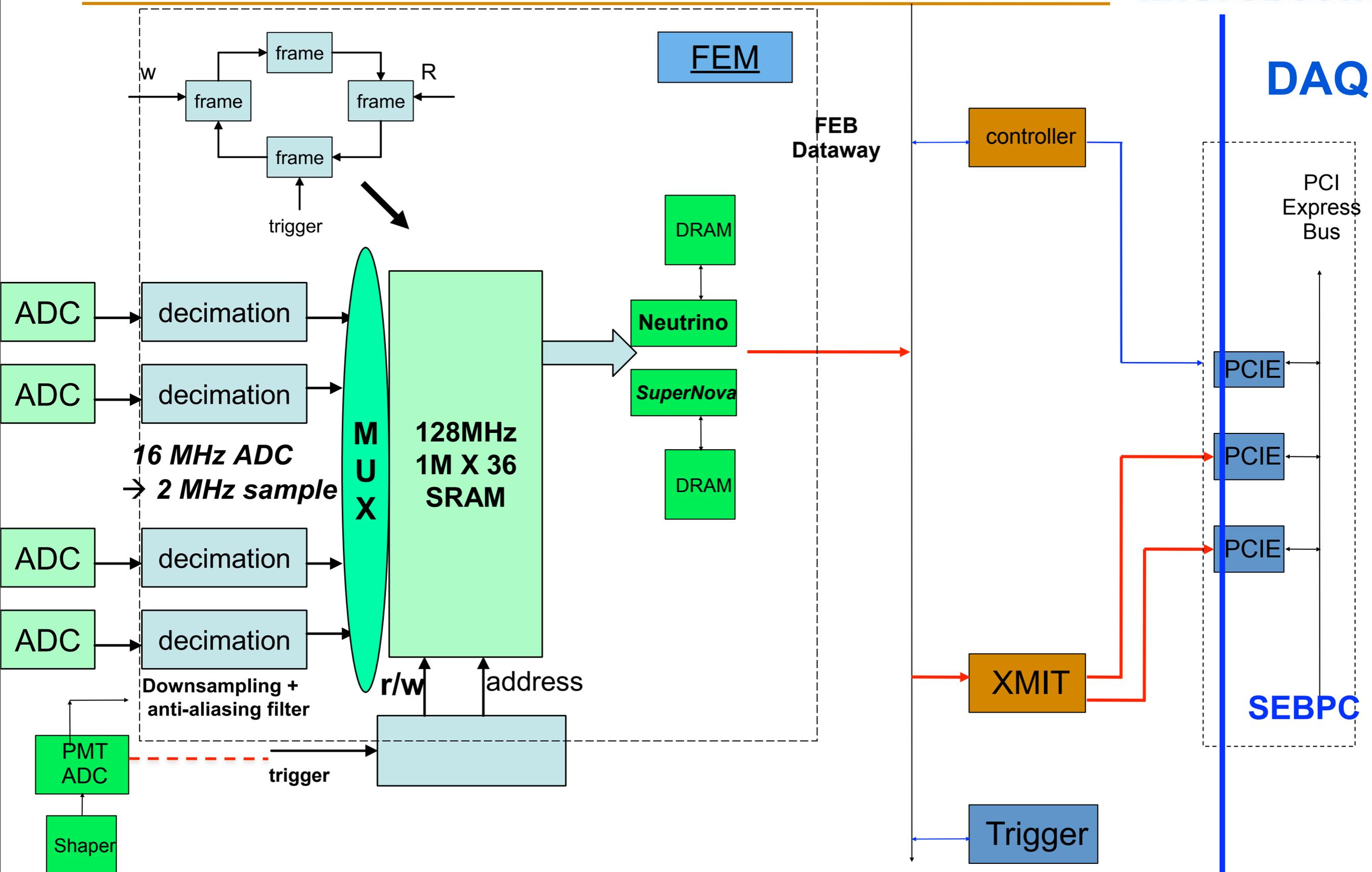


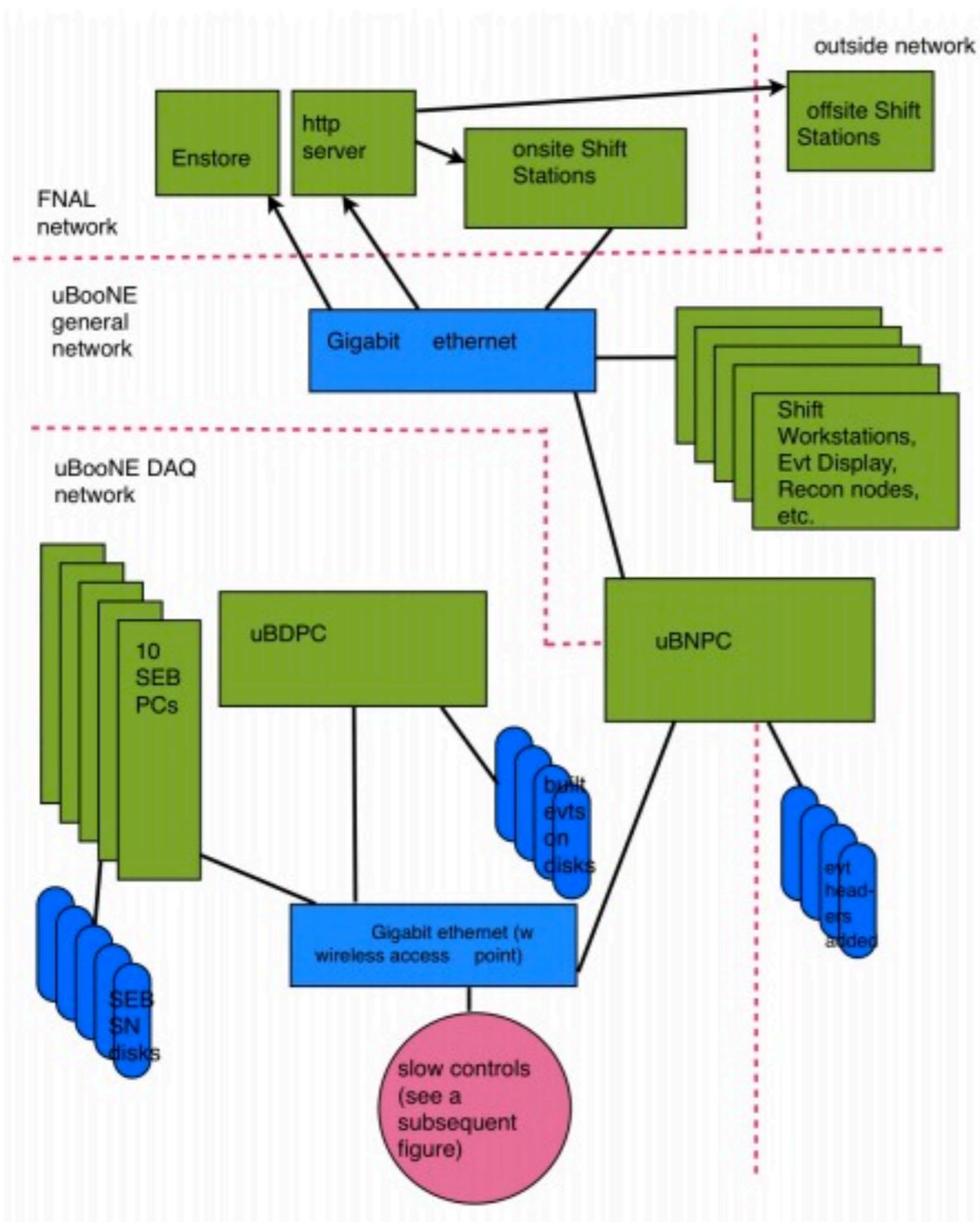
# uBooNE DAQ

Eric Church, uBooNE Director's Review Break-Out-  
FNAL, 12-July-2011.

# System Diagram for Triggered beam events

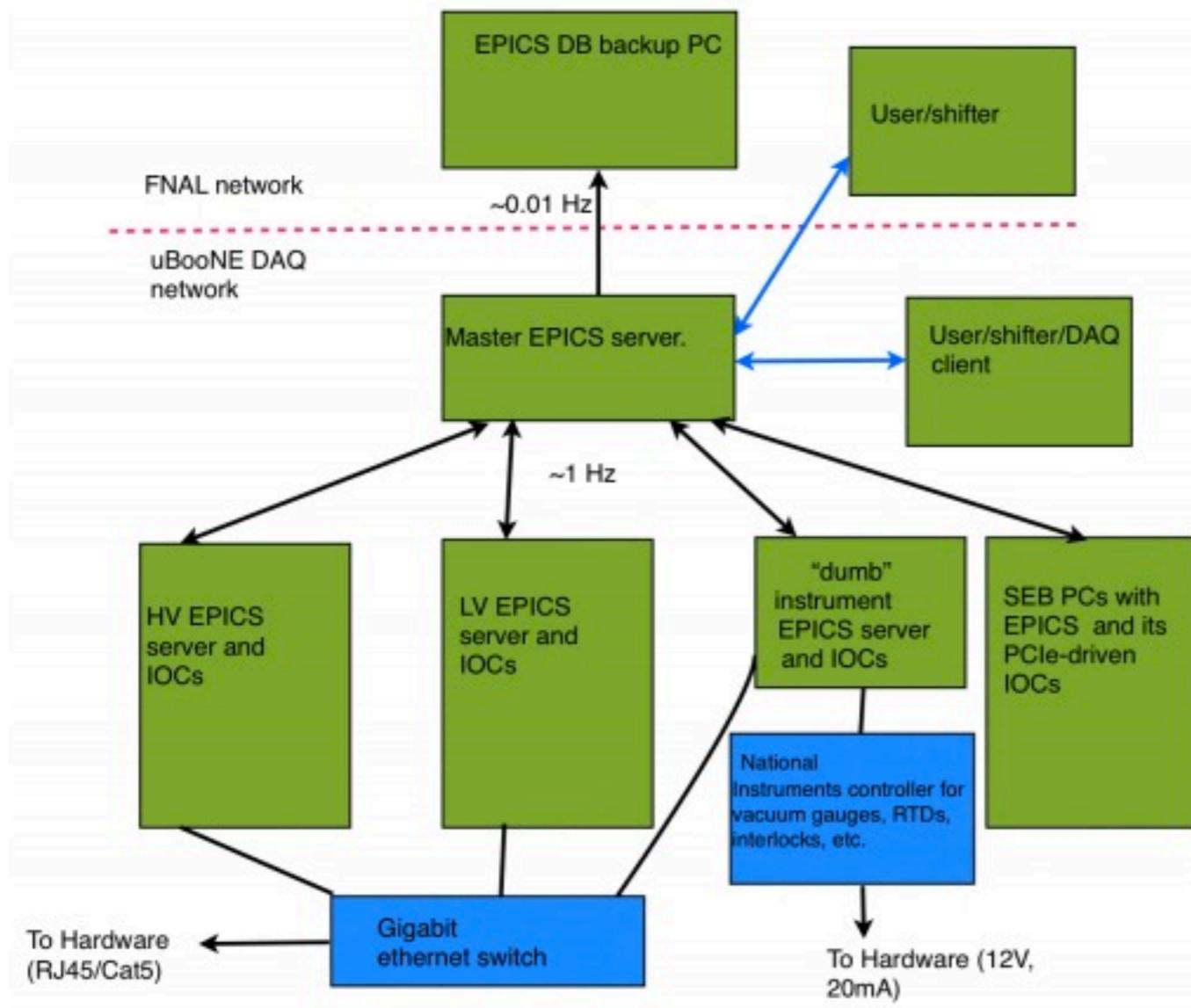


# DAQ System



- SEB PCs
  - Sub Event Buffer PC
  - Receive data from TPC/PMT readout boards through fiber optical links and PCIe cards
  - SEB Processes
    - Trigger loop process
    - Sender process to communicate with uBDPC through TCP/IP
- uBDPC
  - MicroBooNE DAQ PC
  - DAQ Processes
    - Assembler process to build the events
    - Streamer process to write events to local disk
- uBNPC
  - MicroBooNE Nearline PC
  - Nearline Processes
    - To manage raw data files and storage on ENSTORE system
    - To build supernovae event
    - To prepare files for web server

# Slow Control and Monitoring



- IOCs (Input/Output Controllers) live on PCs running EPICS
  - High voltage
  - Low voltage
  - Temperature/pressure/vacuum/interlocks etc.
  - SEB PCs run EPICS at low overhead for voltage/current readings in the crates
- Master EPICS Server
  - All EPICS PCs will report to a Master EPICS server
  - Responsible to push coarsely time averaged quantities at low rate to a DB on the FNAL network

SEBPC TPC code -- C everywhere unless specified. 9-10 instances of these.

from /etc/init.d on crate power-up

## Slow Mon of this crate

EPICS instance, **KSU**

```
main()
{
while()
{
// ~1 Hz local crate
// diagnostics
// over controller PCIe
readPCIeIOC();
write(localDB);
}
}
```

SN data reaper

```
main()
{
while()
{
checkDisk()
rm1HrOldFiles()
sleep(5m)
}
}
```

## Main data-taking Process

```
main()
{
Socket: create, bind, listen() // blocking
// pending assembler
accept()
read(FPGA code)
get(runInstructions) // cal or normal Run
if (calibrate)
{ // on devoted PCIe controller card.
ReadCalibTrigger(delay) // Nevis
}
nnu=0; nSN=0;
while(!endOfRun)
{
// Do this block twice: once for SN, once for triggers.
if (n*sizeof(event)>sizeof(memForManyBuffers))
{ // on devoted PCIe DMA cards.
malloc_pcie(memForManyBuffers);
dataPCIeTPCDriver(memFillFromDMA); // Nevis
headers,data=ReadEvent(eventSize*n)
}

if (lastNuThreadExited) pthread_create(Nu,headers,data)
if (lastSNThreadExited) pthread_create(superNovaData)
checkAndProceed(threadReturnConditions)
} // !endOfRun
}

void NuData()
{ //Write triggered buffer and adjacent, ~5MB/sec
buffers ,walkThroughMemory(); nnu++; writeDataToSocket
();
pthread_exit();
}
void superNovaData() // CD 9th Floor
{ //Write continuous buffers, ~ 50MB/sec
walkThroughMemory(); nSN++; writeDataToLocalDisk();
pthread_exit();
}
```

**connect to Assembler  
on uBDPC**

**pull data from FEMs**

**2-threads: SN and  
triggered data**

**send triggered data to  
assembler across socket**

**write SN data to local disk**

SEBPC **PMT/Trigger** code -- 1 of these.  
from /etc/init.d on crate power-up

**Slow Mon of this crate**

```
EPICS instance, KSU

main()
{
while()
{
// ~1 Hz local crate
// diagnostics
// over controller PCIe
readPCIeIOC();
write(localDB);
}
}
```

**Manage massive SN PMT data**

```
SN data reaper

main()
{
while()
{
checkDisk()
rm1HrOldFiles()
sleep(15m)
}
}
```

```
Main data-taking Process

main()
{
Socket: create, bind, listen() // blocking
// pending each SEBPC
accept()
read(FPGA code)
get(runInstructions) // cal or normal Run
if (calibrate)
{ // on devoted PCIe controller card.
Load(FPGA code) // Nevis
Load(calibTPCParams) // Nevis
FireAndSetCalibTrigger(delay) // Nevis
}

nSN=0
while(!endOfRun)
{
// Same memory allocation for both streams.
if (n*sizeof(event) > memForManyBuffers)
{ // on devoted PCIe DMA cards. These 3 cmds
// just once here, for PMTs.
malloc_pcie(memForManyBuffers);
dataPCIePMTDriver(memFillFromDMA); // Nevis
header,PMTdata=ReadPMTHeader(eventSize*n)
}

readExtrnlTriggers(header,clock)
pthread_create(Nu,headers,PMTdata)
pthread_create(SN,headers,PMTdata)

} // !endOfRun
}

// Same functionality in the two threads to ship triggered
// data over sockets to assembler and write SN data
// to local disk as in TPC SEBPC code. We must also
// read,write triggers here.
```

**connect to Assembler  
on uBDPC**

**Calibrate if desired**

**pull data from PMT FEMs**

**read trigger and clock  
ship data across socket to  
assembler**

uBDPC code -- C everywhere unless specified.

from /etc/init.d on uBDPC power-up, start daemons:  
daqLogd, shmMonitorFile, supervisor

assembler, dataStreamer started with each new run,  
eventually automated by supervisor.

```
supervisor script
#!/usr/bin/python
while True:
    checkAcnet,RWMDData
    checkDB
    checkDaemons
    checkDisks
    checkACNET_DAQ
    checkSundryProcesses
    once_A_Week:
        SuperNovaBuilder()
    if (pastCommissioning):
        start(assembler)
        start(dataStreamer)
```

**scripts to check DAQ health.**

```
assembler: main uBDPC data-taking Process
main()
{
    Socket: create, bind,
    startSHM()
    give(runInstructions) // cal or normal Run
    while(!endOfRun)
    {
        heartBeat(daqLog)
        getRunNumber(daqLog,dBHooks)
        connect(socketSEBs)
        collateData()
    } // !endOfRun
}

void collateData()
{
    connect(socketDS);
    select() // to listen to all SEB PCs on sockets and
            // receive data from them as shipped
    readData(socketSEBs);
    negotiateRaceConditions();
    buildEvent()
    shipData()
}

void shipData()
{
    write(socketDS) ← to dataStreamer
}
```

**shared memory to monitor run from other processes**

**build the event, ship it.**

```
dataStreamer
main()
{
    // socket to assembler
    attachToSHM();
    daqLogReport();
    bind,listen,accept()
    while()
    {
        checkDisk()
        shipOldFilesToEnstore()
        openSubRunFile(fd)
        data=read(socketDS);
        addGlbHeader(data)
        mergeACNET(data)
        writeToLocalFiles(data)
    }
}
```

**0th order DAQ task.**

```
SuperNovaBuilder
main()
{
    attachToSHM();
    daqLogReport();
    readFromSEBDisksBuildWrite()
}
```

**once/few-day SN event build. Will eventually exercise this with a (pretend) SNEWS email.**

# Runs on EC's Powerbook

The image shows a Mac OS X desktop with several terminal windows open. The top bar indicates the date is Wednesday, February 23, 2011, at 12:37 PM, and the user is Eric Church. The desktop has a dock at the bottom with various application icons.

Key terminal windows and their content:

- Terminal - fake\_seb - 57x23:** Shows a loop of commands: `te$[1 + ${i}].out -p ${31415 + ${i}} -t 2000`. It lists several "Killed" entries for PID 22345 through 22354.
- Terminal - fake\_assemblerMu - 80x24:** Shows a loop of commands: `fake_assemblerMulti: finished 2100000 events. Wrote 20999996.` through `fake_assemblerMulti: finished 2320000 events. Wrote 23199996.`
- Terminal - shmMonitor - 80x24:** Shows output from the `shmMonitor` tool, including statistics like "Run number = 0 (DATA MODE)", "Event number = 733501, Latent = 0 (=0.0%)", and "Total Rate = 104785 Hz, Latent Ave = nan Hz". It also shows a list of detector parameters like "Beam", "Strobe", "Laser", etc.
- Terminal - daqLogd:** Shows the execution of `daql` and `daqlLogd` commands, including the output of `ps aux | grep daqL` which shows the `daqlLogd` process running.
- Terminal - top - 85x29:** Shows the output of the `top` command, listing running processes. The `fake_assembl` process is highlighted with a red circle, showing it is using 81.8% CPU.

Overlaid text annotations:

- fake\_SEBs** (circled in red)
- fake\_assemblerMulti** (circled in red)
- writes @25 MB/sec** (circled in red)
- shmMonitor** (circled in red)
- top** (circled in red)
- daqlLogd** (circled in red)

# Next work, in order

- SEBPC server is in Test Stand in LCCI08. Some PCIe code has been exercised there: data rates on bus verified. Some EPICs code too: monitor/report Event Number from Shared memory.
- Add new KOI uBDPC & a network switch to that Test Stand in next couple weeks. Network these up. Repeat slide 8, but on 2 real machines. (Something like this has already been done at KSU. All rates are encouraging.)
- Add a second SEBPC to network. Assembler can then be tested on uBDPC. Need to gain some experience with reading/passing-on compressed uBooNE fake data. This is in progress.
- PCIe communication. Understand Jungo driver to interrupt, write/read registers, do DMA. Put fake data onto optical bus, and pull it out into SEB PC memory. (This is well along. See Chi's slide 20, docdb1570)

# Assembler and clients on 2 Hosts running SL6

Same or higher performance with (mock) SEBs on one host and assembler on the other

The screenshot shows a terminal window with the following output:

```
fake_assemblerMulti: Finished 500000 events, Wrote 57533799.  
fake_assemblerMulti: Finished 5100000 events, Wrote 57633799.  
fake_assemblerMulti: Finished 5200000 events, Wrote 57733799.  
fake_assemblerMulti: Finished 5300000 events, Wrote 57833799.  
fake_assemblerMulti: Finished 5400000 events, Wrote 57933799.  
fake_assemblerMulti: Finished 5500000 events, Wrote 58033799.  
fake_assemblerMulti: Finished 5600000 events, Wrote 58133799.  
fake_assemblerMulti: Finished 5700000 events, Wrote 58233799.  
fake_assemblerMulti: Finished 5800000 events, Wrote 58333799.  
fake_assemblerMulti: Finished 5900000 events, Wrote 58433799.  
fake_assemblerMulti: Finished 6000000 events, Wrote 58533799.  
fake_assemblerMulti: Finished 6100000 events, Wrote 58633799.  
fake_assemblerMulti: Finished 6200000 events, Wrote 58733799.  
fake_assemblerMulti: Finished 6300000 events, Wrote 58833799.  
fake_assemblerMulti: Finished 6400000 events, Wrote 58933799.  
fake_assemblerMulti: Finished 6500000 events, Wrote 59033799.  
fake_assemblerMulti: Finished 6600000 events, Wrote 59133799.  
fake_assemblerMulti: Finished 6700000 events, Wrote 59233799.  
fake_assemblerMulti: Finished 6800000 events, Wrote 59333799.  
fake_assemblerMulti: Finished 6900000 events, Wrote 59433799.  
fake_assemblerMulti: Finished 7000000 events, Wrote 59533799.  
fake_assemblerMulti: Finished 7100000 events, Wrote 59633799.  
fake_assemblerMulti: Finished 7200000 events, Wrote 59733799.  
fake_assemblerMulti: Finished 7300000 events, Wrote 59833799.  
fake_assemblerMulti: Finished 7400000 events, Wrote 59933799.  
fake_assemblerMulti: Finished 7500000 events, Wrote 60033799.  
fake_assemblerMulti: Finished 7600000 events, Wrote 60133799.  
fake_assemblerMulti: Finished 7700000 events, Wrote 60233799.  
fake_assemblerMulti: Finished 7800000 events, Wrote 60333799.  
fake_assemblerMulti: Finished 7900000 events, Wrote 60433799.  
fake_assemblerMulti: Finished 8000000 events, Wrote 60533799.  
fake_assemblerMulti: Finished 8100000 events, Wrote 60633799.  
fake_assemblerMulti: Finished 8200000 events, Wrote 60733799.  
fake_assemblerMulti: Finished 8300000 events, Wrote 60833799.  
fake_assemblerMulti: Finished 8400000 events, Wrote 60933799.  
fake_assemblerMulti: Finished 8500000 events, Wrote 61033799.
```

The 'shmMonitor' window shows the following output:

```
Run number = 0 (0 Hz noise)  
Event number = 60833799, Latent = 0 (+0.01)  
Total Rate = 0 Hz, Latent Ave = nan Hz  
Instant Rate = 448719.0 Hz, Latent Inst. = 0.00 Hz  
Started Wed 21-Dec-2009 18:00:00 EST, 0 Hz  
Current Wed 21-Dec-2009 18:00:00 EST, 0 Hz  
Time Elapsed = 0.00 hr  
Rate Info: instant(average), total count(latent)  
Beam: 0.00(nan) Hz, 0(0) Gausse: 28; 0.00(nan) Hz, 0(0) Gausse;  
0.00(nan) Hz, 0(0) total: 0.00(nan) Hz, 0(0)  
Strobe: 0.00(nan) Hz, 0(0)  
Laser: 0.00(nan) Hz, 0(0)  
top - 14:56:16 up 235 days, 15 min, 7 users, load average: 0.10, 0.45, 0.15  
Tasks: 295 total, 4 running, 289 sleeping, 0 stopped, 0 zombie  
Mem0: 0.00(nan) Hz, 0(0)  
Cpu(s): 0.22us, 7.62ms, 0.03s, 31.82id, 0.00us, 0.00ms, 0.00s, 0.02st  
Mem: 24545012k total, 2300134k used, 944473k free, 151144k buffers  
Swap: 25730680k total, 77278k used, 25653402k free, 23020176k cached  
PID USER PR NI VIRT RES SHR S CPU% MEM% TIME+ COMMAND  
23158 gahs 15 0 13660 1128 864 S 14.0 0.0 0:23.56 fake_seb  
23159 gahs 15 0 13660 1128 864 S 13.3 0.0 0:17.98 fake_seb  
23162 gahs 15 0 13664 1132 864 S 13.3 0.0 0:23.30 fake_seb  
23156 gahs 15 0 13664 1128 864 R 13.0 0.0 0:23.17 fake_seb  
23154 gahs 15 0 13660 1124 864 S 12.5 0.0 0:23.25 fake_seb  
23157 gahs 15 0 13664 1128 864 R 12.5 0.0 0:23.39 fake_seb  
23163 gahs 15 0 13660 1120 864 R 12.5 0.0 0:23.14 fake_seb  
23163 gahs 15 0 13664 1120 864 S 12.5 0.0 0:23.22 fake_seb  
23155 gahs 15 0 13664 1120 864 S 12.5 0.0 0:23.32 fake_seb  
23160 gahs 15 0 13656 1124 864 S 12.0 0.0 0:23.05 fake_seb  
23216 gahs 15 0 22100 2384 1888 R 0.5 0.0 0:00.16 top  
1 root 15 0 13356 200 168 S 0.0 0.0 0:02.94 smit  
2 root RT -5 0 0 0 0 5 0.0 0.0 0:00.47 nsgratcon/0  
3 root 34 19 0 0 0 0 5 0.0 0.0 0:00.00 kseFtrngd/0  
4 root RT -5 0 0 0 0 5 0.0 0.0 0:00.00 unchdbg/0  
5 root RT -5 0 0 0 0 5 0.0 0.0 0:01.25 nsgratcon/1  
6 root 34 19 0 0 0 0 5 0.0 0.0 0:00.00 kseFtrngd/1
```

First uB SEB server (uboonedaq01.fnal.gov)  
Located on a shelf in LCC at FNAL

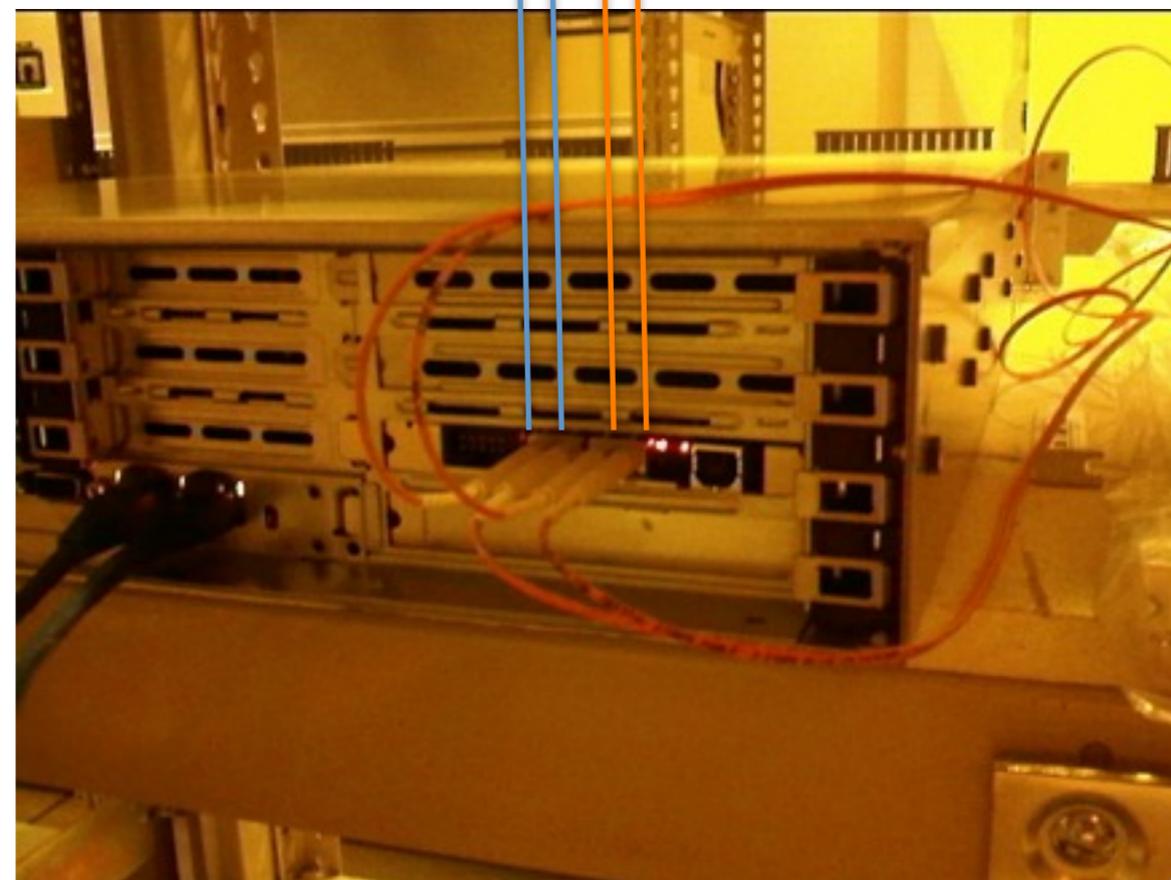


Presently 6TB of space  
(expandable up to 12TB)

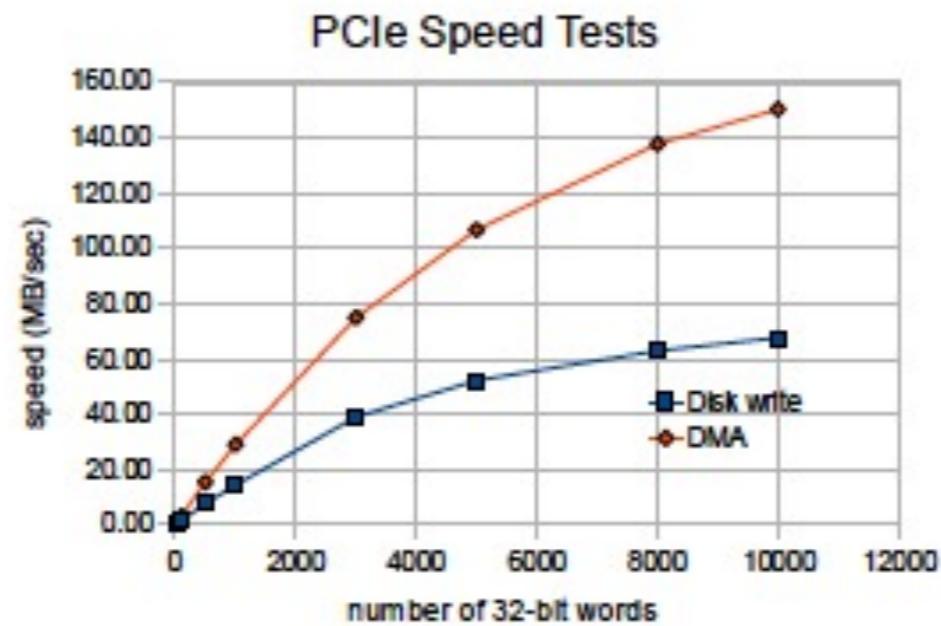
Koi 2U Dual AMD Opteron 6128 Storage Server

Present PCIe hardware settings (for diagnostics tests):

Out is patched to In for both lanes for testing.



# PCIe data rates



Rates look good w.o. even yet making the effort to optimize reads/writes.

# uBooNE DAQ project

- We have a uBooNE DAQ project: <https://cdcv.sfnal.gov/redmine/projects/uBooNEDAQ>
- git code repository.
- plans, instructions.
- We meet fortnightly for now.

# Work, cont'd

- Nearline coding can start when we have some sensible MC output fc files. Nearline: 2nd level triggers, physics stream splitting, EVD, Recon (LArSoft) processes (?), database, file management, ...
- EPICs coding, databases. PAB HV Test Stand -- an actual useful milestone toward EPICS-ing things up, is coming along.
- Overall System State Management, scripts, unit tests, ...



# Coders

- Nevis+Yale: 2 50% FTEs each, one 9th floor 50% FTE
- EPICs: KSU 50% FTE; one 9th floor 50% FTE
- State, SuperNova read-out, etc.: one 9th floor 50% FTE

# Test Stand Schedules

- PAB HV for PMTs: EPICs exercise which will be directly translatable to Detector Hall.
- LCC Test Stand at FNAL: Testing/developing code, shipping all forms of fake data. Now through Detector Hall occupancy in 2013.
- Feb/Mar, 2012: Proto 2 Test Stand at Nevis will come to FNAL. Electronics in a crate: actual FEM cards and XMIT and Controller card, so we can move FPGA-generated data around. Then “real” ADC data. {Test per Chi’s slide 17, docdb-1570.}