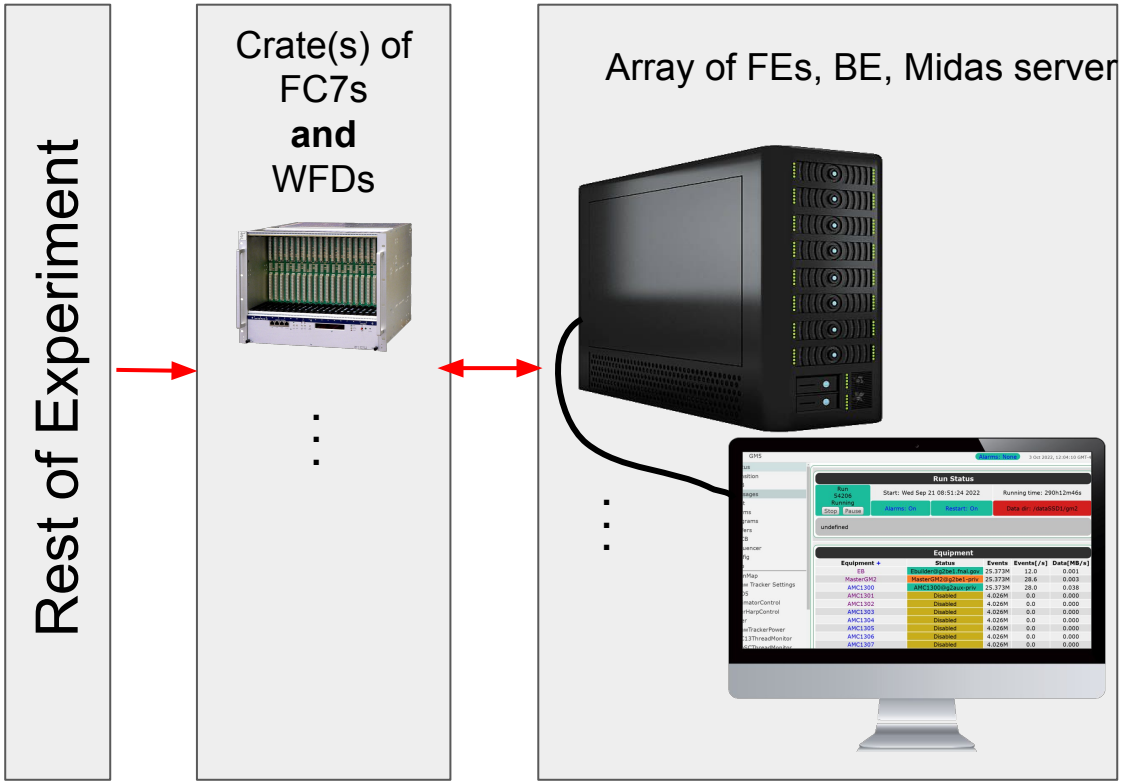
Data Acquisition (DAQ)

Jack Carlton
University of Kentucky

g-2 DAQ (Modified for One Crate Support)

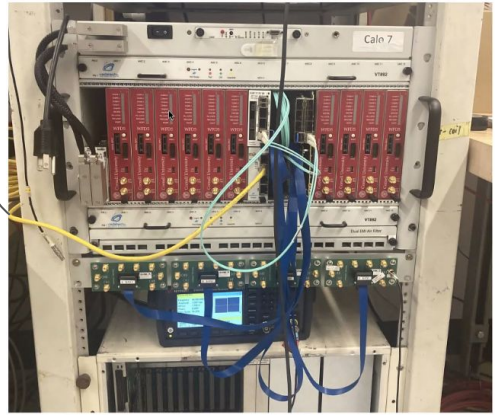
- Retains g-2 hardware, but made more flexible
- Same general process:
  - Communicate with μTCA crate, initialize hardware
  - Read TCP packets from μTCA crate
  - Write to midas data banks



Midas Framework

- C/C++ (mostly) package of modules for
  - run control,
  - expt. configuration
  - data readout
  - event building
  - data storage
  - slow control
  - alarm systems
  - Etc.
- Can link with custom software

Hardware Requirements



- Micro Telecom Computing (μTCA) crate with Modules:
  - Waveform Digitizers (WFD5(s)/Rider(s))
  - Controller (FC7)
  - MicroTCA Carrier Hub (MCH)
  - Advanced Mezzanine Card (AMC)
- “Frontend” computer with available PCIe slots for the following…
- Meinberg PCIe Clock Card
  - Custom connector
- 10 Gigabit Ethernet Network Interface Card (10GbE NIC)
  - SFP+ connectors

Software Requirements

- “Frontend” computer needs to be running Redhat-Enterprise Linux 7 (RHEL7)
  - Examples: Scientific Linux 7 (SL7), CentOS 7
- Midas
- Various other open source software libraries (root, boost, cactus, etc.)
- Some custom software libraries (DAQ frontend code, unpacking libraries, etc.)
- Software installation completely handled by installer on RHEL7 systems

Installation, in a perfect world:

```
[1] git clone
git@github.com:PIONEER-Experiment/g
m2daq-installer
[2] ./install.sh
```

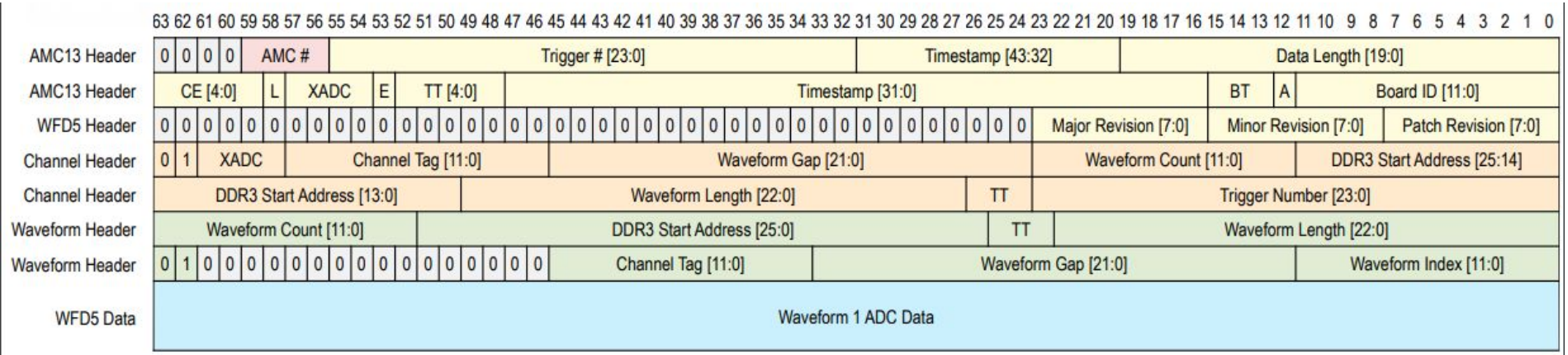
patience…

```
[3] source ./setup_environment.sh
[4] ./start_midas_webpage.sh
```

Open browser, localhost:8080

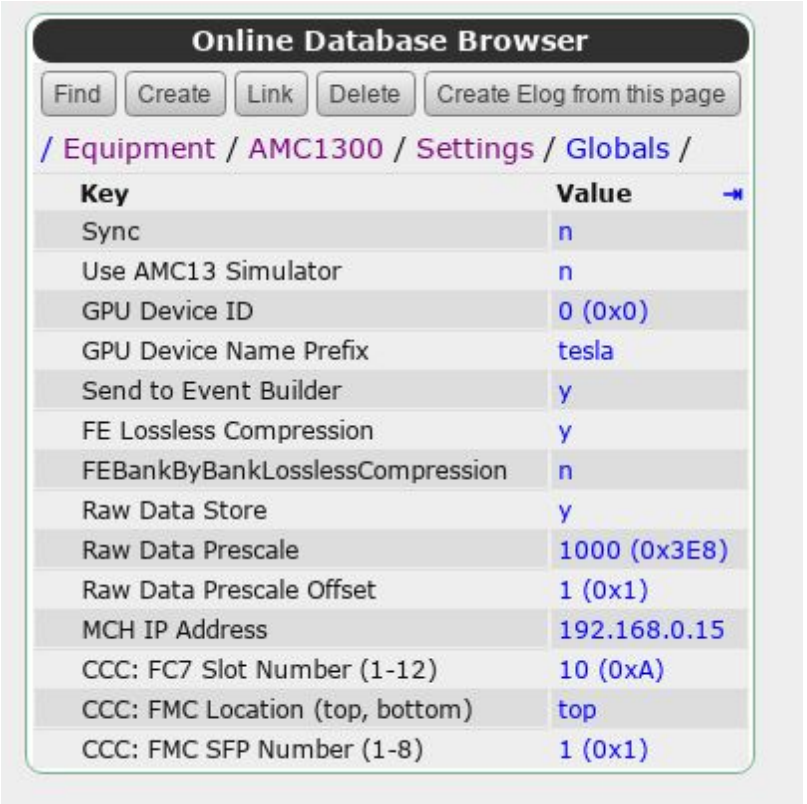
Data Output

- Data is output “raw” in midas “CR” data banks
  - Written to run{#}.mid.lz4 files by mlogger
- Unpacked C++ data structure using [unpacking library]
  - Custom analyzers can import unpacking library
  - Unpacking library include in installer



Online Database (ODB)

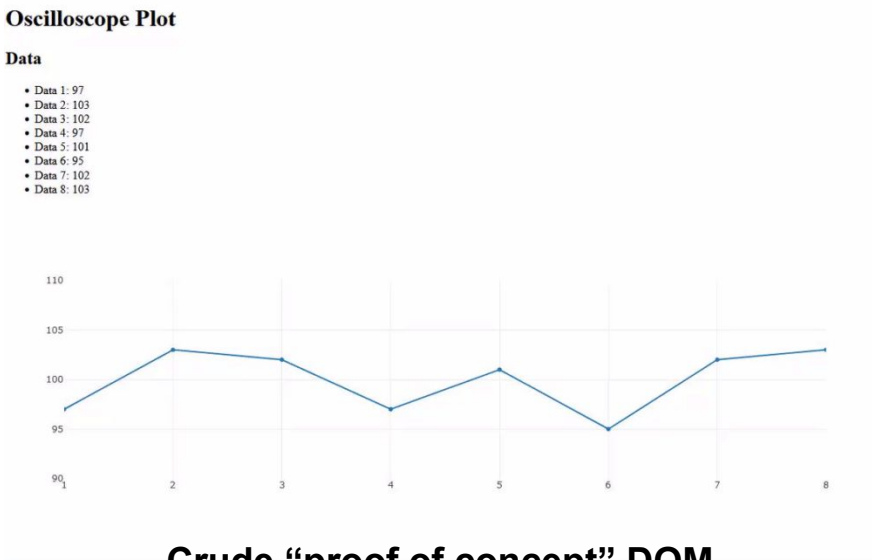
- GUI on midas webpage
  - Also available command line
- Allows for “on the fly” adjustments between runs
- Built in configurations:
  - Midas webpage
  - Logger write location
  - Webpage update rate
  - Etc.
- Custom configurations
  - Configure hardware
  - etc.



Custom Software

- Can write “clients” that connect to midas experiment
  - Python
  - C++

- Allows for user to write software to fit their needs, for example:
  - Data Quality Monitor
  - Offline Analysis
  - Automatic ODB management



Crude “proof of concept” DQM

Future Projects (Things We're Working On)

- Ensuring UW machine has running DAQ before PSI beamtime

- Improve DQM framework to be more adaptable using midas, unpacking, and ZeroMQ libraries

- Direct communication between WFDs/FPGAs and CPU/GPU using PCIe communication
  - Avoids the need for μTCA crates
  - Speeds up data transfer rate (PCIe3x8 = 8GB/s = 64 Gb/s > 10 Gb/s)
  - Possibility for direct communication to GPU (faster data processing)