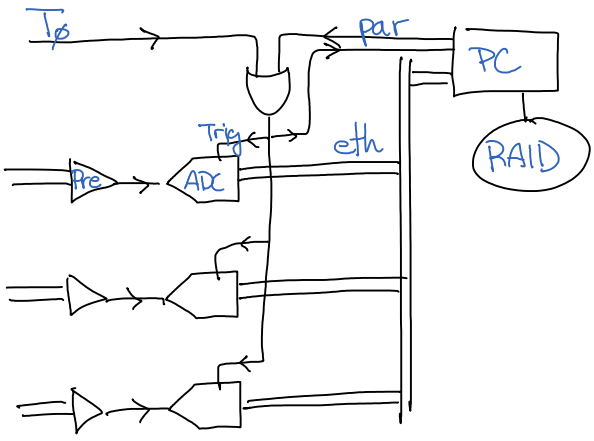
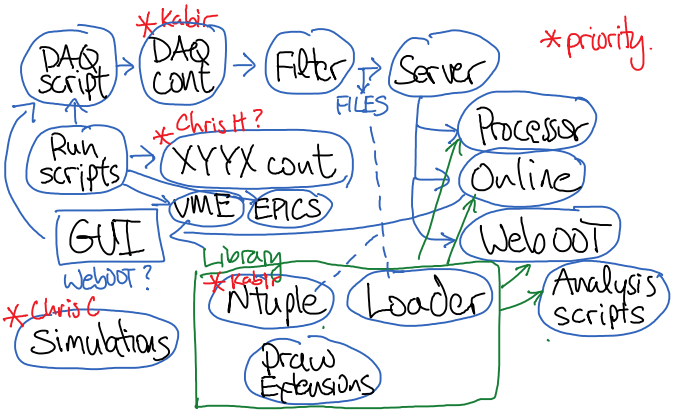
DAQ control suite

Wednesday, July 2, 2014
12:52 PM

* Hardware



* Software



* Aux Data - each to separate file

Dirty DAQ - BM, SF, XYYX on (par)
VME - proton current
EPICS - chopper, shutter, power
Chad - temp, B-field, pressure
XYYX - position
Runlist - checklist

⊛ DAQ controller - C/CLI [command line interface]

a) gate T∅ from par
b) read out ADC's >/dev/null
c) connect port >/dev/stdout
d) ungate trigger from par
e) count pulses on par to N
f) gate T∅ from par
g) logs >/dev/stderr

* Filter program - C/CLI: one instance per module

a) read event < /dev/stdin

b) analyze event header / checksum
c) output timestamp / header
d) select channels / compress > /dev/stdout
e) report errors > /dev/stdout

* Server – C / CLI, one instance per module
a) connect to socket ~/sock else report error
b) stream < /dev/stdin > ~/sock

* Data processor – write to aux. ntuples
a) data quality bits / dropped pulses – 1st pass
summary asymmetry.
b) pseudo pulse / (pedestals, gain) – 2nd pass

* Online – C++/ROOT
a) read data < files [sockets]
optional open run file / golden run file
b) update list of histograms realtime
c) check alarms...
d) command line

* Online Online – WebOOT
– webpage w/ default histograms
– clickable interface to build draw statement
– modifiable draw statement
– dump processed ntuple / histogram data

⊛ Ntuple – C++/ROOT
a) SetBranchAddress()
b) MakeSelector() – auto updates. / members in same class.
c) Aux branches
d) friends – behave w/chains
e) live analysis branches – asymmetry.

†) hierarchy: seq/pulses × det/ch.

* Loader – CINT: initializer for scripts
  - option parser ⇒ Cint variables, TOption
  - data sets from file, command line
  - chain / friend builder
  - aux data – geometry factors

* Draw extensions – ROOT
  - multidraw: extensions to Draw syntax
  - drawselector: C++ functions in Draw(" "), small classes in script.

* Analysis scripts – Cint/Python

* DAQ script – Python / CLI
  - set up sockets, online
  - keep track of run number / runlist
  - run | filter | tee > file server

⊛ XYYX controler – Python / CLI
  - move individual axes
  - keep track of current position / log file.
  - home / verify position

* VME readout – C
  - see Forest's code

* EPICS readout

* Run Scripts – Python
  - tailored for different measurements
  - call DAQ script, XYYX controller

* GUI - ROOT / Python TK / ...
  - run control widgets
  - embedded online analysis.