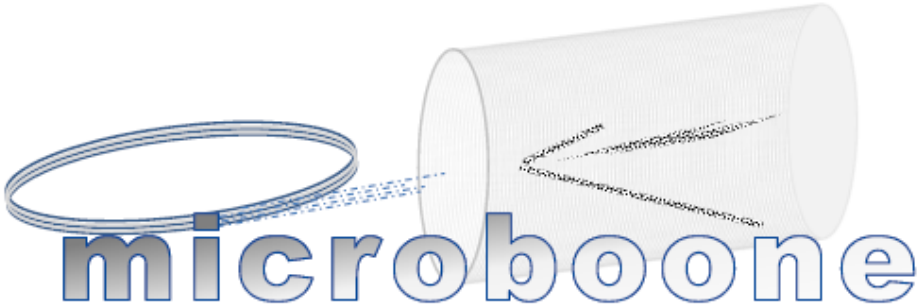


MicroBooNE DAQ Experience

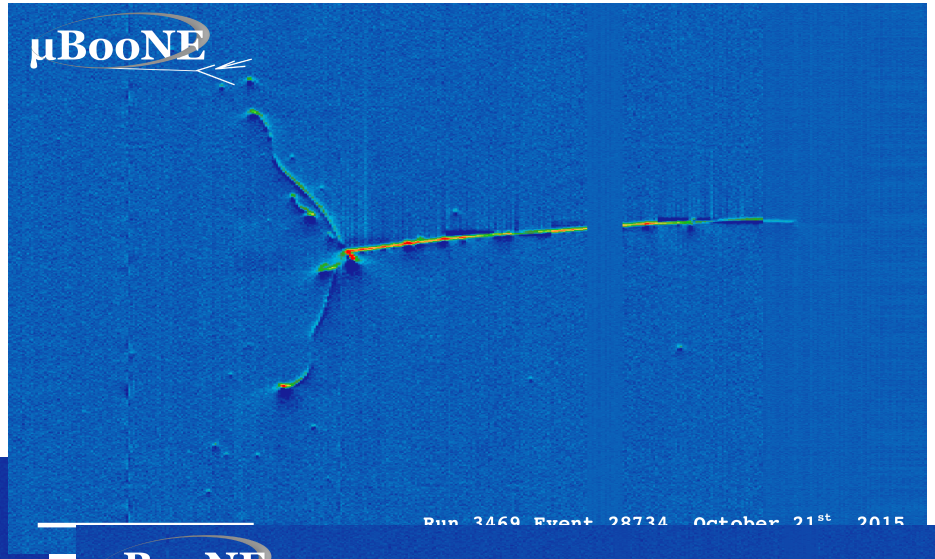
Eric Church, PNNL

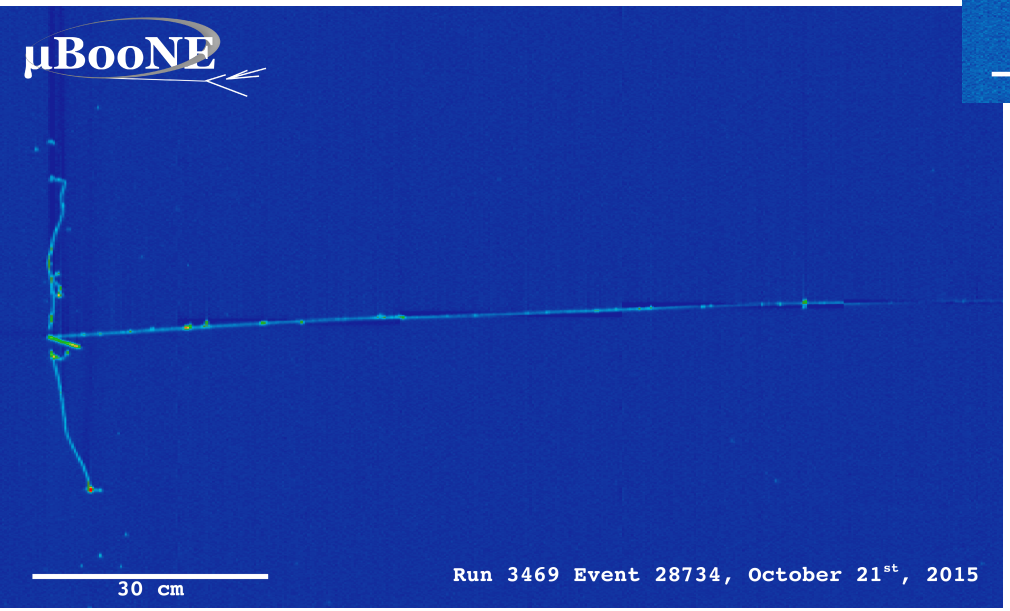
SBN/DUNE DAQ Meeting

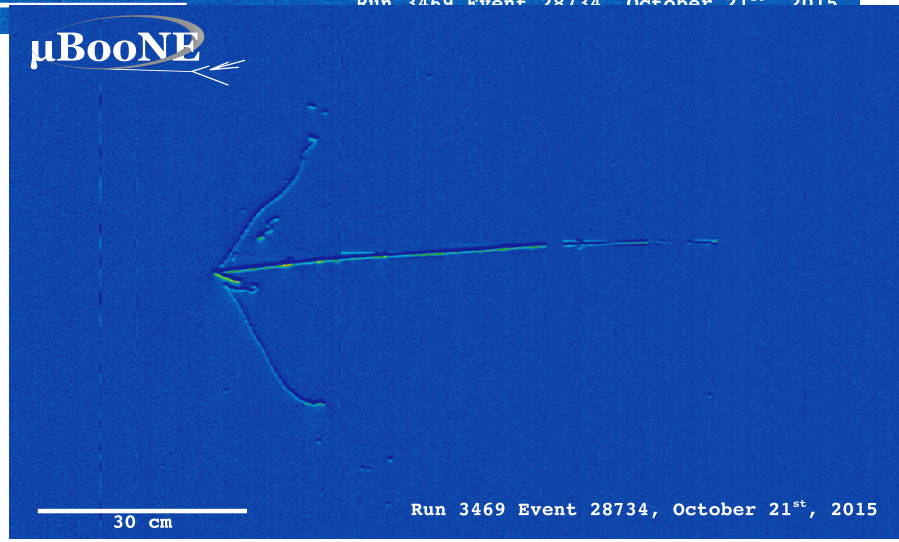
20-Nov-2015

MicroBooNE Status

- MicroBooNE has data!

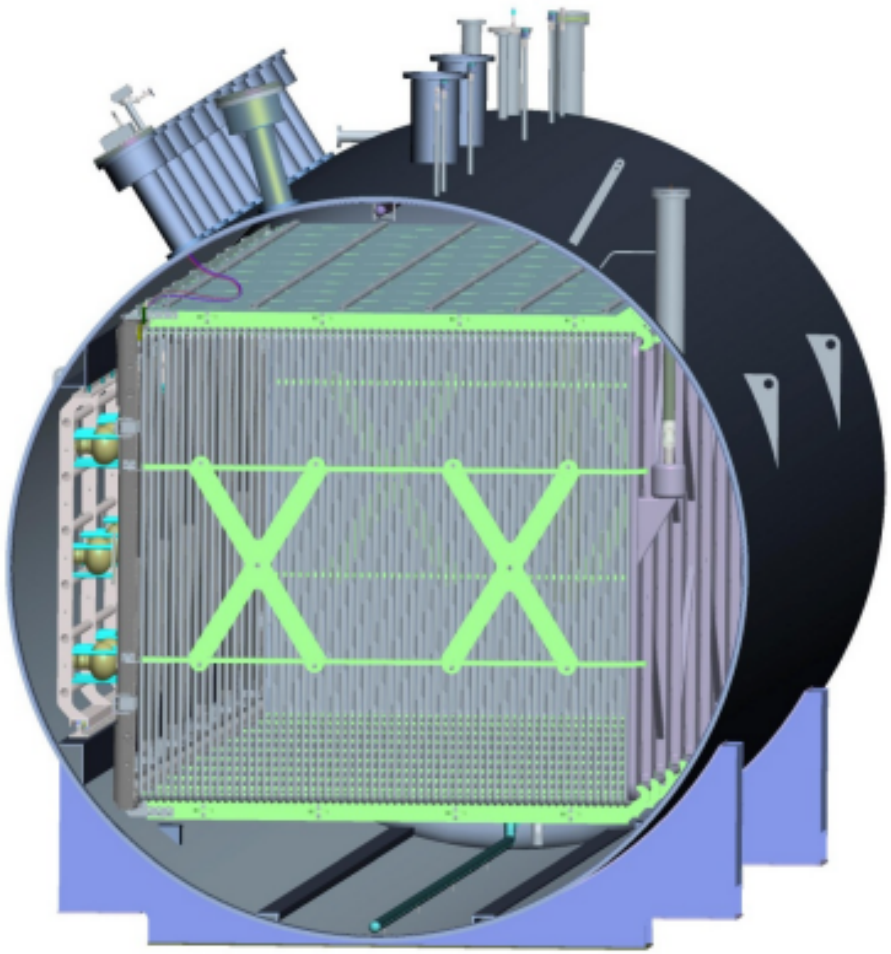






MicroBooNE instrumentation

- 8256 wire channels
- 32+4 PMTs



data on Wires are the projections in 3 views. Allows reconstructing back to 3D tracks and showers.

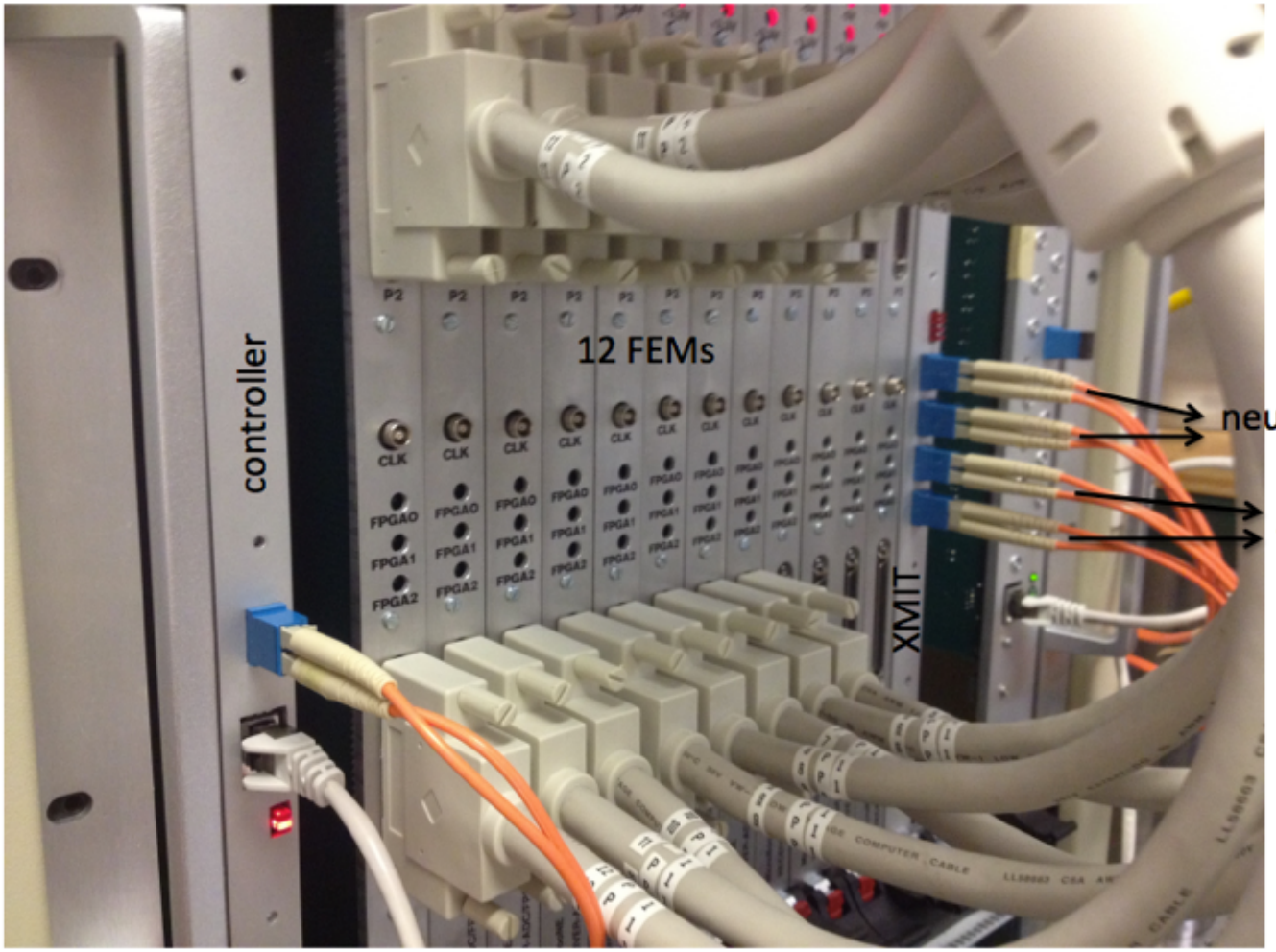
PMT data to trigger, to determine t0 for cosmics subtraction, and perhaps late/early light for pID.

Out at LArTF now

DAQ Servers

Test Stand

Nevis warm electronics



Fibers into SEB-computers

SEB == Sub Event Buffer <==> basically one crate.

One SEB computer services one crate.

There are two data streams.
(1) Triggered
(2) Supernova:
continuous stream

Both come through the "XMIT" card on duplex fibres.

Three NEVIS custom PCIe cards per SEB.

Overview of DAQ Project

- The MicroBooNE DAQ is responsible for reading out, assembling events, writing them to file.
- But also:
  - Monitoring and Control: EPICS database
  - Beam data concatenation
  - Online monitoring (A massively useful tool, about which regrettably almost nothing further in this talk)
  - Run Control
    - Calibration runs, Laser Runs, Regular runs ...
  - Swizzling (DAQ binary to ART ROOT format)
  - DAQ dbs and interaction with Offline db
  - File management (PUBS), ...

2 data streams

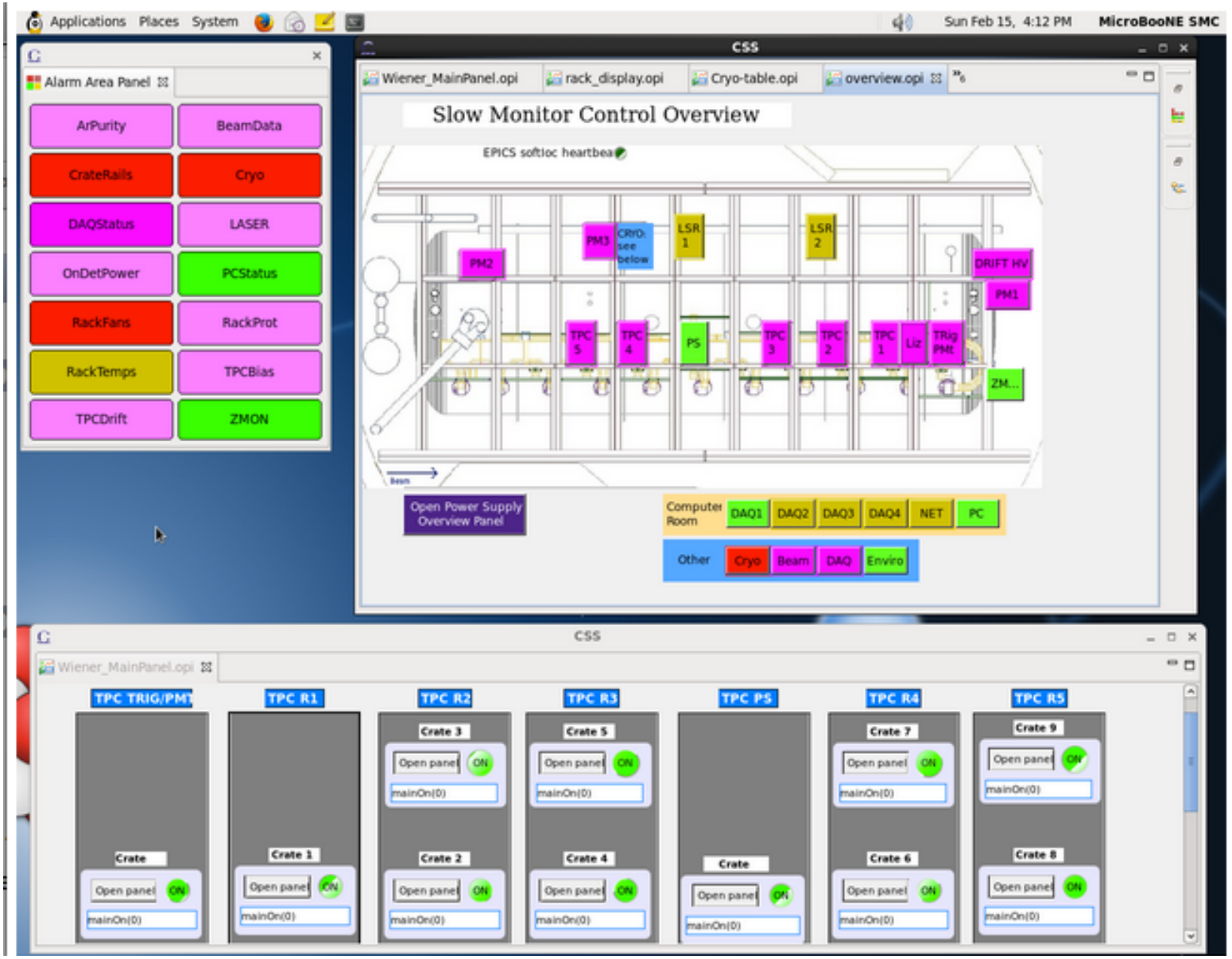
- Beam physics: triggered stream
  - confirm/refute/dispute miniBooNE BNB sub-400 MeV excess
  - x-sections in Argon
  - But also: NuMI events
  - Laser calibrations
  - “Strobe” events
- non-Beam physics: Supernova stream
  - Every single 1.6 msec frame, one after the other
  - Will fill seven 2 TB disks in two days on each crate.
  - GPS time of event will allow to dig through and recover 1-2 hrs around the candidate SN, as reported from SNEWS
  - non-trivial disk-read, network-heavy task.
    - data moving, reaping; other bkgd processes always running

This stream still being commissioned

Slow Monitoring and Control

- we run EPICS on our DAQ DB server
- The Wiener power supplies for the TPC warm electronics crates are under password-protected remote control/monitoring
- Similarly, with the ASICs power and drift and bias voltages
- Rack fans, temperatures are monitored by on-rack single board computers (Glomations) and report to EPICs. Cryo monitoring via IFIX to EPICs.
- Computer temperatures, fan speeds, RAID status also reported to EPICs. Ganglia metrics are harvested and also go to EPICs.
- Impedance between Detector/Building reports also
- All summarized for shifter and alarmed/archived.

SlowMonCon pictures



uB DAQ Test Stand

- We have a Test Stand at LArTF, as well, with two crates in computer room.
  - One server reads one crate's worth of FEMs for wires (~960 wires)
  - One server reads at least one FEM card in one crate of PMTs
  - A third server is the test stand event builder.
  - All servers are on a separate private network.
  - These crates do not read actual PMTs or wires in the tight space of the computer room
  - Though, if needed we have a test flange (chimney) we can connect to at DAB
- We test kernel/DAQ upgrades, if needed, here.
- These serve as our hot-swap machines too

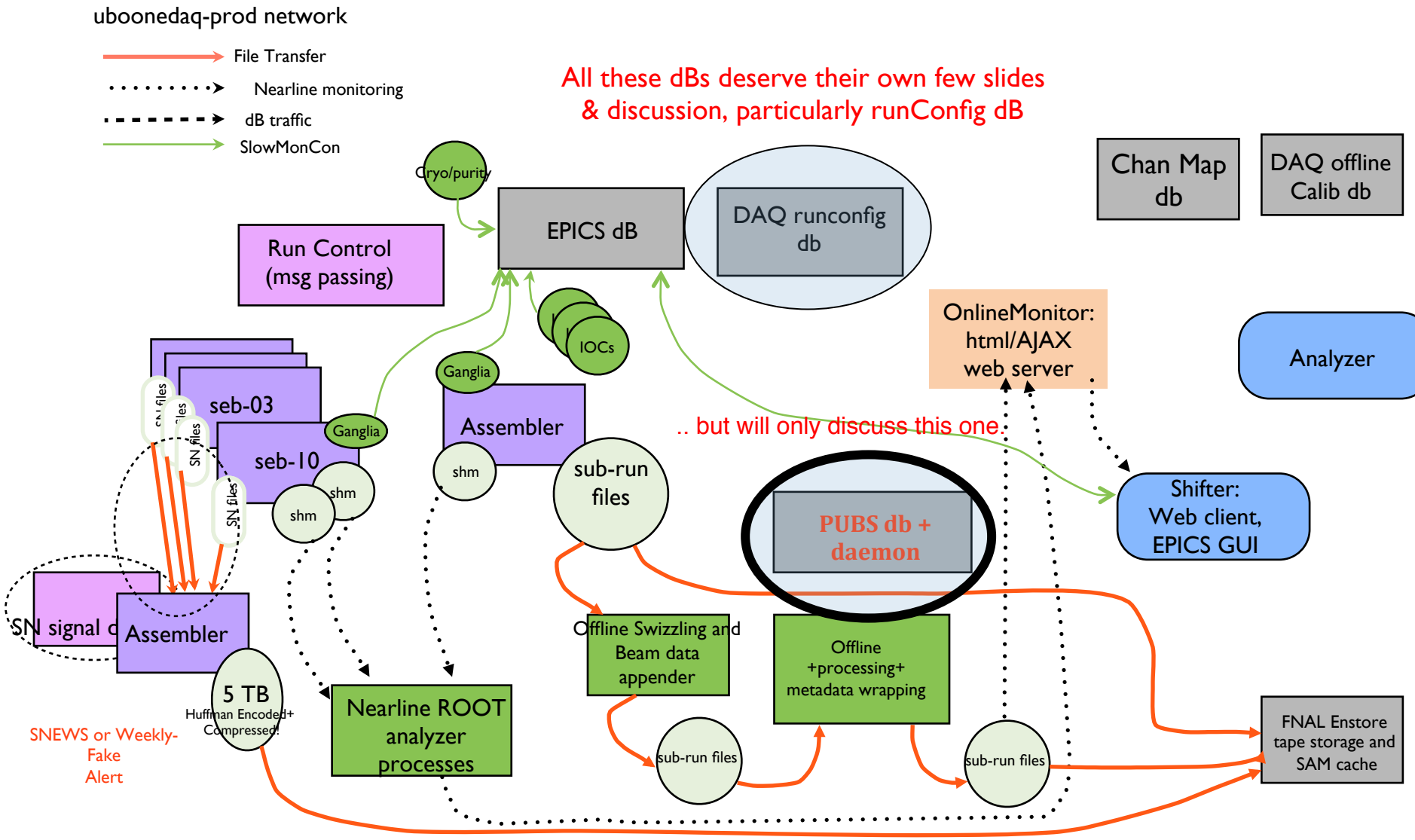
Server administration

- The SCD-SLAM team — Rennie Scott, Bonnie King, et al. — formally maintain all 15 of our production servers
  - It's been a very smooth and professional arrangement
  - We run SLF6.6
    - We have satisfactory root access, as well, as needed for modest scope changes
    - There will soon be ACL rules to only allow access in via 2 gateway nodes

DAQ software support

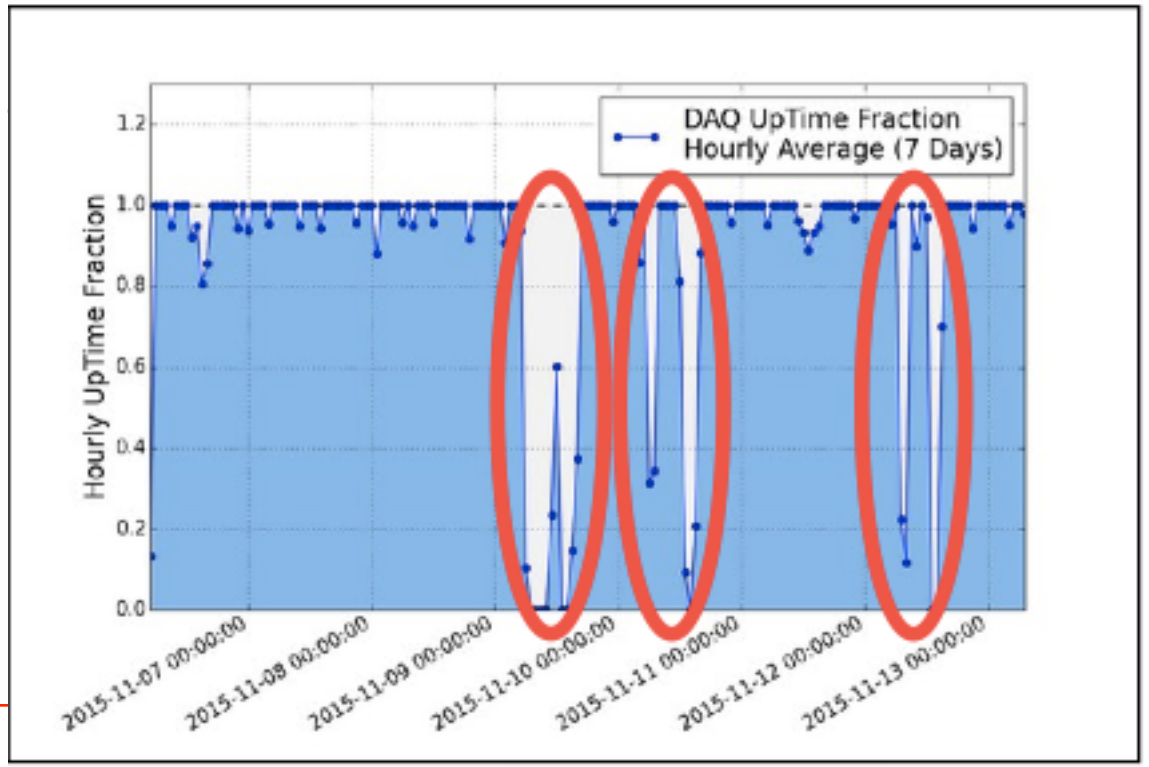
- SCD SSA team — Kurt Biery's group, mainly Gennadiy Lukhanin— has been instrumental in writing/supporting DAQ software
  - It's a very scalable, C++ solution, with the assembler and the 10 "sebApp" processors each inside a state machine, communicating via sockets. Shared memory segments for online
  - We have key **artdaq** components — fragments, event stores — but broke off our needs from that code repository.
  - Cmake/cetbuildtools build system; no mrb
  - We make our own UPS products: uboonedaq and uboonedaq-datatypes
    - The latter is the interface to LArSoft for the Swizzler

Online/Offline DAQ systems



PUBS: Python/Postgres-uB-Scripting

- Online version:
  - Moves our ~170MB/sec of raw binary data out to Enstore
    - Checksums along the way
    - Metadata creation
    - Registration to SAM
    - Ensures completion
    - Monitoring
    - Beam accesses Indicated: otherwise, 97+% DAQ uptime



DAQ UpTime (Hourly, Last 7 days)

PUBS Online 2

- However, 170MB/sec is actually onerous.
  - This alone is within a factor of 2 of FNAL's total tape-writing limit
- Swizzled (art-root format) is ~x1.7 bigger
- Minimal LArSoft reco is x8 bigger.
  - This is untenable.
  - There are other bottlenecks that begin to pop up, like dCache size, which must hold data as it is processed until next stage of Reco runs.
  - Cpu: Tens of minutes per event to process
- We will therefore run in open trigger only until mid-Jan
  - At which time we commission the PMT trigger
- We are processing ~2 percent of our data now, and will dedicate a processing campaign in January, which will take few wks to catch up.

PUBS Offline

- Swizzling => 3 streams
- Reconstruction Stage1 and Stage2
  - All on FermiGrid right now
- We still may go back to doing this Online when the firehose of data turns into a manageable stream.
  - This was the original plan
- MicroBooNE data management team doing a heroic job stemming the flood.
- This is just plain difficult, and I think there is no silver bullet if we insist that all this data is valuable.
- Will eventually be turned over to SCD:OPOS
- **… in general: dB coding is not to be marginalized**

Further uB DAQ difficulties

- The warm electronics / DAQ boundary.
  - Porting of solutions from warm electronics readout team to the uB DAQ code proper has been problematic at times
    - Configuring the crates (Recipes) brings transcription errors and guesswork
    - Monitoring solutions are different in a threaded environment than when performed serially.
    - FPGA <-> implementation is difficult to manage
  - The readout portion of the MicroBooNE DAQ could have been finished much earlier if many niggling little readout problems could have been reproduced and fixed. Instead, workarounds were developed during commissioning at the 11th hour that got us from erratic performance to reasonable run lengths (hours, not minutes) at high rates.

Readily-analyzable data

- Writing Binaries allows DAQ flexibility, but the hurdle to turning around analysis – of noise, x-talk, channel mapping – is significant.
- SBND planning to write out ROOT files from the beginning.
- Writing out art-ROOT file will still require a swizzling step somewhere.
  - LArSoft was in development during this period. Changes to raw::RawDigits raw::OpDetHits were/are always happening.

Run Control

- A proper Run Control GUI, ala NOvA's remains in the works. Still being worked on

- At some point we took most resources off this, deciding it was more heavyweight than called for and we instead have…

  - A big python script that works great.
    - A configuration is chosen via runConfig db
    - Failed runs are re-started automatically
    - Automatic eLog entries
    - A cool buffalo logo.

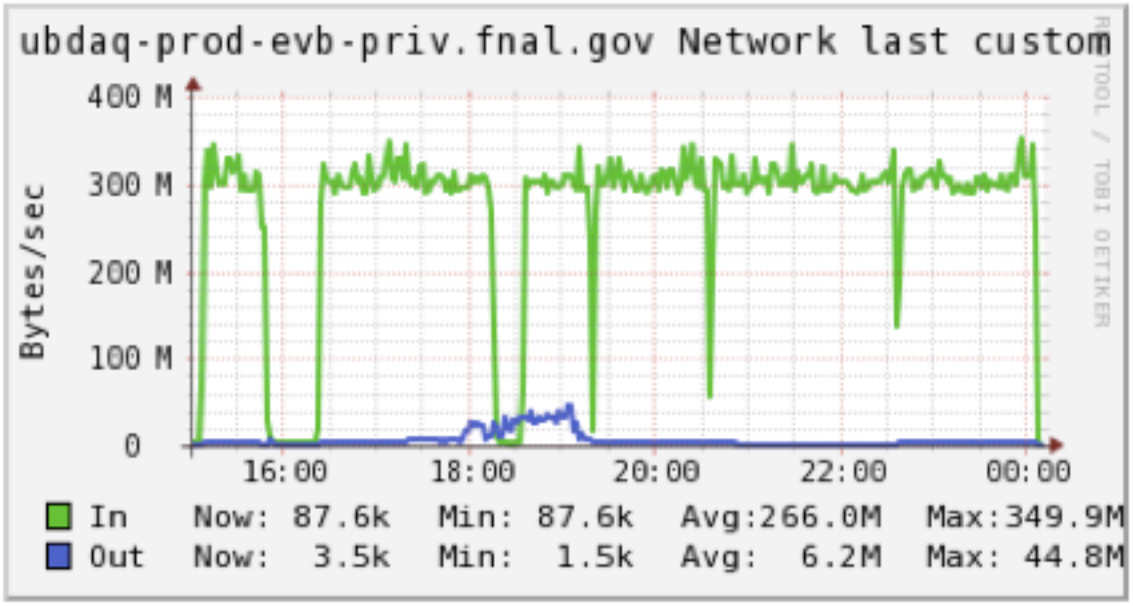
Conclusion

- This is (to most of us) a whole new technology, with MANY new challenges, among them massive data rates
- The DAQ work has not been trivial: lots of effort on: warm electronics readout, strong code design, and lots of C++1x and db coding and python scripting
- Only a very strong Readout and DAQ team and collaboration generally, and much useful guidance from SCD have allowed this to be successful
- We hope to preserve a lot of it for SBND
- Let's not under-estimate amount of good people and person-hours required to build future LArTPC DAQs

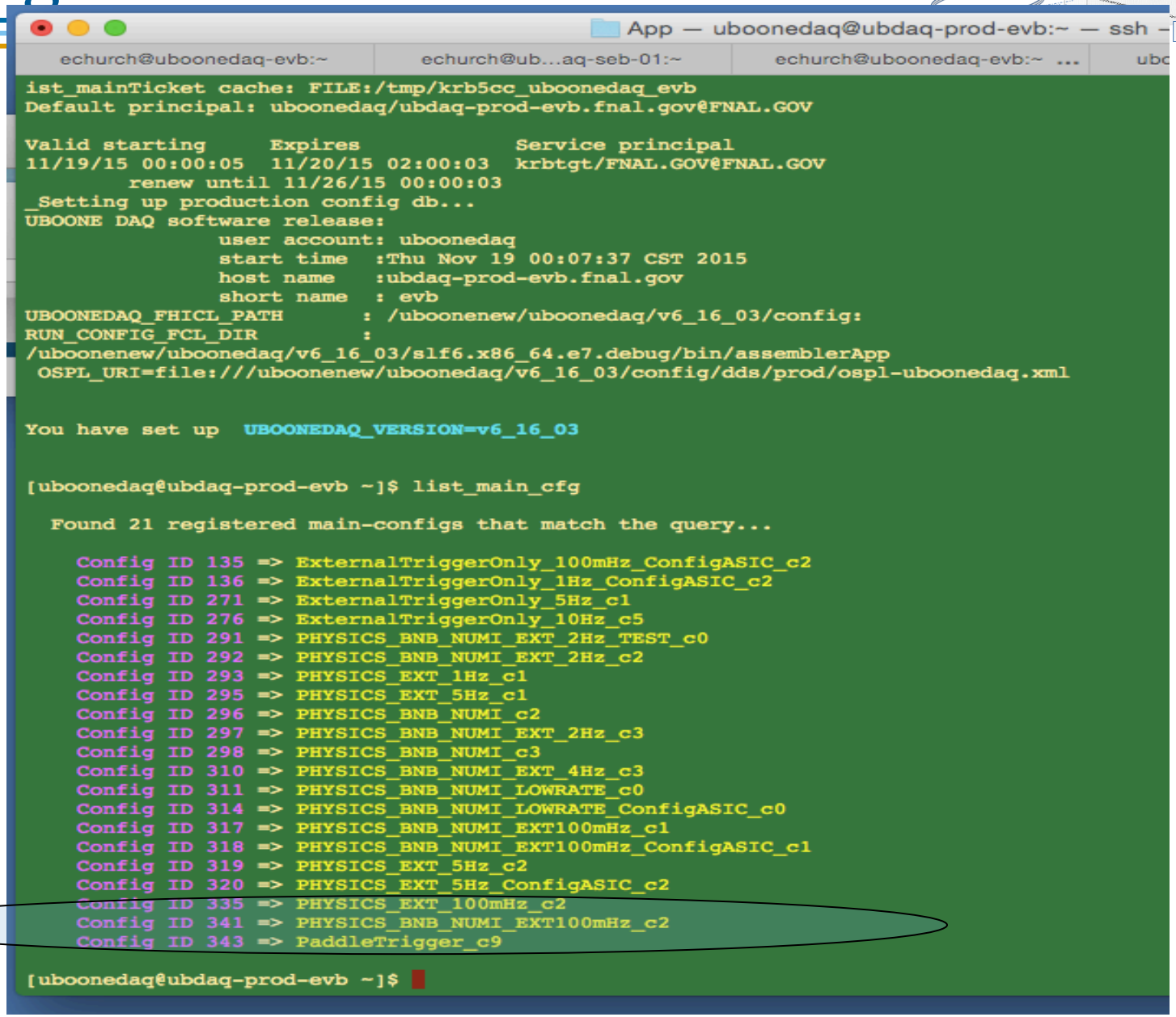
Backup Slides

Ganglia network rate

- We can in fact run steadily at 10 Hz, above which point we begin to be throttled by disk IO limits



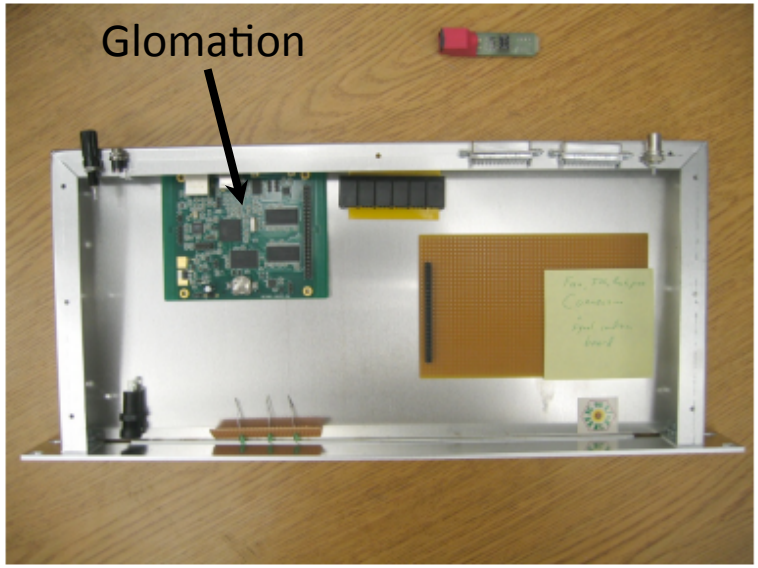
runConfig db



341 is our usual Config ID

Glomation Single Board Computer GESBC-9G20

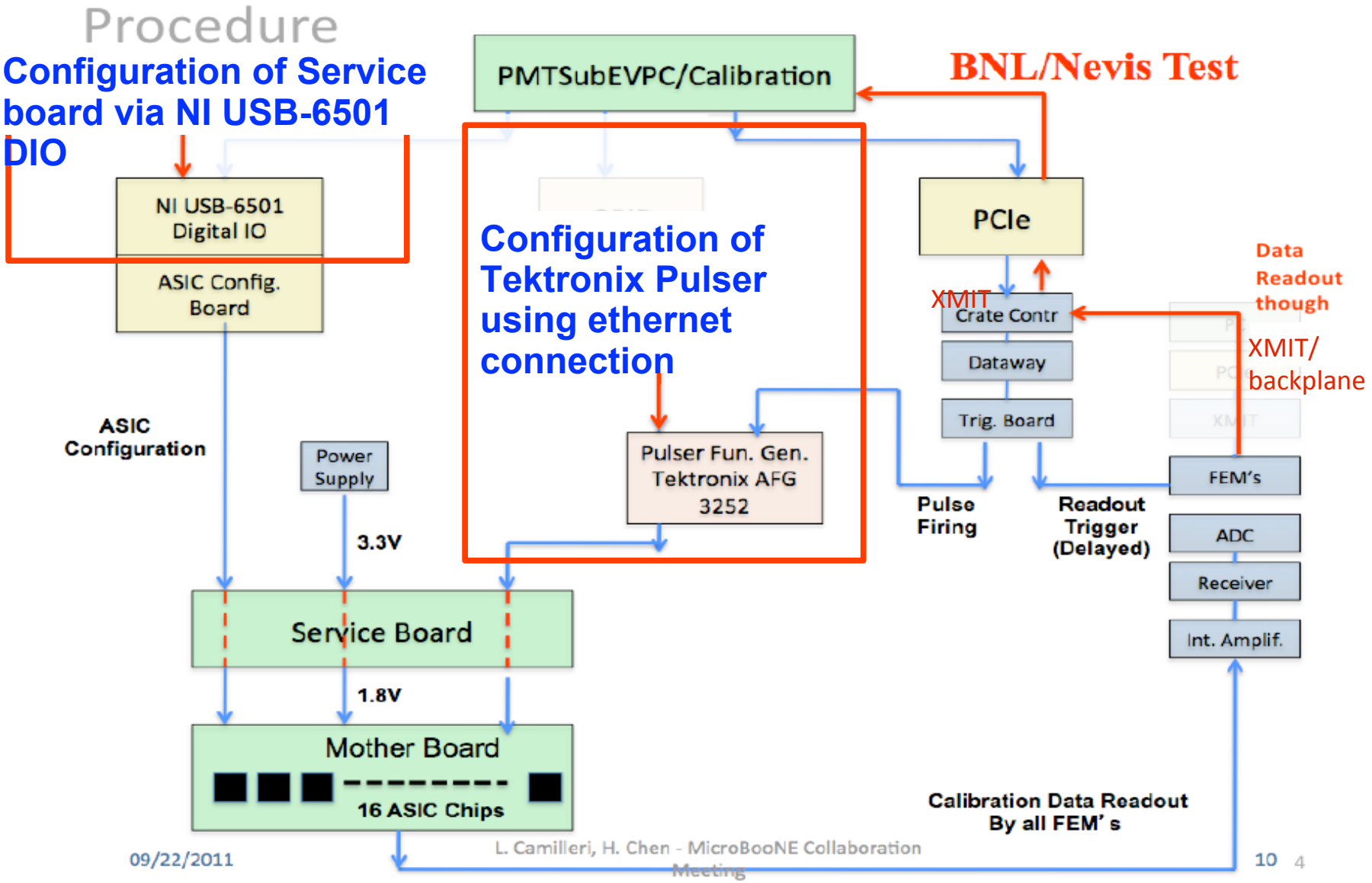
- SBC includes
  - Linux OS
  - Ethernet
  - RS232
  - USB
  - 40 digital I/O
  - 4 ADC
  - I2C and SPI bus



- Interfaces directly with
  - **Glassman Drift HV - RS232**
  - **Rack Temperature – I2C bus using Maxim DS1624**
  - **Rack Fanpack – digital I/O**
  - **PMT HV BiRa supplies**

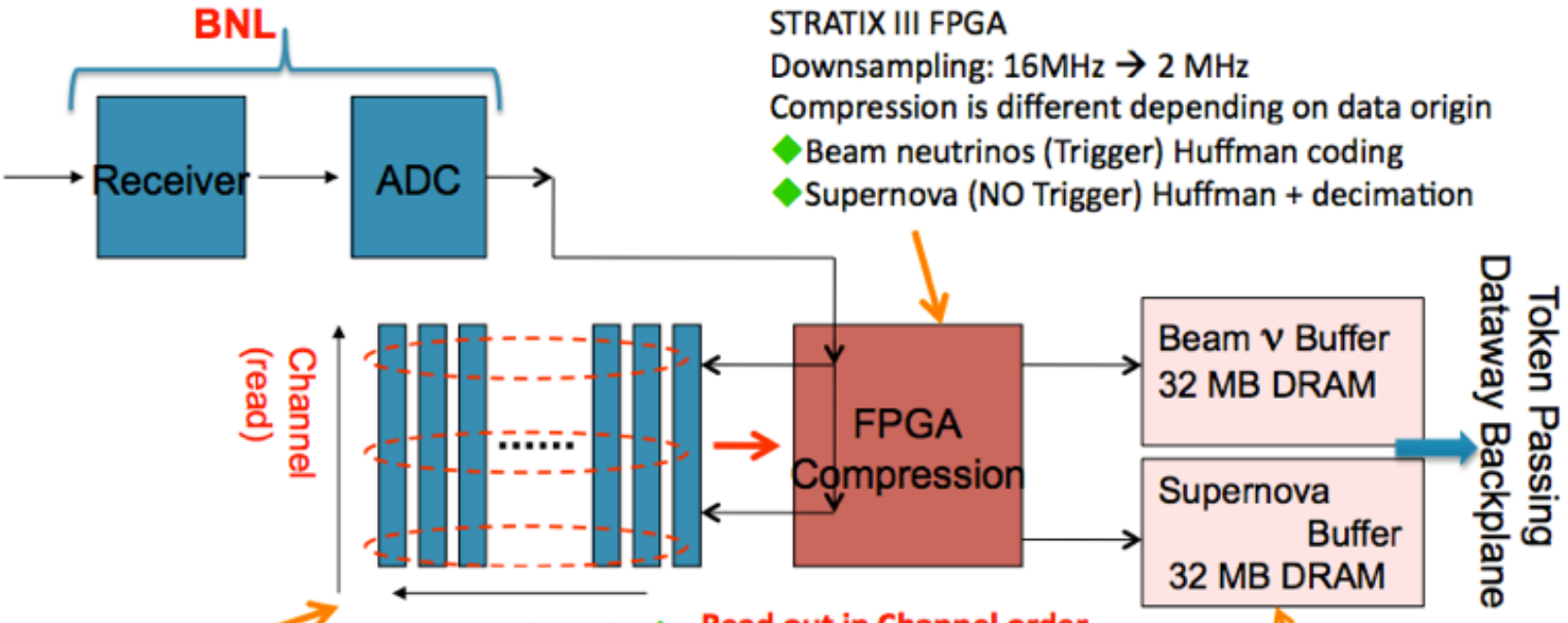
Glenn Horton-Smith, KSU

Calibration Hardware

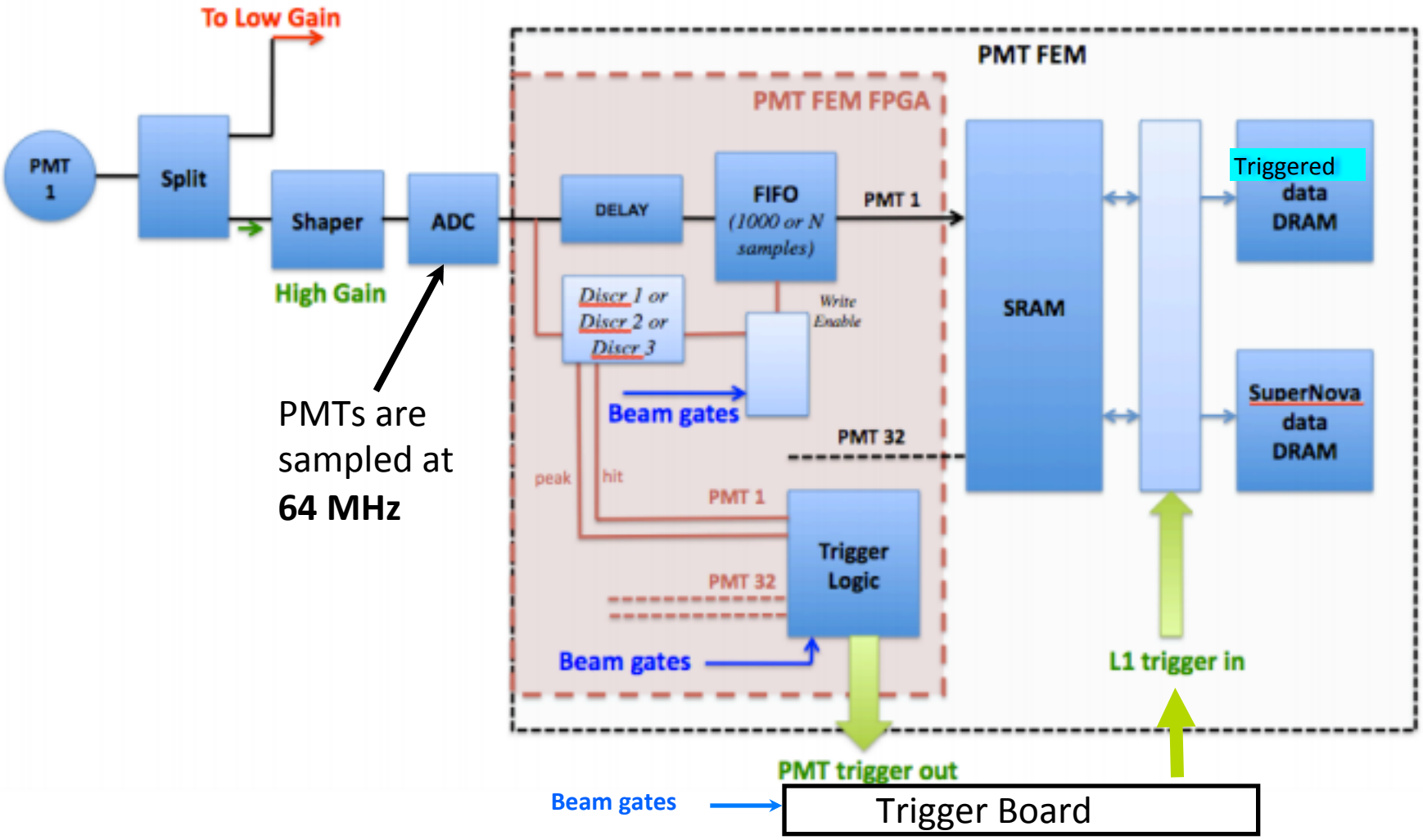


Overview: Digitizing Boards: Hardware and Tasks

Continuous Data rate per card: 64 wires x 2 MHz = 128 MHz 12 bit ADC words.



Trigger


To Low Gain
PMT 1
Split
Shaper
ADC
High Gain
PMTs are sampled at 64 MHz
PMT FEM
PMT FEM FPGA
DELAY
FIFO (1000 or N samples)
PMT 1
Discr 1 or Discr 2 or Discr 3
Write Enable
Beam gates
PMT 32
peak
hit
PMT 1
PMT 32
Trigger Logic
Beam gates
SRAM
Triggered data DRAM
SuperNova data DRAM
L1 trigger in
PMT trigger out
Beam gates
Trigger Board

SBND Upgrades

- An R&D/upgrade program is imagined – though nothing is formally proposed yet -- in which strictly COTs components, including this HTG PCIe card with its 120 Gbps each-way could handle one-third of the whole detector. So, 3 of ‘em in one server.