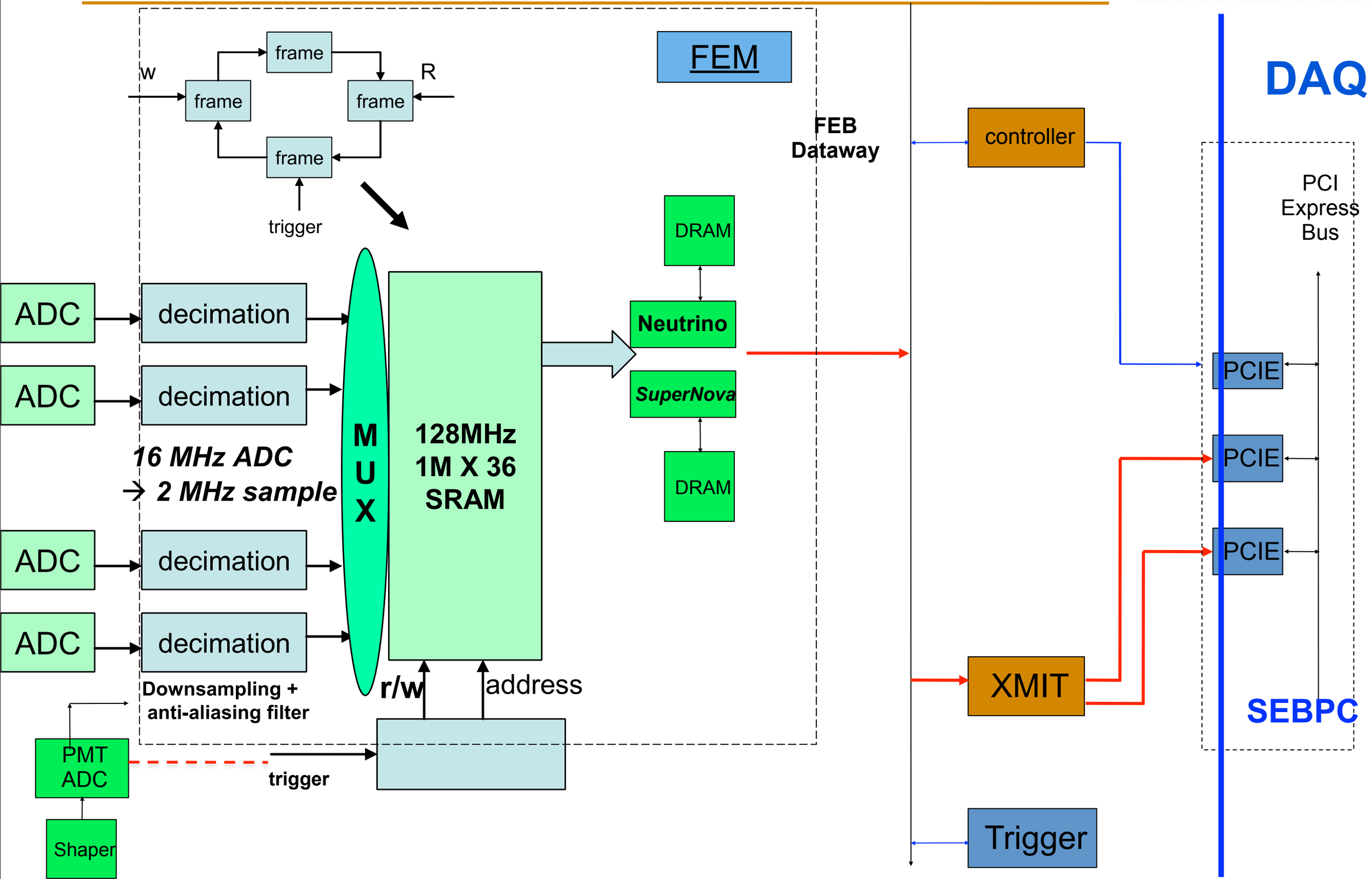
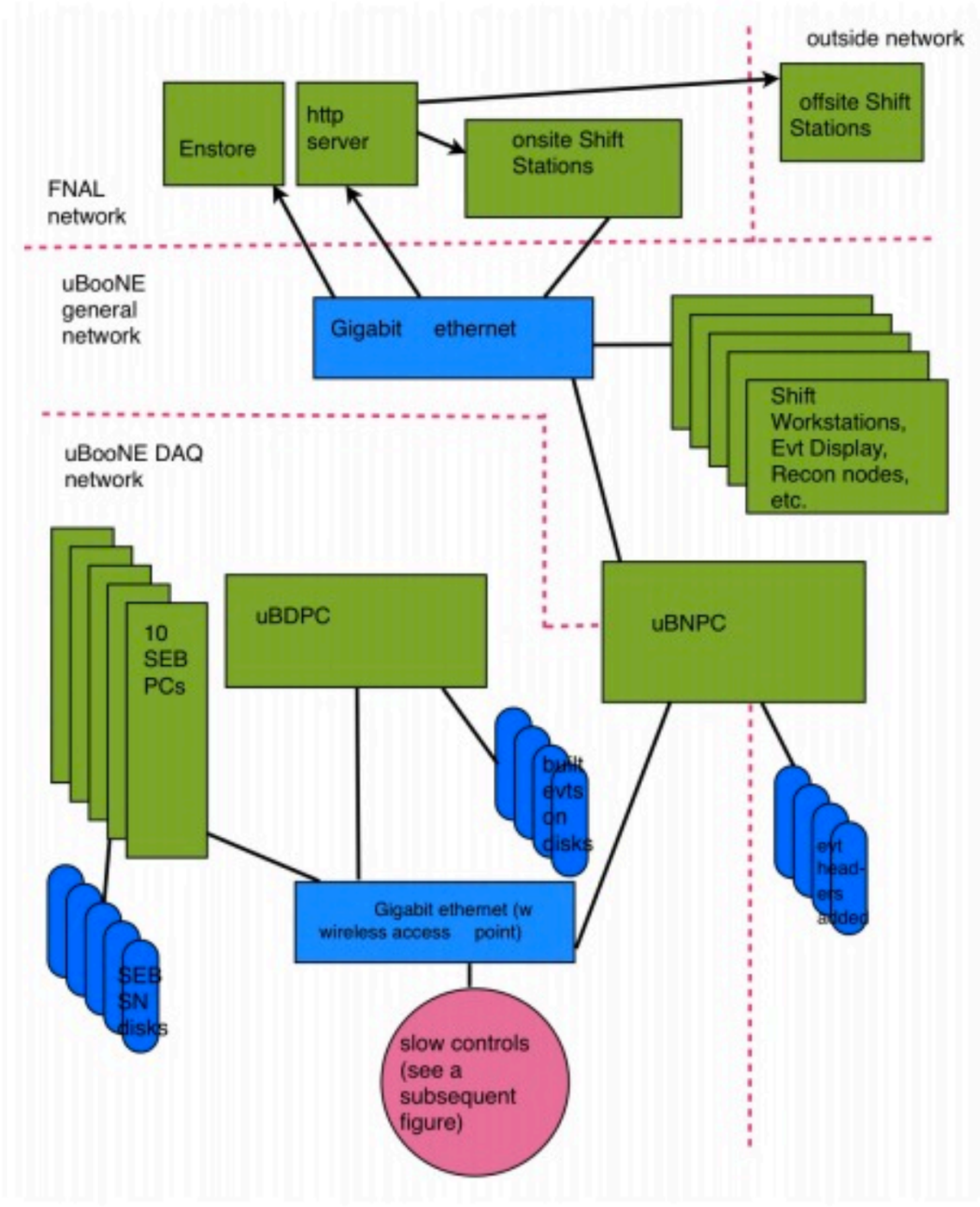
uBooNE DAQ

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

System Diagram for Triggered beam events

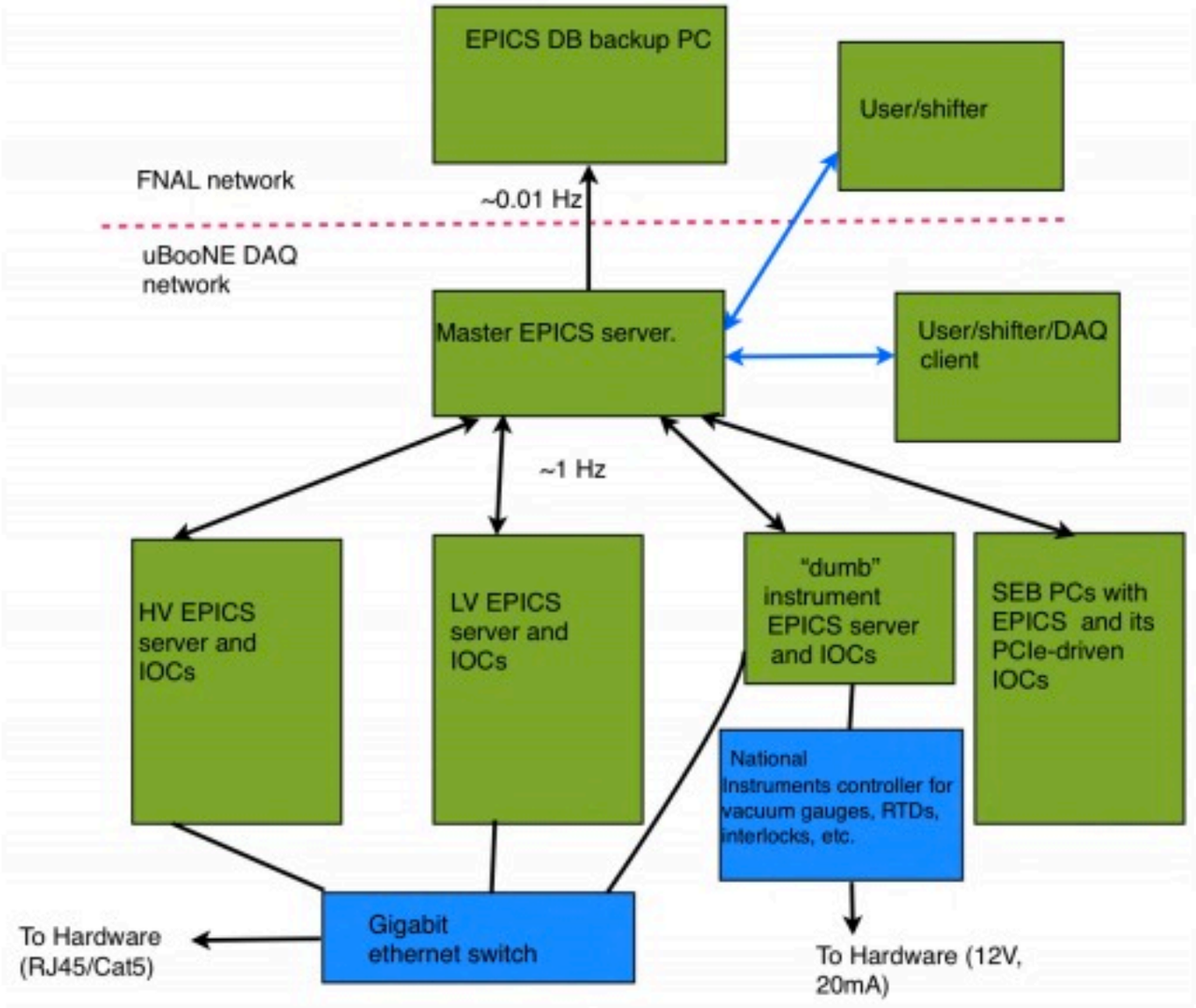


DAQ System



- 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.

shared memory to monitor run from other processes

build the event, ship it.

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
    }
```

supervisor script

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

scripts to check DAQ health.

dataStreamer

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

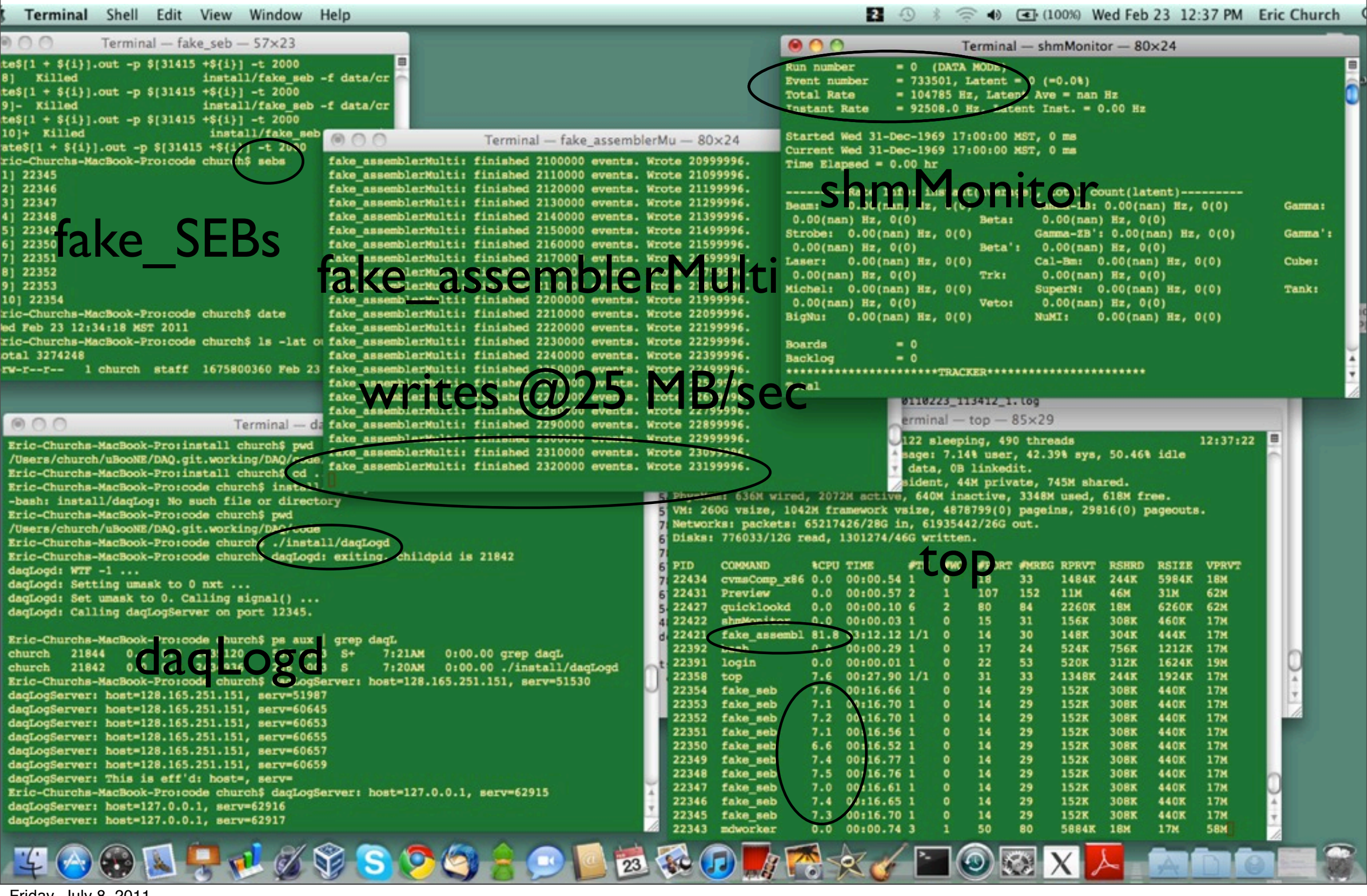
0th order DAQ task.

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



Next work, in order

- SEBPC server is in Test Stand in LCC108. Some PCIe code has been exercised there: data rates on bus verified. Some EPICs code too: monitor/report Event Number from Shared memory.

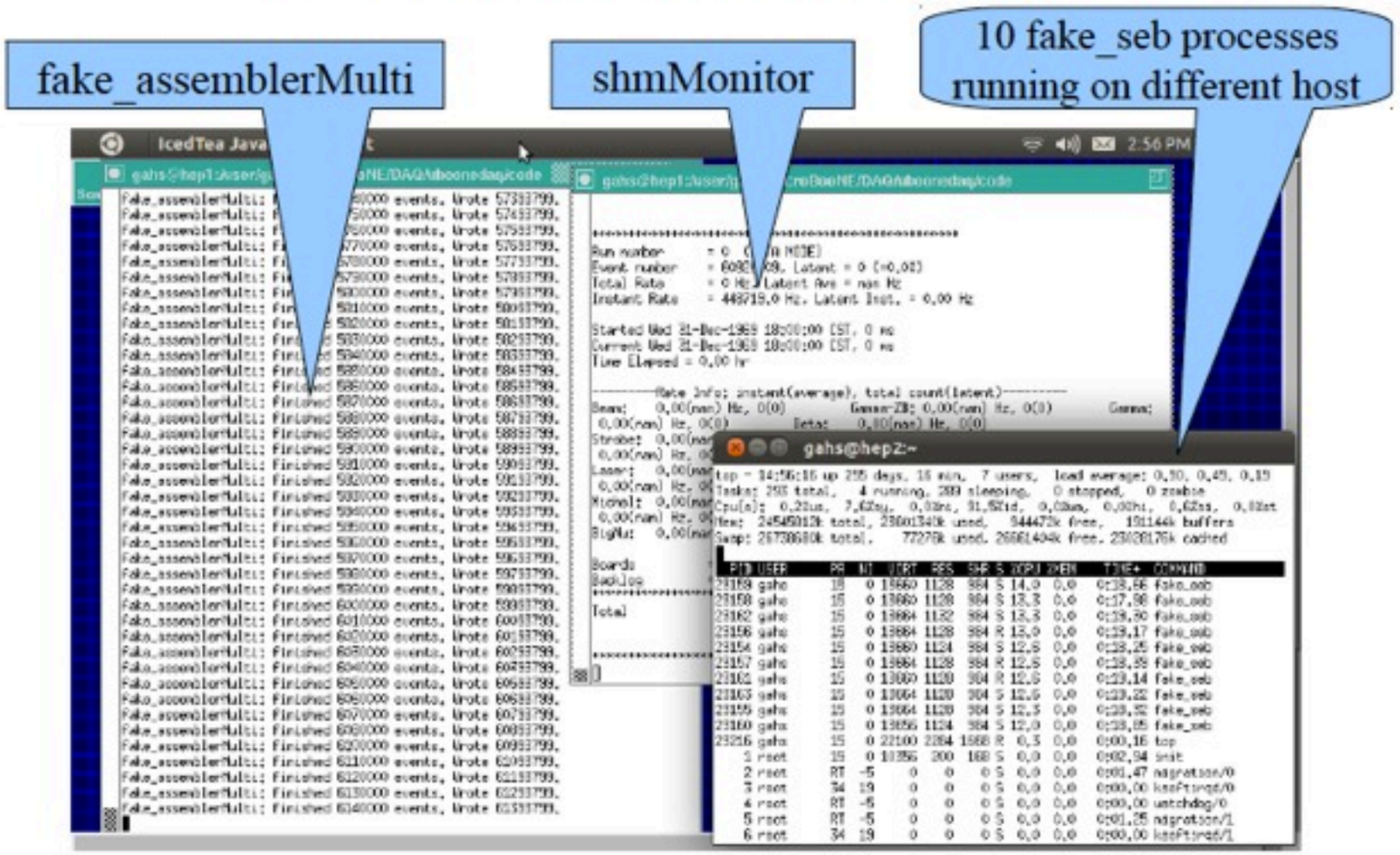
- Add new KOI uBDPC & a network switch to that Test Stand in next couple weeks. Network these up. Repeat slide 8, but on 2 real machines. (Something like this has already been done at KSU. All rates are encouraging.)

- 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



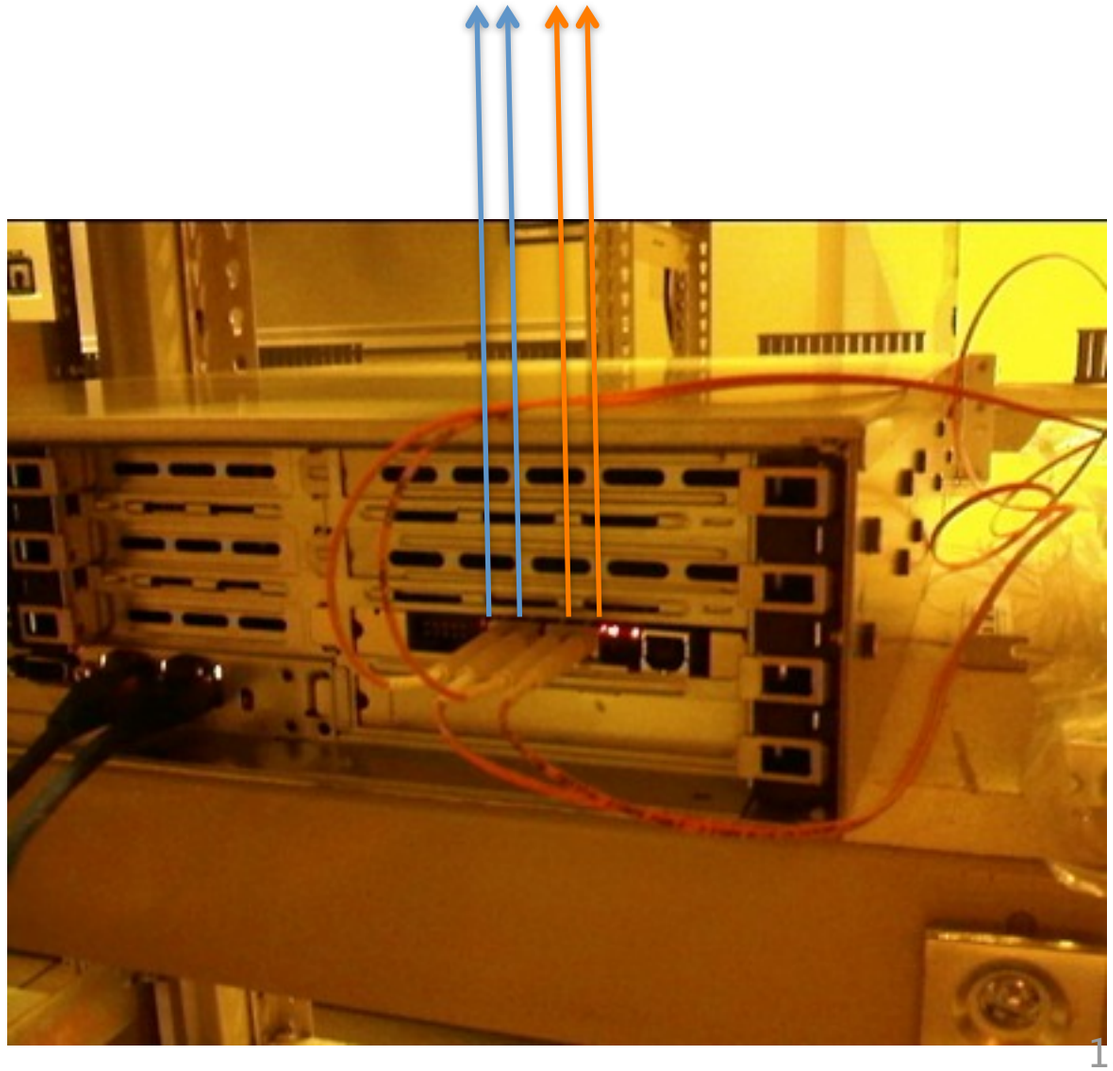
First uB SEB server (uboonedaq01.fnal.gov)
Located on a shelf in LCC at FNAL

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

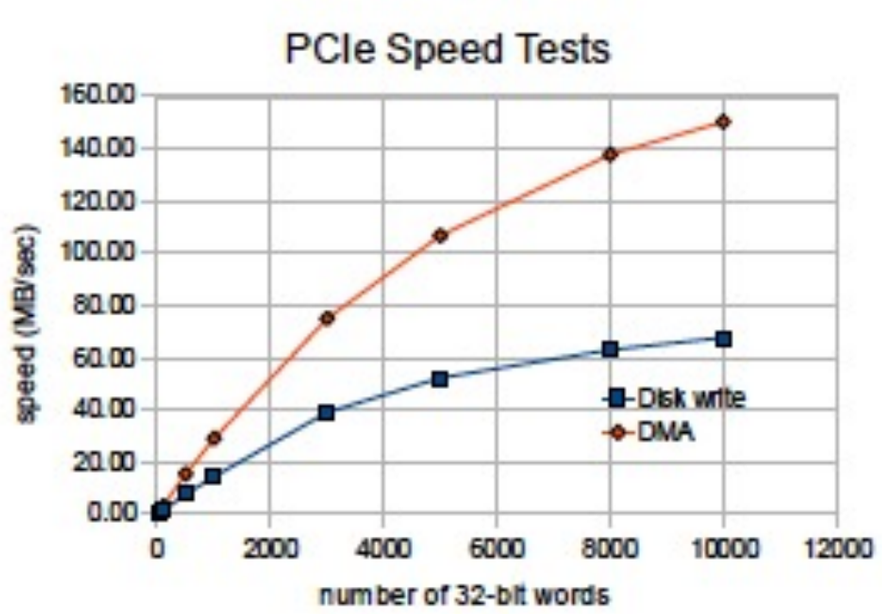
Koi 2U Dual AMD Opteron 6128 Storage Server

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

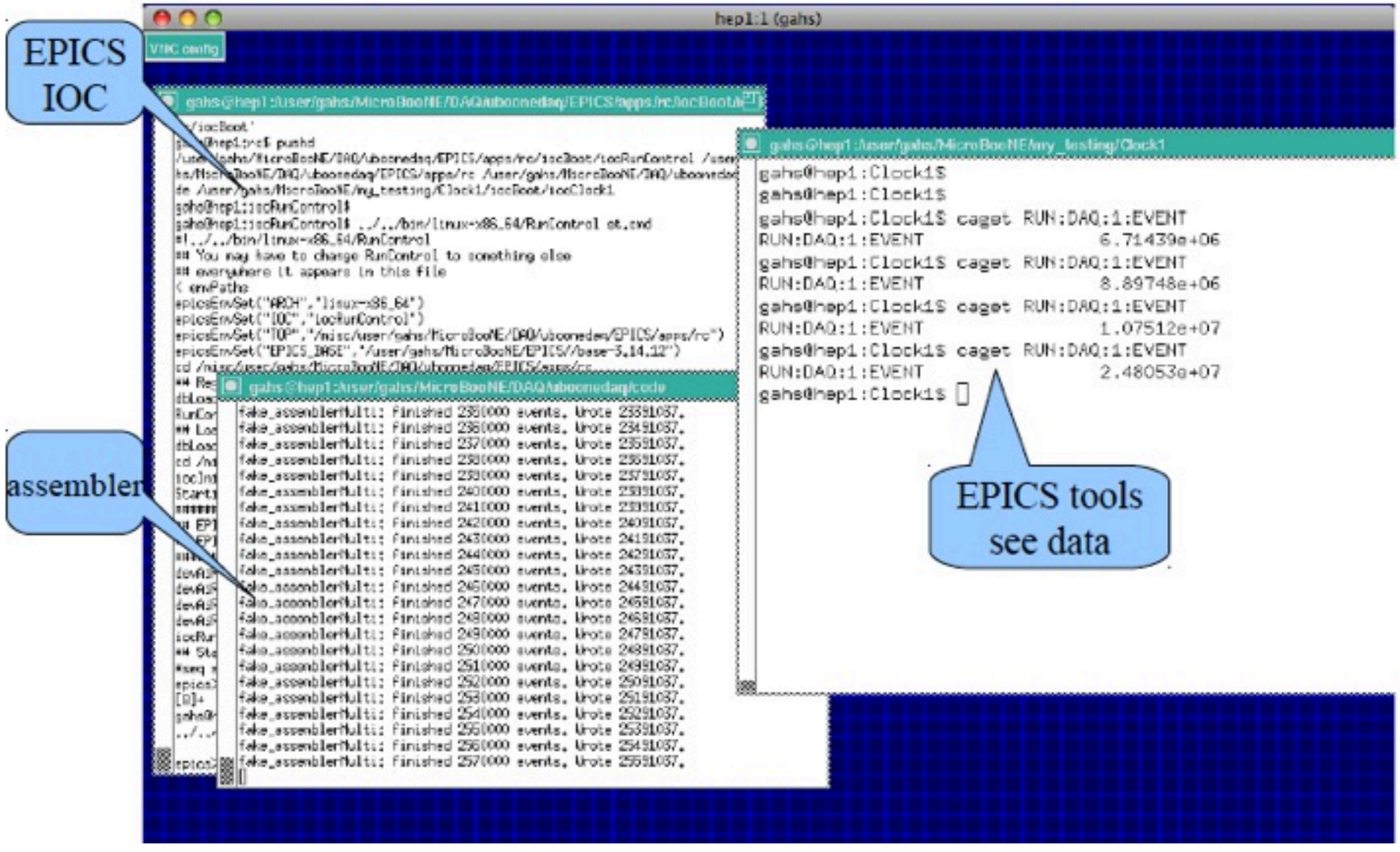
- We have a uBooNE DAQ project: https://cdcvs.fnal.gov/redmine/projects/uboonedaq
- git code repository.
- plans, instructions.
- We meet fortnightly for now.

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, ...

EPICS progress

Separate EPICS IOC and uB assembler processes, shared memory



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.}