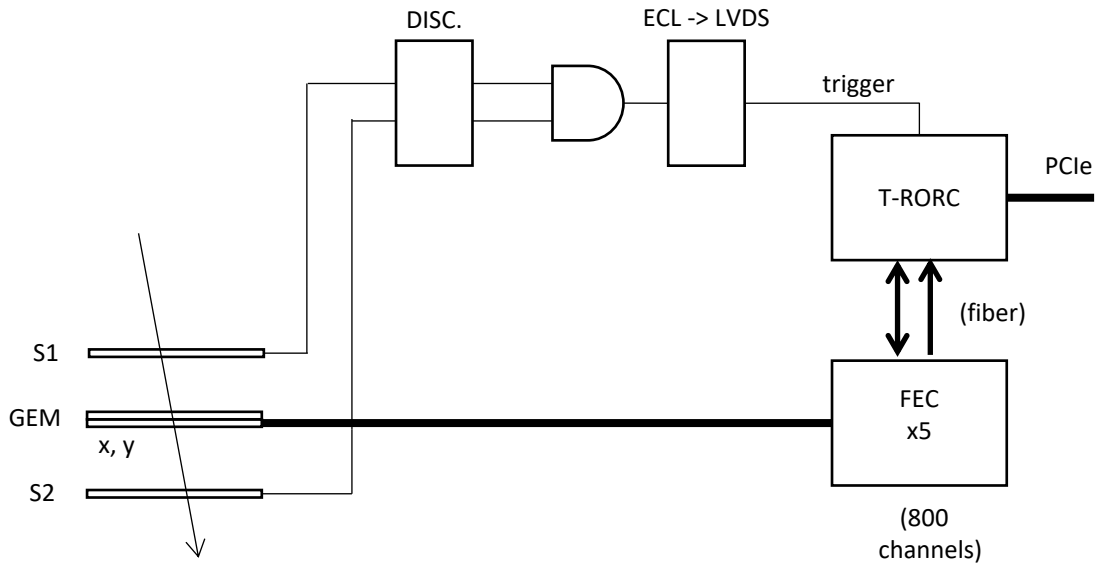
Recovered clock period: 3.2000ns
Std Dev: 3.28 ps
Enabling the Mellanox data on same link, Std Dev → 3.30 ps
Enabling the KCU040 data on same link, Std Dev → 6.42 ps

Scope single pulse period measurement: Std Dev: 6~7 ps
DPX mode (200 Gsample/s): 700~900 fs

Answer : yes!

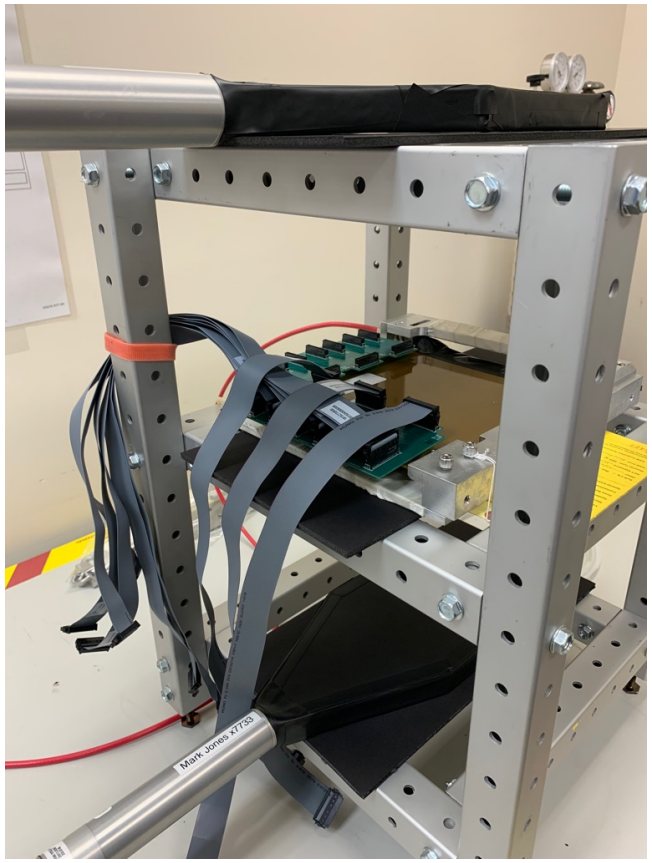
Streaming GEM RO cosmic Ray Test Setup – Ed & Eric

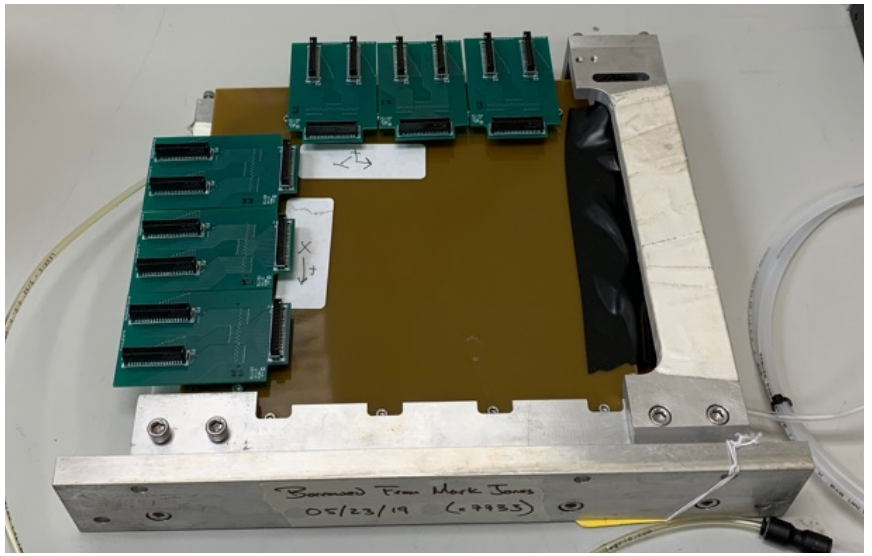
- Data is continuously streaming from the FECs to the T-RORC.
- Most of the 9 Gb/s data sent from each FEC to the T-RORC (45 Gb/s total) consists of sync packets that serve to keep the serial links from the SAMPAs active when there is no hit data to send.
- With the current firmware the T-RORC transmits all of this data directly to PC memory.
- Trigger zero suppresses - T-RORC captures a time window after the trigger.
- Plan to modify the firmware of the T-RORC to remove need for this trigger.
  - Then we can acquire data in a truly continuous fashion.

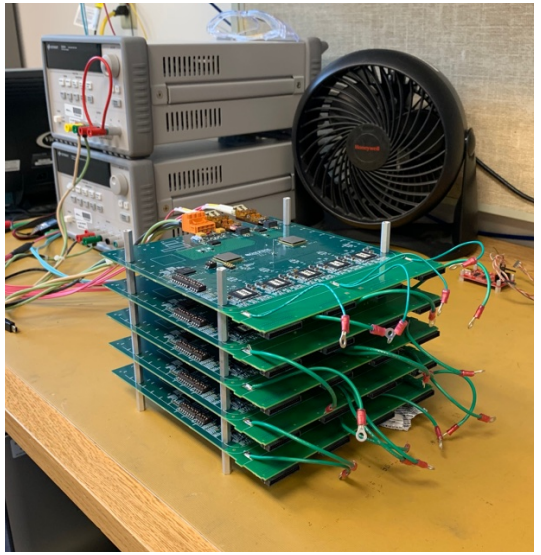


S1, S2	– plastic scintillators
GEM	– 1x, 1y plane 384 ch. Each
T-ROC	– ALICE /ATLAS Readout receiver
FEC	– ALICE Front End Card (JLAB version), 5 SAMPA chips = 160 channels

Test stand pictures







Top row left to right :

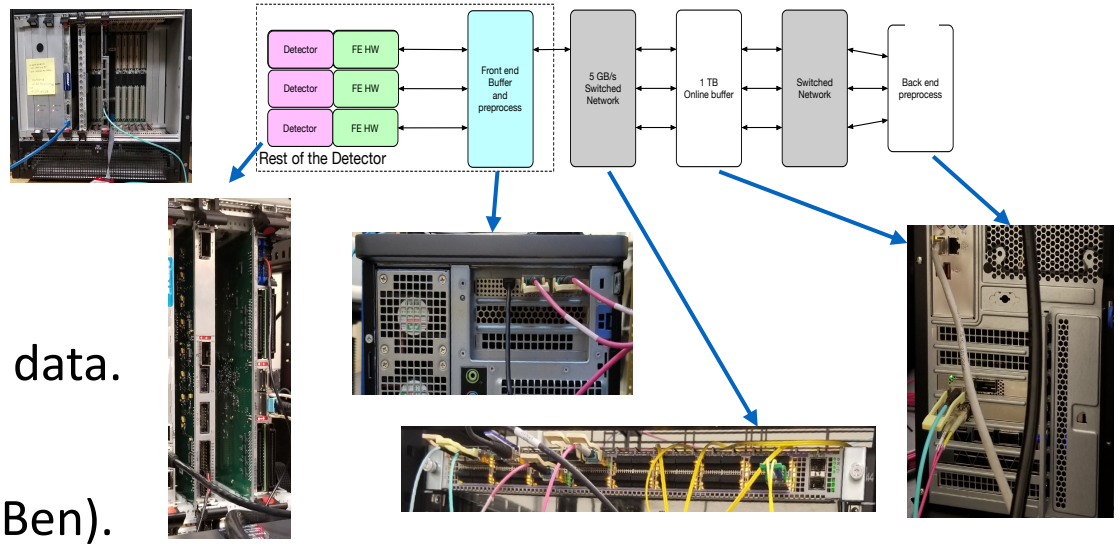
- Test stand on bench in INDRA lab
- Test stand closeup.
- Closeup of GEM before mount

Bottom right :

- Front End Card stack.

So where are we now?

- In the EIC streaming DAQ design key elements are:
  - A data source outputting on fiber.
  - A front end buffer with FPGA.
  - A high speed low latency network.
  - An online compute resource to buffer and process data.
  - A timing system.
- We are testing streaming readout of 250 MHz fADCs (Ben).
- We are testing a GEM detector readout in the INDRA lab. (Ed and Eric).
- We are testing clock distribution over network. (William)
- In the INDRA lab at JLab we have a test stand using
  - a VETROC TDC to provide a Front End data source
  - A Xilinx FPGA KCU1500 PCI board as a front end buffer and preprocessing device.
  - A Linux PC, with 100 Gbit/s network link as the online buffer/back end processing node. (William)



Computing - networking

- The diagram to the right shows how the lab is connected to the outside world via ESNet.
- JLab is connected to the outside world via a 10 Gbit/s link.
  - Link connects to the ELITE Metro ring shared by sites in the local area.
  - ELITE is linked to ESNet via a 10 Gbit/s links to Atlanta and Washington (for redundancy).
  - We could go up to 20 Gbit/s by using the backup path.
- We are hopeful that ESNet will upgrade us to 100 Gbit/s.
  - Will need an ELITE upgrade since we would be able to fill ELITE ourselves and it is a shared resource.

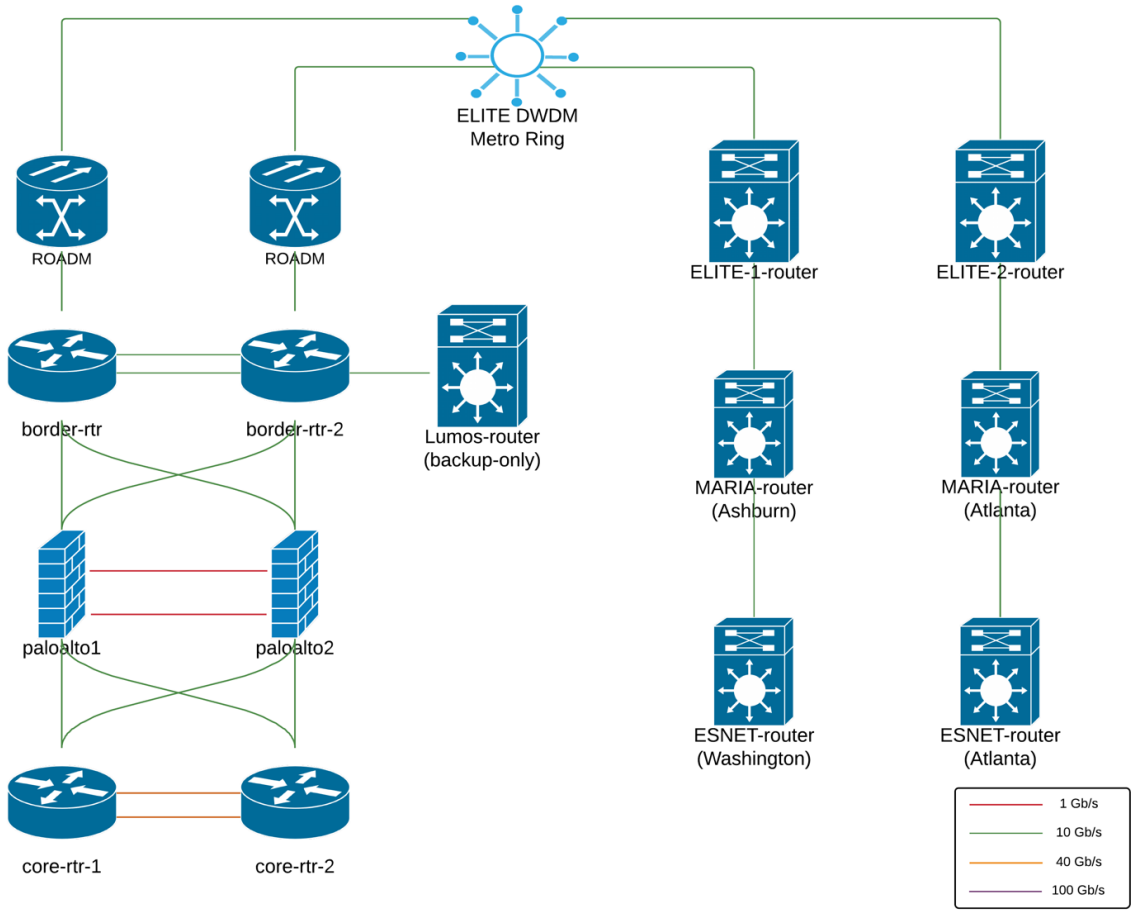
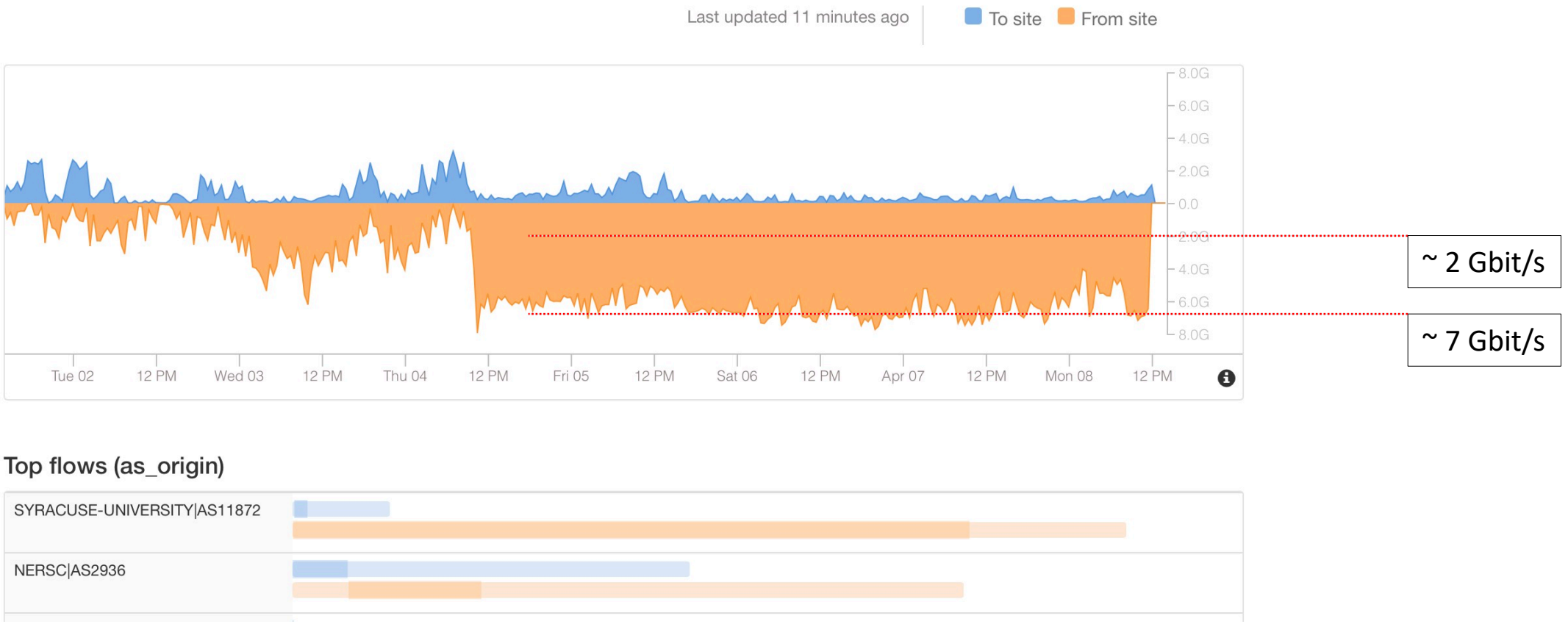


Figure 12 Layer 2 path to ESNet

Data transfers offsite on April 4th 2019

- Timeline starts with David Lawrence running GLUEX reconstruction at NERSC – very choppy data flow. ~2 Gbit/s
- Around noon Thomas Britton et al start simulation at Syracuse. ~7 Gbit/s
  - Simulation was artificially high due to misconfiguration of job. (Same file cached on disk sent with every job).



Data transfer onsite – few months ago.

- Halls connected to the central Lustre filesystem via network.
  - Data copied from Lustre to tape.
- Local jobs run against data staged on Luster.
- System not designed to run jobs on OSG or NERSC.
- Data sent offsite by pulling from tape to Lustre then pulling from Lustre using gateway nodes.

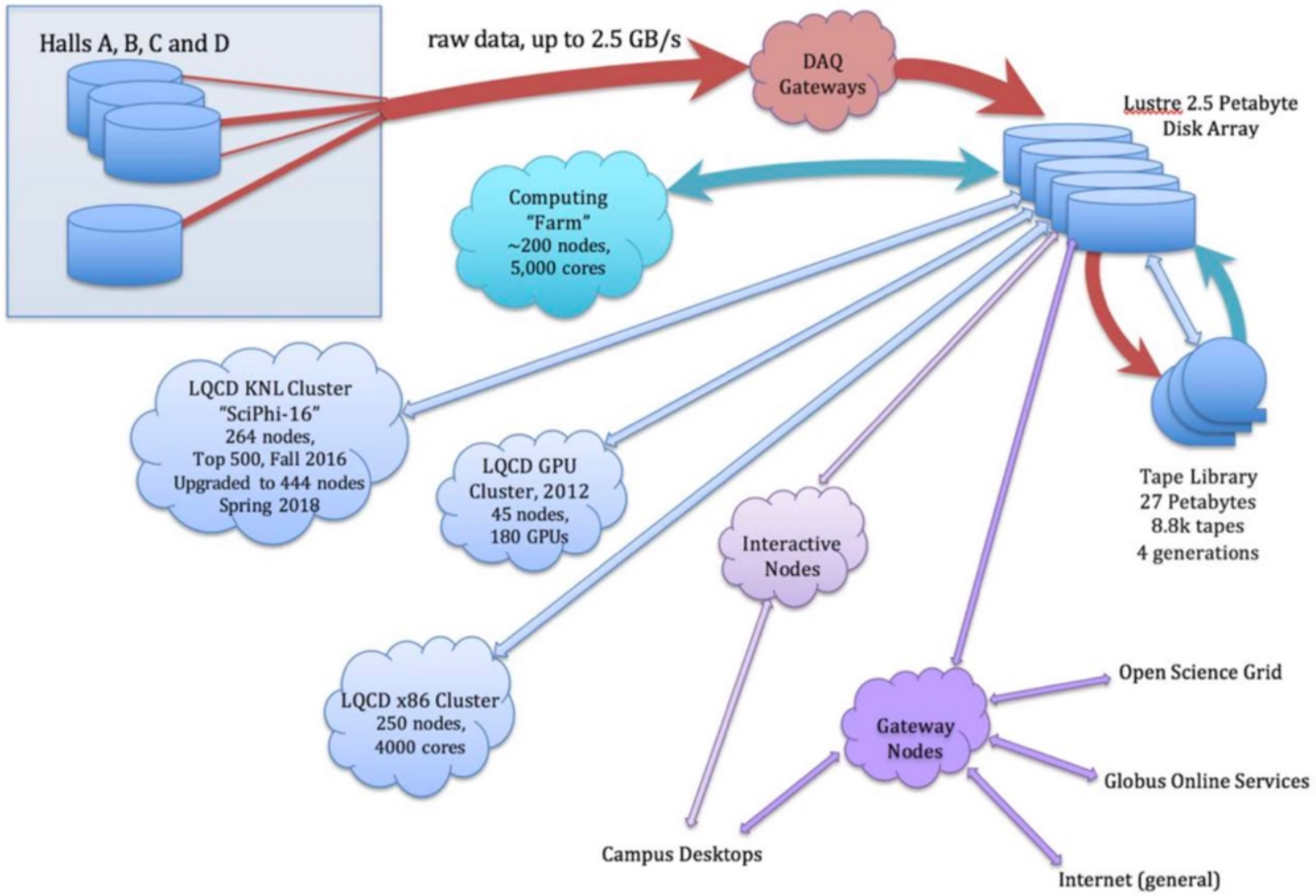
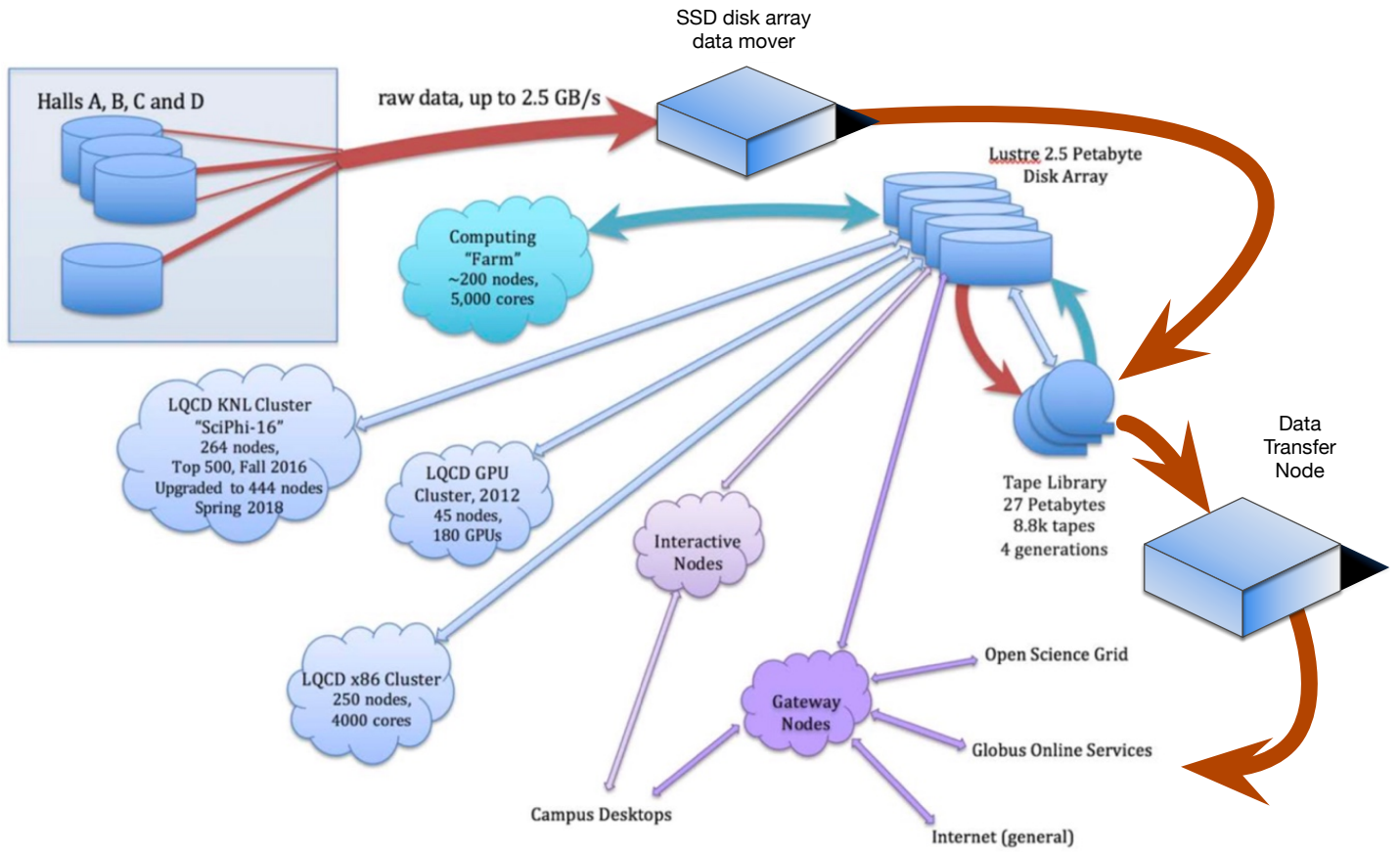


Figure 7 Jefferson Lab local compute and storage

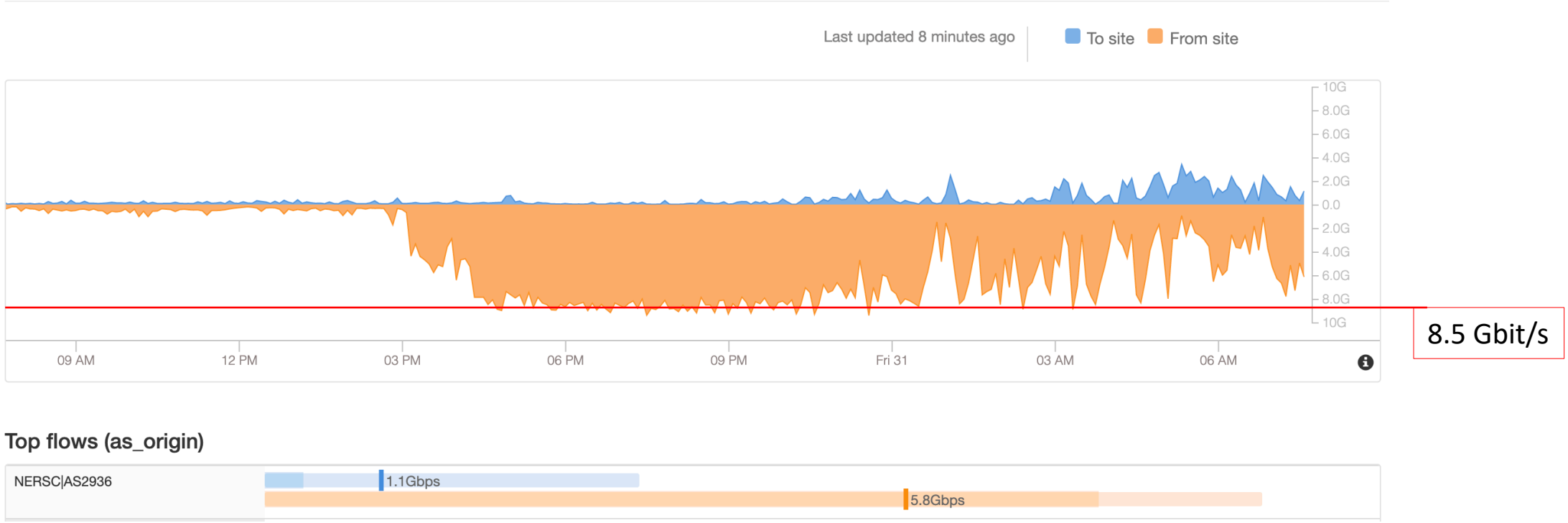
Infrastructure upgrades.

- We added a SSD disk array to stage raw data and speed up writing to tape.
  - Data goes to tape without touching Lustre.
- Bought HW to add a second one as a backup.
- Borrowed the spare to set up a Data Transfer Node (DTN) to pull data from tape and send directly to NERSC and OSG.



Thursday May 30th - Hot off the press

- David Lawrence is able to get a peak of 8.5 out of the available 10 Gbit/s.
  - The sawtooth stuff overnight is the result of (1) tuning up the tape access and (2) being on the edge of the number of SSDs we really need. Fixes for both of those are in the pipeline.
- We put out a REQ for a permanent, designed for purpose, DTN.



Backup slides

- The following slides report on other work that the rest of the group did…

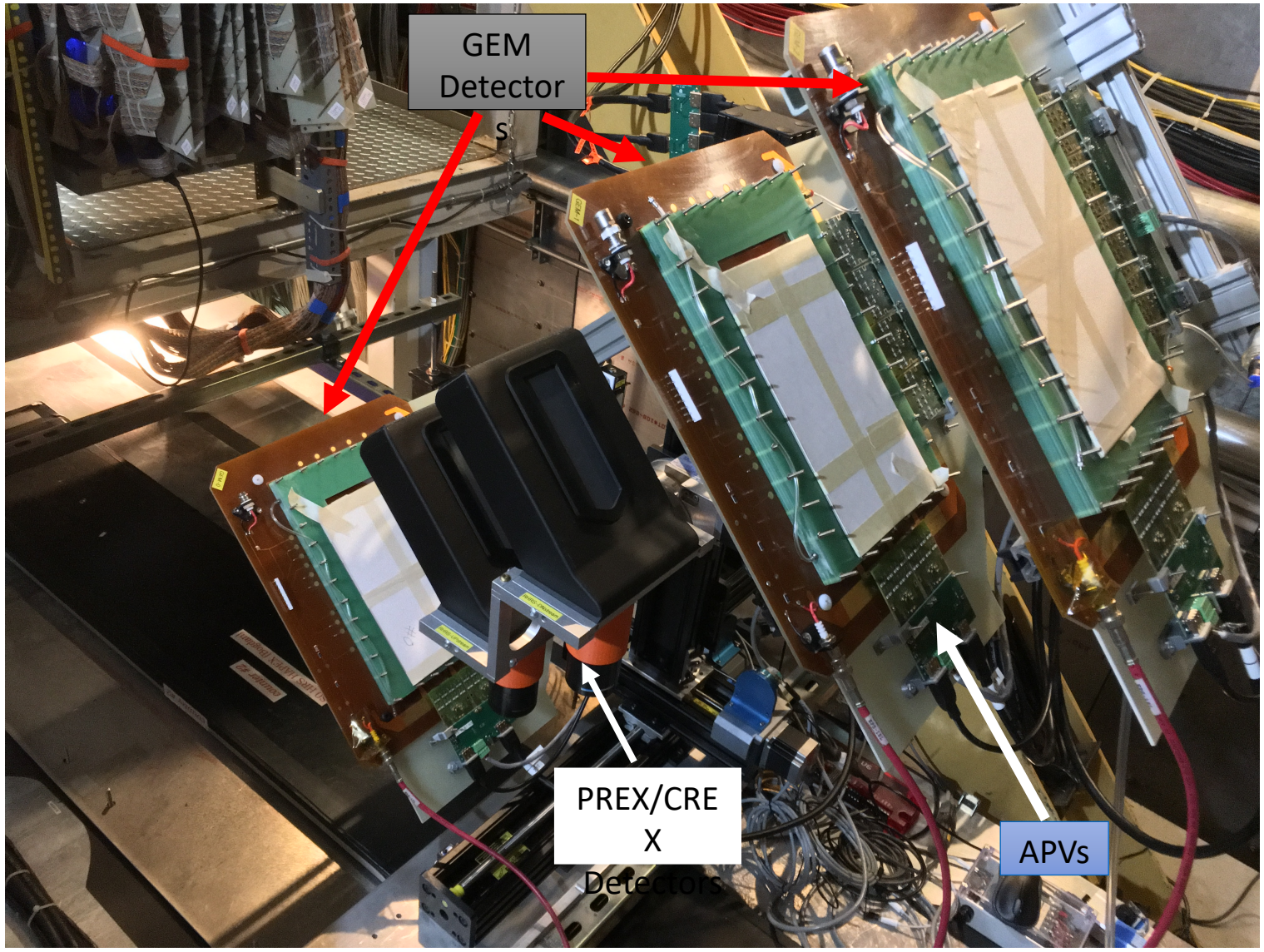
Hall support - William

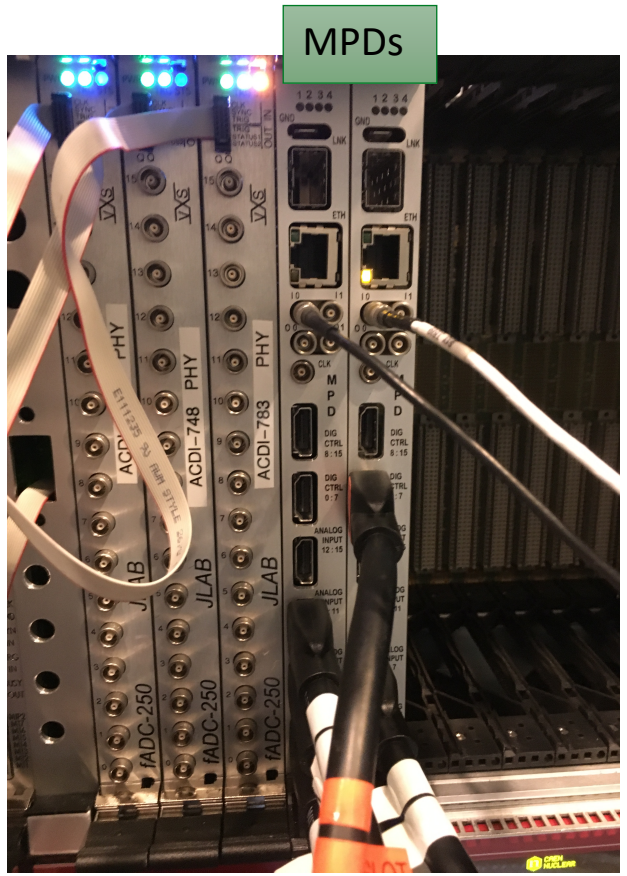
- HPS run in Hall B preparation: Will use the PCIexpress TI instead of the TI_functions built-in the ATCA daq board (from SLAC).
  - Ready for Summer running.
- NPS (for Hall C) configuration study: TI pass through function study, can we allow the trigger distribution system to be expanded to more than 9 crates without requiring TS/TD boards and the trigger distribution crate?
  - The firmware is compiled, needs NPS setup tests.

Hall support – Bryan Moffit

- Multi-Purpose Digitizer (MPD):
  - Currently used for configuration and readout of APVs (which readout GEM detectors). Uses HDMI connections for digital control (i2c for configuration, trigger, sync, and clock) and analog readout.
  - Will be used in upcoming Hall A - PREX/CREX for profiling the elastic/quasi-elastic distribution in the HRS focal plane. GEM data will be compared to HRS Vertical Drift Chamber data. Future GEM detectors to use this sort of DAQ setup for SuperBigbite experiments.
  - Debugging MPD firmware and software driver.
  - Debugging and understanding the configuration and readout of APVs.
- Hall A Moller DAQ upgrade:
  - Upgrading from FADC250-V1 to FADC250-V2.
  - Upgraded module, firmware, and software provide greater flexibility for scaling various detector logic combinations using a LUT.
- CentOS7 for Intel-based VME Controllers:
  - Moving from proprietary GE/ABACO kernel drivers to in-tree linux kernel driver for better support of current Linux kernels.
  - Provides some support for new (not quite ready for production) FPGA based VME-PCI bridge (so called VIVO FPGA), as well as Tempe and Universe VME bridges.
  - Newer OS will support both i686 and x86_64 interfaces to kernel driver.

Hall A - PREX/CREX HRS Focal Plane Detectors





Hall support Ed

- Trigger Supervisor Data Tagger Module (Hall A)
  - PREX-II requires the ability to confirm that data read from ROCs originated from the same trigger.
  - Idea is for the Trigger Supervisor to send the ROCs an *event ID* with each *Strobe* rather than an event type based on the trigger accepted – hardware generated 8-bit event number
  - When the complete event is built from the ROC event fragments, the *event ID*s from the distributed ROCs can be compared to verify synchronization.
  - It would be difficult to modify the existing TS to perform the function described above.
  - With help from Jeff Wilson of the FE group a modified TS design is ready for fabrication.

- FADC250 Moller Firmware Upgrade (Hall A)
  - With custom firmware a data acquisition system for the Hall A Moller Polarimeter is implemented based on a single FADC250.
  - Readout of multiple internal scalers is initiated by the receipt of a helicity trigger from the accelerator.
  - Readout of data based on user defined internally generated triggers (prescaled) enables a study of the polarimeter's systematics.
  - With Hai Dong (FE group) we have implemented and tested new user requested features of FPGA firmware.

What Vardan did and other stories.

- AFEC CODA Run-Control
  - Implement user addressed messages and fault-recovery-action suggestions in order to minimize DAQ related beam-time losses.
  - Working with Hall DAQ administrators to incorporate experiment specific auto-recovery actions.
  - Allow users to attach a user script to the DAQ “Reset” transition.
  - Migration to the JDK 11

- CLARA
  - Consulting and support
  - Develop an algorithm for complete DPE shutdown in case of a user engine failure (user need to classify engine failures and associated actions). This is a Hall-B request. Note that the notion of the platform shutdown is against of the streaming microservices philosophy (DPE is design to be fault tolerant and will survive user engine problems, unlike traditional frameworks, such as JANA).
  - Developing a SFCM (Spatial Fuzzy C Mean) algorithm based, generic cluster finding service that will run both on CPU and GPU. Tests will be done on the local qcd12kmi node. For the GPU development I will be using OpenACC and will require PGI or Cray compilers to be installed. I already requested to install a free version of the PGI compiler for initial development and tests.
  - Developing a generic ET service, that can be configured at runtime to customize bank structure for a specific data stream.
  - Changing CLARA native transient data format from Google’s Protocol Buffers to Apache Arrow.
  - Developing a generic convertor service that will convert EVIO to Apache Arrow.

- NERSC deployment
  - Created an account at NERSC and at Globus
  - Created jlab-clas12 endpoint at the Globus for CLAS12 data-set transfers
  - Installed CLARA and CLAS12 software stack at NERSC

CODA3 backend and EVIO - Carl

- Carl is making progress
  - EB builds at over 2.6 GB/s on gluonraid4 – not writing to file
  - ER writes stably at 1.7 GB/s
  - EVIO using a common file format with HIPO is working and performs stably.
  - Have tested in DAQ lab and on GLUEX online machines.
  - Need multiple threads in parallel to compress at this rate – 450 MB/s per thread.
  - Compression is clearly working but meaningful tests require data as close to real GLUEX data as possible so that the compressibility is the same.
  - Hoped to have this done by this presentation but Carl has had limited time.

- David Abbott has updated the CODA 2.6 EB to work with CODA 3 DAQ (interfaces to Run Control, EB and EVIO formats are all different).