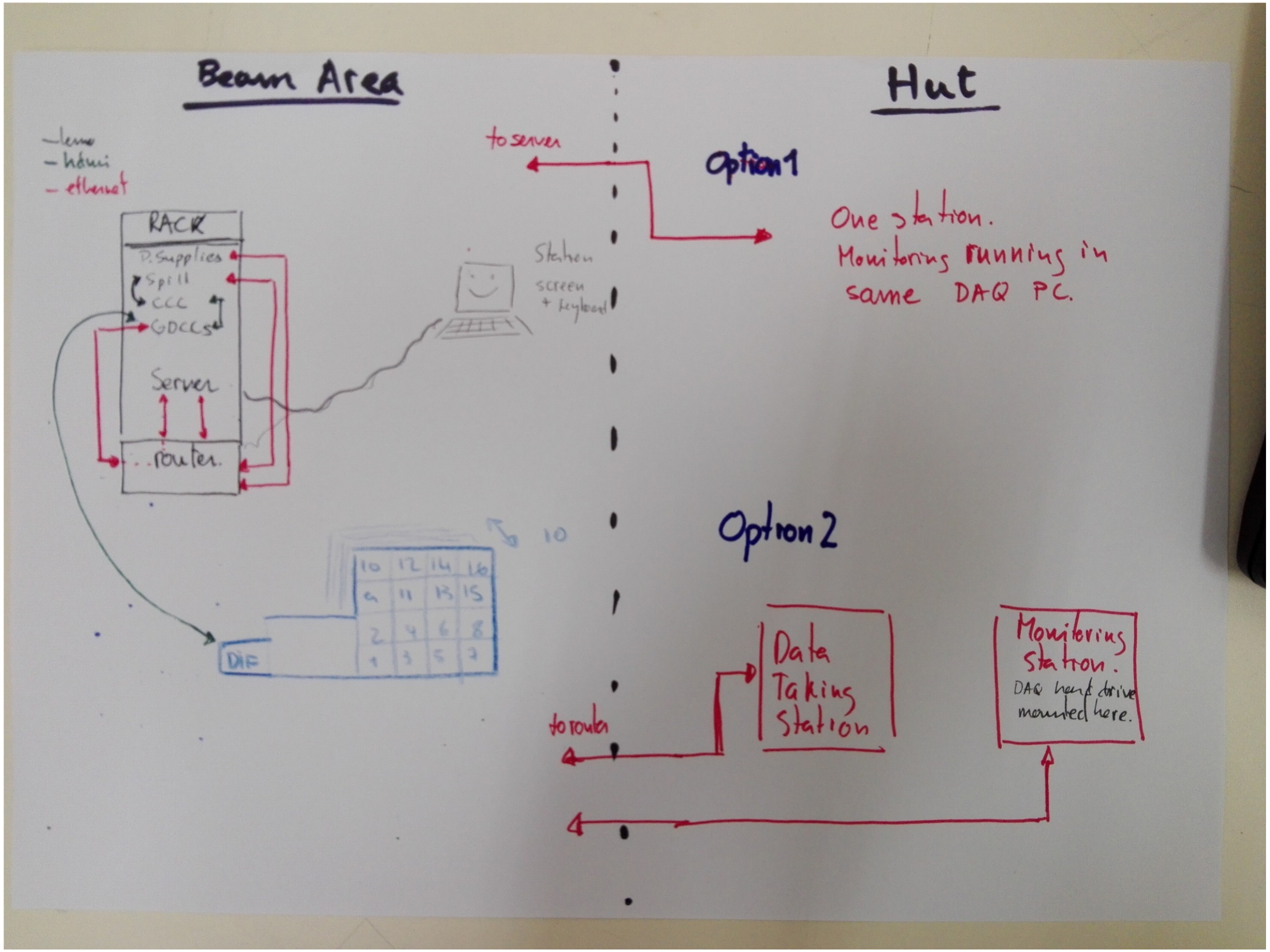
Monitoring & DAQ & Data Handling

- Networking
- Calicoes/Pyrame 3: scripting
- Monitoring
  - Sure: quasi online
  - Data Integrity
  - Online ?
- Data handling (grid ?)

SiWLC ECAL Beam Test Readiness Meeting

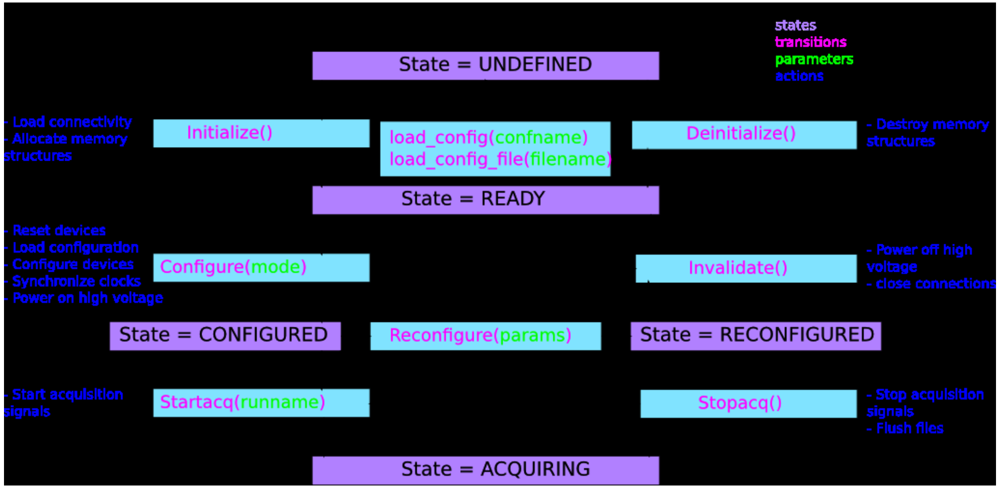
A. Irles, LAL, 24th April 2017

Networking


Beam Area
Hut
— lemo
— hdmi
— ethernet
RACK
P. Supplies
Spill
CCC
GDCCs
Server
router.
Station
screen
+ keyboard
to server
Option 1
One station.
Monitoring running in
same DAQ PC.
10
10 12 14 16
9 11 13 15
2 4 6 8
1 3 5 7
DIF
Option 2
to router
Data
Taking
Station
Monitoring
station.
DAQ hard drive
mounted here.

Calicoes

- Pyrame/calicoes 3 http://llr.in2p3.fr/sites/pyrame/calicoes/documentation/
  - Calicoes is a software for control-command and data acquisition of the Silicium/Tungsten Electromagnetic Calorimeter (SIW-Ecal) for the future ILC Detector. It is based on the Pyrame framework.
- FSM, graphical interface, scripting



states
transitions
parameters
actions

State = UNDEFINED

- Load connectivity
- Allocate memory structures

Initialize()

load_config(confname)
load_config_file(filename)

Deinitialize()

- Destroy memory structures

State = READY

- Reset devices
- Load configuration
- Configure devices
- Synchronize clocks
- Power on high voltage

Configure(mode)

Invalidate()

- Power off high voltage
- close connections

State = CONFIGURED

Reconfigure(params)

State = RECONFIGURED

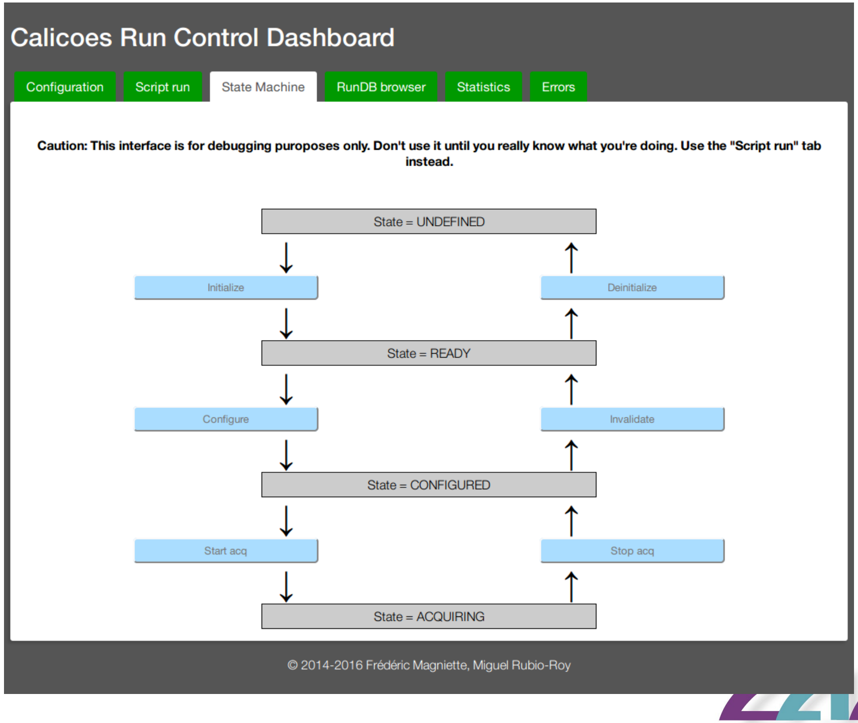
- Start acquisition signals

Startacq(runname)

Stopacq()

- Stop acquisition signals
- Flush files

State = ACQUIRING



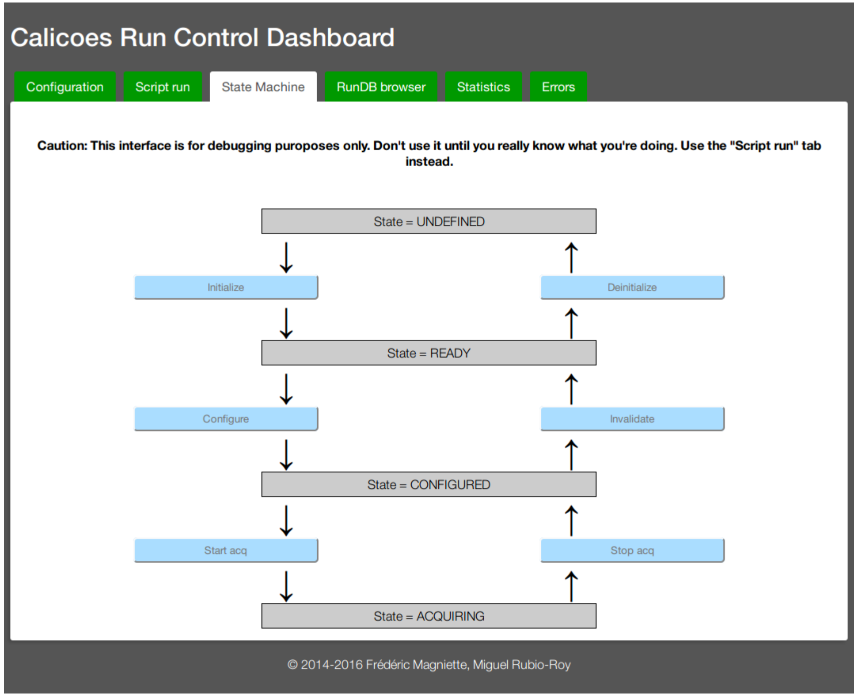
Calicoes Run Control Dashboard

Caution: This interface is for debugging puroposes only. Don't use it until you really know what you're doing. Use the "Script run" tab instead.

© 2014-2016 Frédéric Magniette, Miguel Rubio-Roy

Calicoes

- Data Taking (graphical interface)
- Initilization/configuration done only once (or every time we change the configuration file)
- Start/Stop manually:
  - Run name decision done manually
  - Stop manually
  - No conversion done after the run.



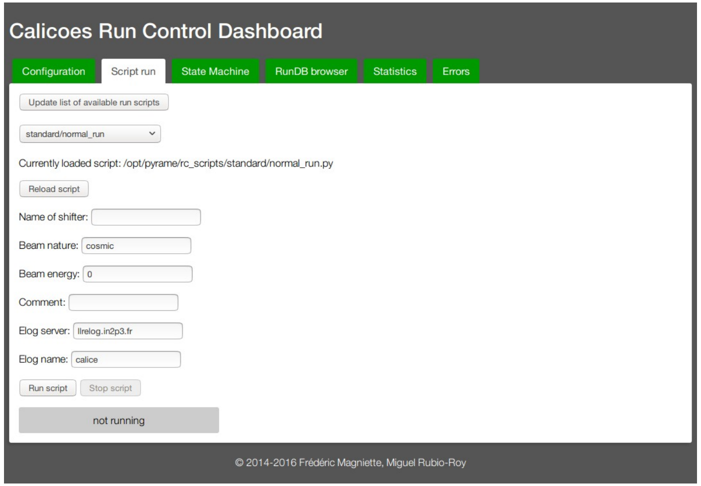
Calicoes Run Control Dashboard

Caution: This interface is for debugging puroposes only. Don't use it until you really know what you're doing. Use the "Script run" tab instead.

© 2014-2016 Frédéric Magniette, Miguel Rubio-Roy

Calicoes

Scripting


Calicoes Run Control Dashboard

Currently loaded script: /opt/pyrame/rc_scripts/standard/normal_run.py

Name of shifter:

Beam nature: cosmic

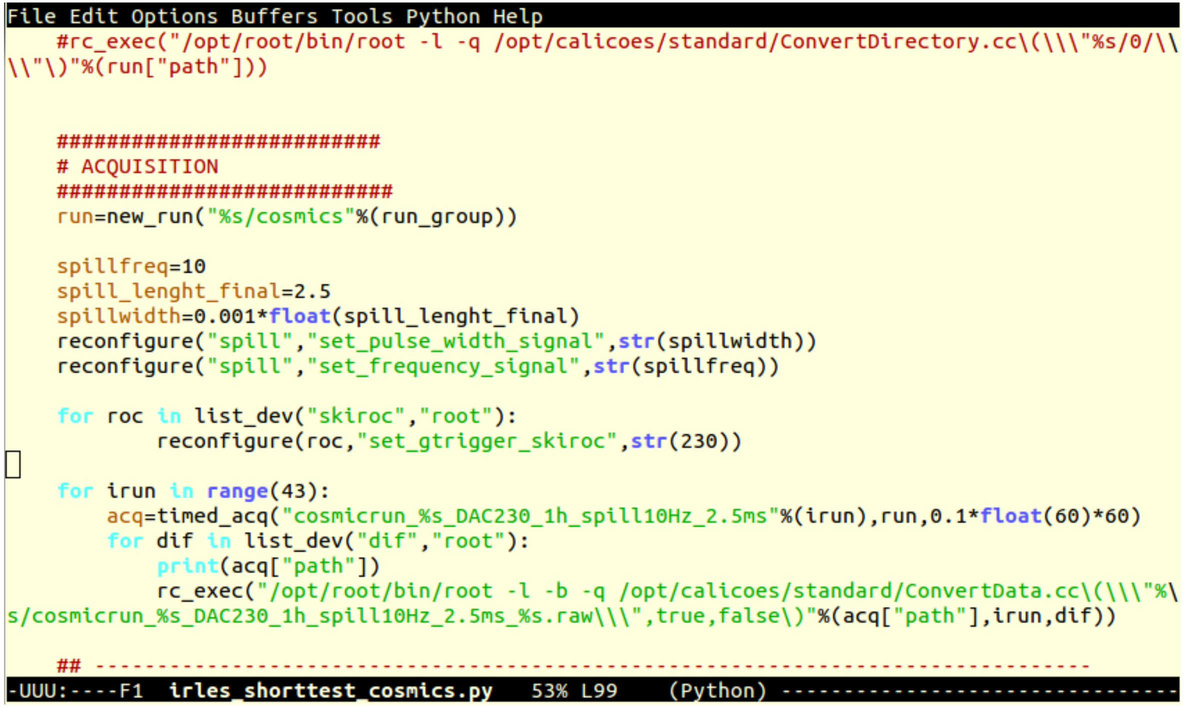
Beam energy: 0

Comment:

Elog server: llrelog.in2p3.fr

Elog name: calice

© 2014-2016 Frédéric Magniette, Miguel Rubio-Roy



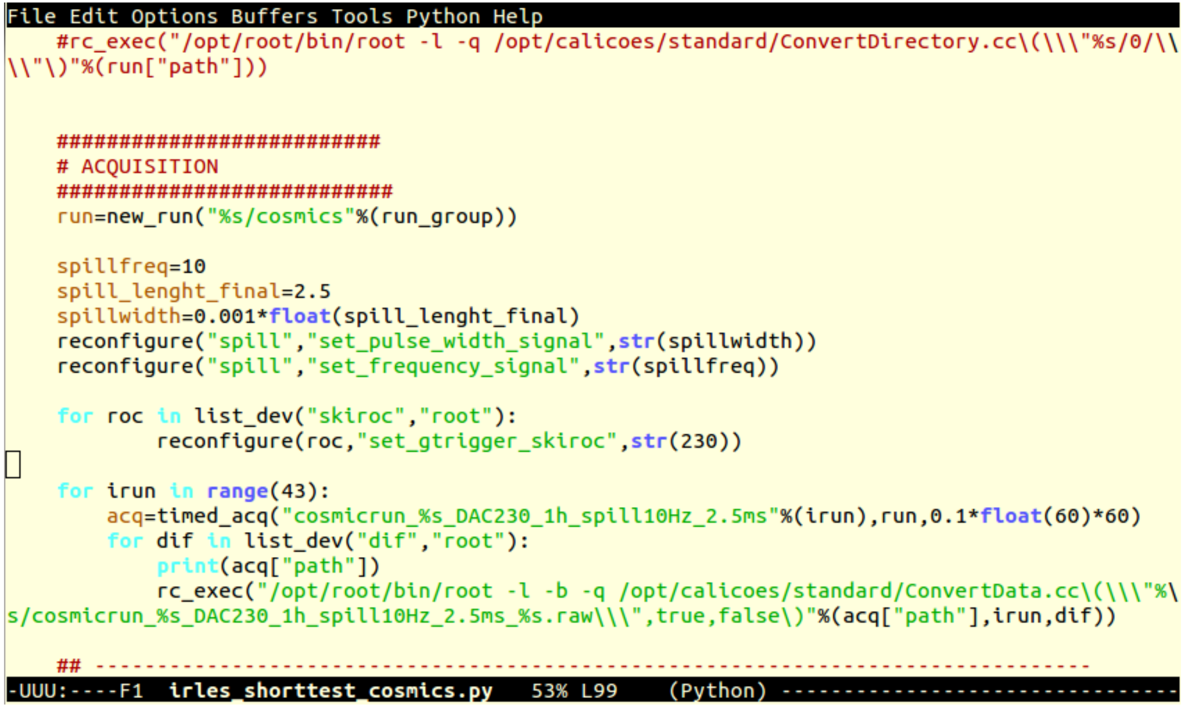
```
    #rc_exec("/opt/root/bin/root -l -q /opt/calicoes/standard/ConvertDirectory.cc\(\\\"%s/0/\\
\\"\)"%(run["path"]))


    ##########################
    # ACQUISITION
    ###########################
    run=new_run("%s/cosmics"%(run_group))

    spillfreq=10
    spill_lenght_final=2.5
    spillwidth=0.001*float(spill_lenght_final)
    reconfigure("spill","set_pulse_width_signal",str(spillwidth))
    reconfigure("spill","set_frequency_signal",str(spillfreq))

    for roc in list_dev("skiroc","root"):
            reconfigure(roc,"set_gtrigger_skiroc",str(230))

    for irun in range(43):
        acq=timed_acq("cosmicrun_%s_DAC230_1h_spill10Hz_2.5ms"%(irun),run,0.1*float(60)*60)
        for dif in list_dev("dif","root"):
            print(acq["path"])
            rc_exec("/opt/root/bin/root -l -b -q /opt/calicoes/standard/ConvertData.cc\(\\\"%\
s/cosmicrun_%s_DAC230_1h_spill10Hz_2.5ms_%s.raw\\\",true,false\)"%(acq["path"],irun,dif))

    ## --------------------------------------------------------------------------------
-UUU:----F1  irles_shorttest_cosmics.py    53% L99    (Python) --------------------------------
```

Calicoes

- Easy conversion after every acquisition → rc_exec allows to run terminal commands.
- Suitable for long runs
- Easy change of configurations (scans, etc) → reconfigure command
- Run hierarchy (folder):

  folder_created_by_thescript_withdate/subrun_folder/

  - Acq1name_difID.raw (and root)
  - Acq2name_difID.raw
  - ...



```
    #rc_exec("/opt/root/bin/root -l -q /opt/calicoes/standard/ConvertDirectory.cc\(\\\"%s/0/\\
\\"\)"%(run["path"]))


    ##########################
    # ACQUISITION
    ###########################
    run=new_run("%s/cosmics"%(run_group))

    spillfreq=10
    spill_lenght_final=2.5
    spillwidth=0.001*float(spill_lenght_final)
    reconfigure("spill","set_pulse_width_signal",str(spillwidth))
    reconfigure("spill","set_frequency_signal",str(spillfreq))

    for roc in list_dev("skiroc","root"):
            reconfigure(roc,"set_gtrigger_skiroc",str(230))

    for irun in range(43):
        acq=timed_acq("cosmicrun_%s_DAC230_1h_spill10Hz_2.5ms"%(irun),run,0.1*float(60)*60)
        for dif in list_dev("dif","root"):
            print(acq["path"])
            rc_exec("/opt/root/bin/root -l -b -q /opt/calicoes/standard/ConvertData.cc\(\\\"%\
s/cosmicrun_%s_DAC230_1h_spill10Hz_2.5ms_%s.raw\\\",true,false\)"%(acq["path"],irun,dif))

    ## --------------------------------------------------------------------------------
```

-UUU:----F1 irles_shorttest_cosmics.py   53% L99   (Python) ----------------------------------

Optimal for individual runs ?

Monitoring

- Scripting allows to do quasi online conversion to root file and run monitoring
  - Monitoring in the same machine ?
  - I vote for a different →

    With DAQ Hard drive mounted on it.

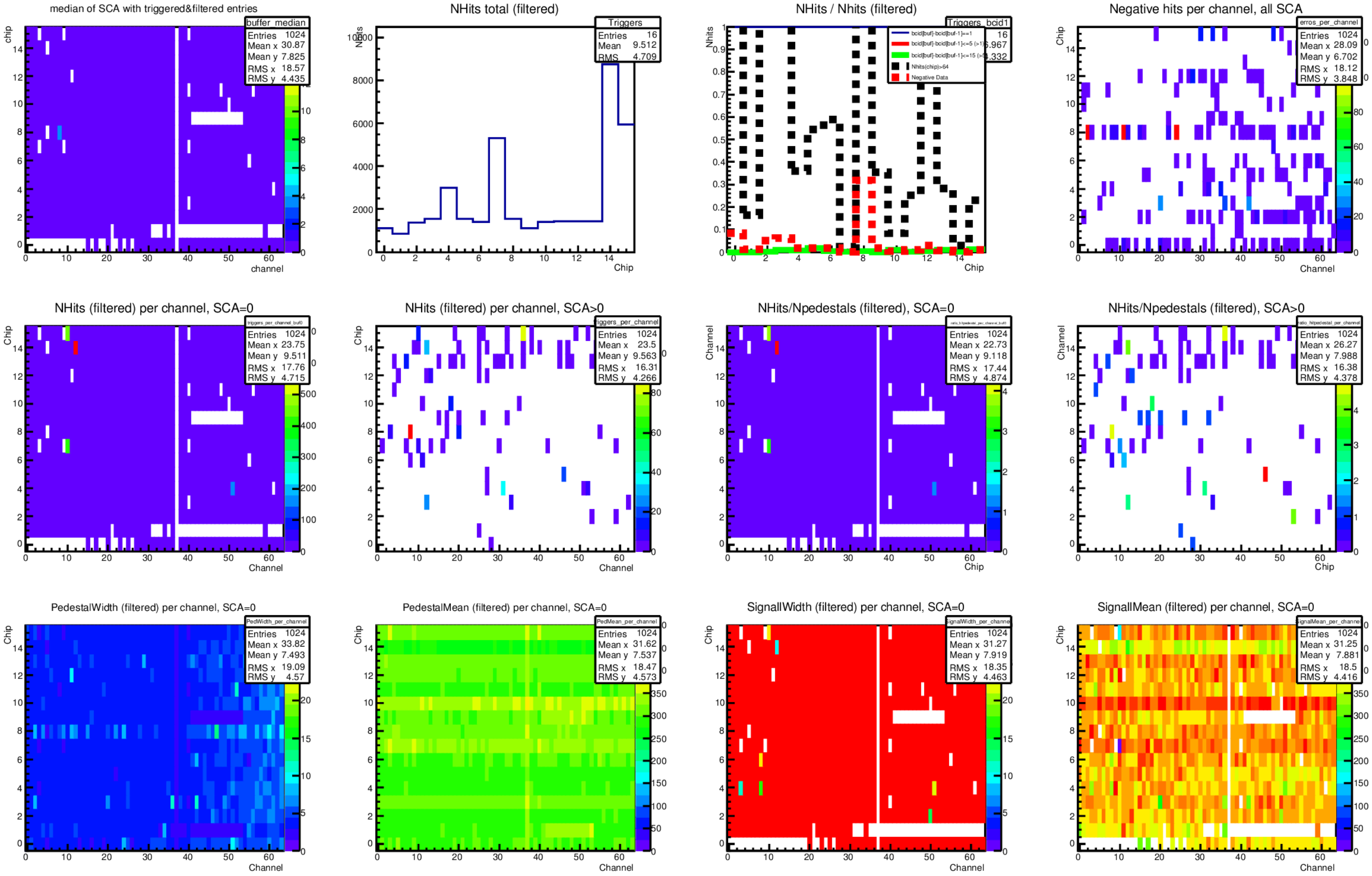
    At the end of the conversion → send monitoring command via ssh.

Monitoring: tpecal

- **Semionline Data Quality** analysis framework that uses **root** files: temporary repository https://github.com/airqui/tpecal/
- Fulfills two functions:
  - Quick monitoring (chip and channel modules)
  - Quick analysis module manager: scurves, pedestal (ADC) extraction, MIP fit with pedestal subtraction, etc.
- Executable: easy to script it
  - ./tpecalana FILETOMONITOR FILETOSAVE MonitorType
  - We can analyze several slabs in parallel and perform different analysis
- For the moment it does single SLAB monitoring, not correlations. :-(
- Work in progress. Preliminary plots. → see examples

MonitorType = MonitorChannel

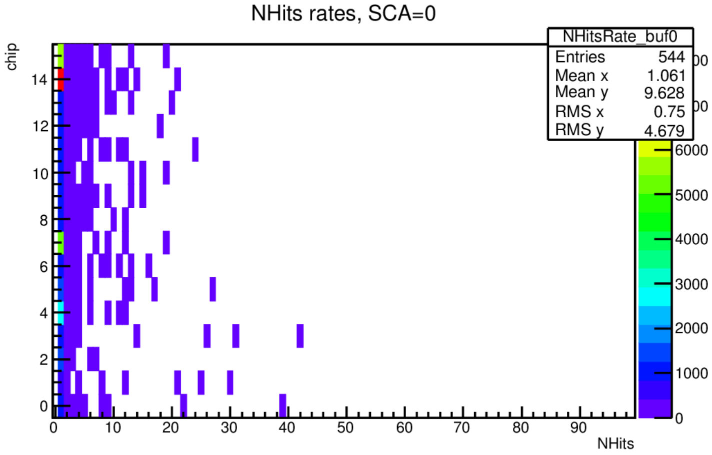
- Takes ~60 s to analyze 36h cosmic run

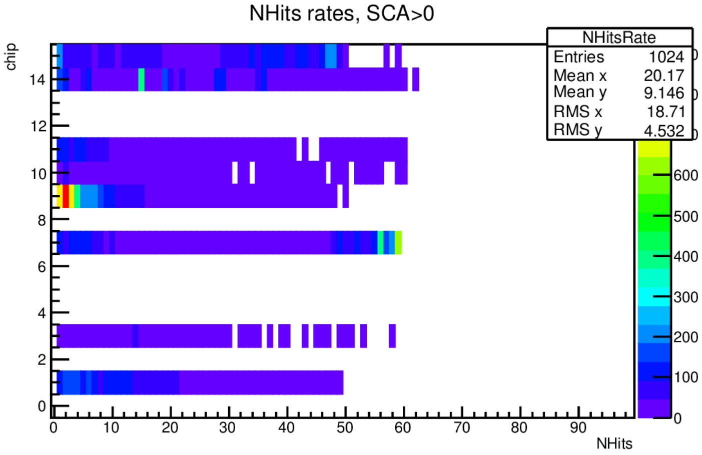


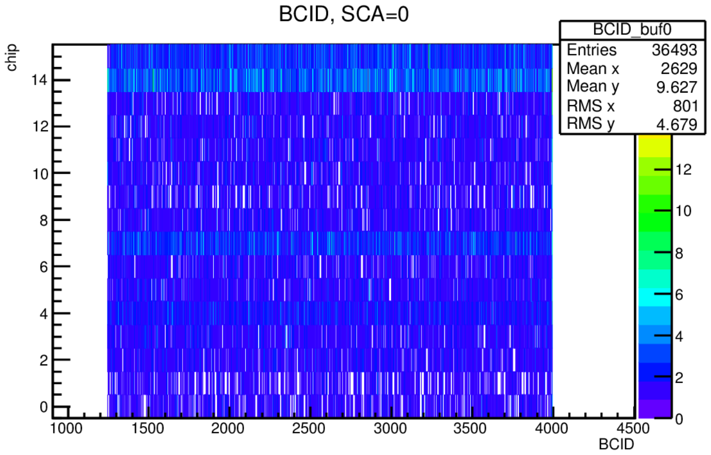
LABORATOIRE DE L'ACCÉLÉRATEUR LINÉAIRE

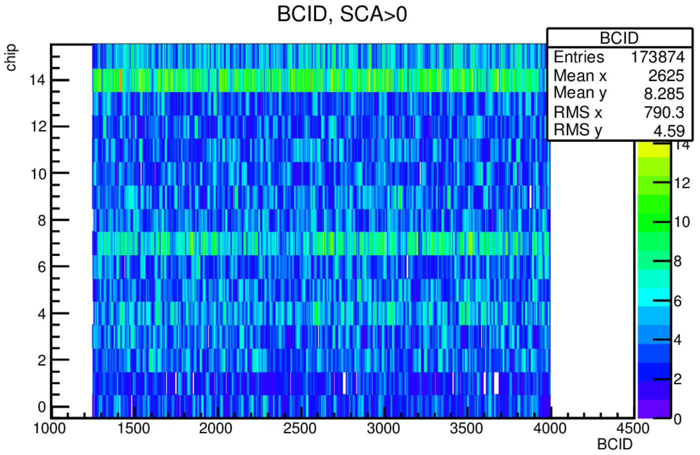
MonitorType = MonitorChip

- Takes ~40 s to analyze 36h cosmic run







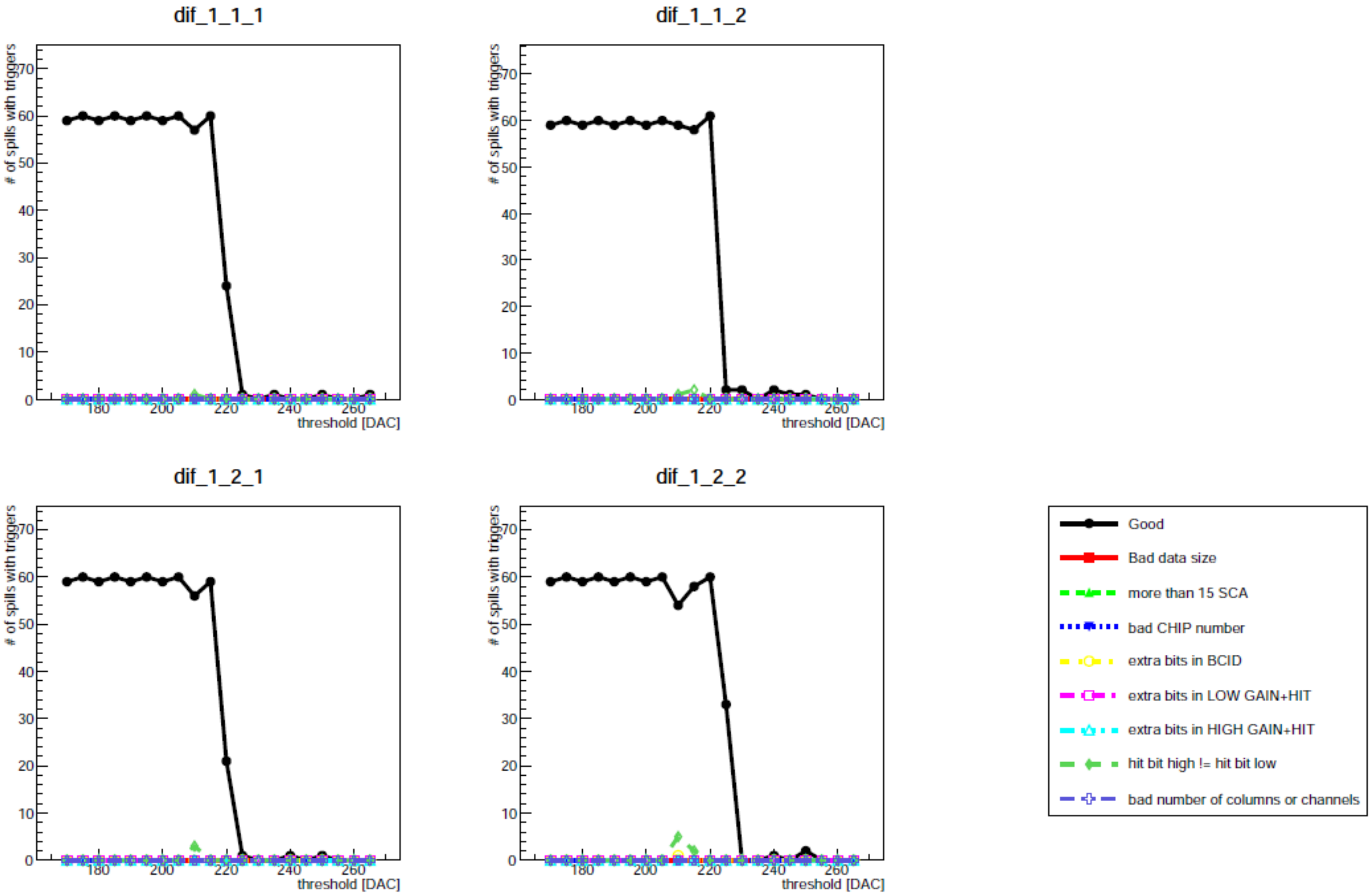


MonitorType = MonitorDIF, ECAL …

- Number of correlated hits,
- Hit maps, pedestal maps grids, data integrity rates

Data Integrity

- Data Integrity checks have been added to the raw to root converter → screen output and histogram saved
  - Online ?
- It has proven to be very useful when optimizing the spill length for cosmic runs


dif_1_1_1
dif_1_1_2
dif_1_2_1
dif_1_2_2
# of spills with triggers
threshold [DAC]
Good
Bad data size
more than 15 SCA
bad CHIP number
extra bits in BCID
extra bits in LOW GAIN+HIT
extra bits in HIGH GAIN+HIT
hit bit high != hit bit low
bad number of columns or channels