Development of versatile online monitoring for EUDAQ2

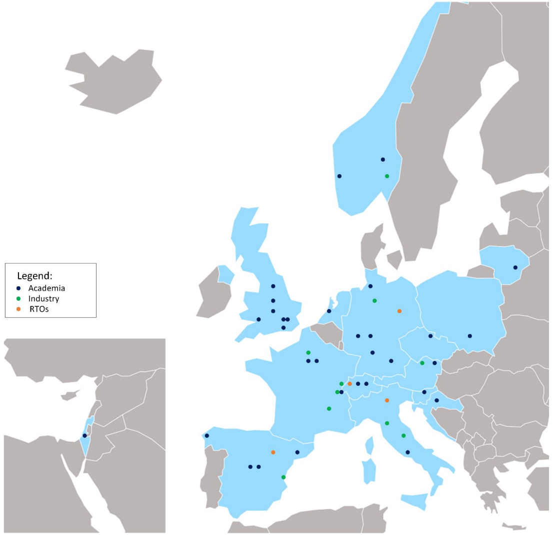
Andreas Loeschcke Centeno
a.loeschcke-centeno@sussex.ac.uk

Fabrizio Salvatore, Lennart Huth (lennart.huth@desy.de)

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101004761.

AIDAinnova

- 4 year programme (30 M€ Budget)
- Development of common detector projects
- Upgrades to research infrastructure
- 42 participants from 15 countries



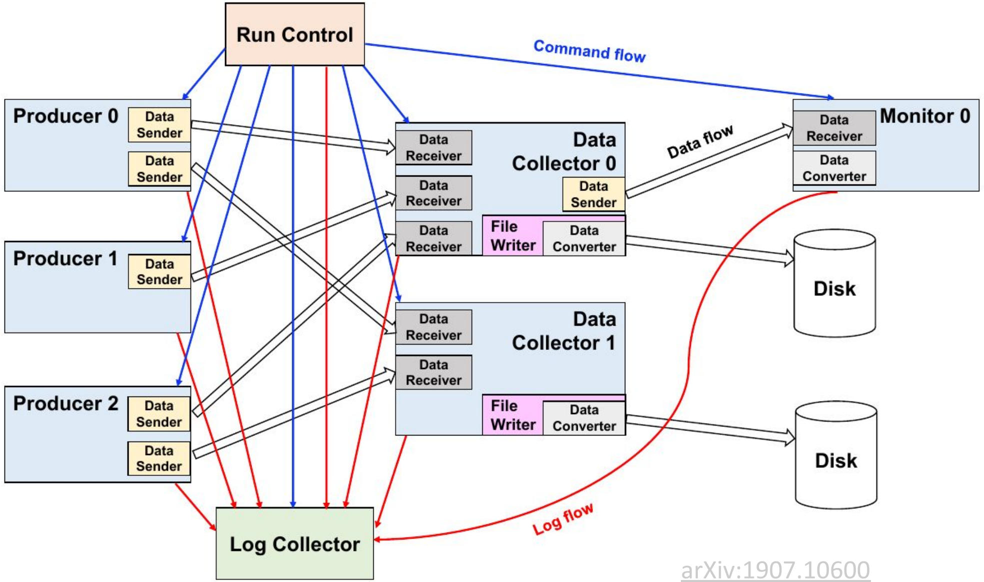
AIDAinnova

WP 3: Test beam and DAQ infrastructure

- Development of common test beam resources (hardware & software)
- Task 3.4: Development of versatile online monitoring for EUDAQ2
  - Objective: release monitoring tool usable by any test beam user with any kind of detector

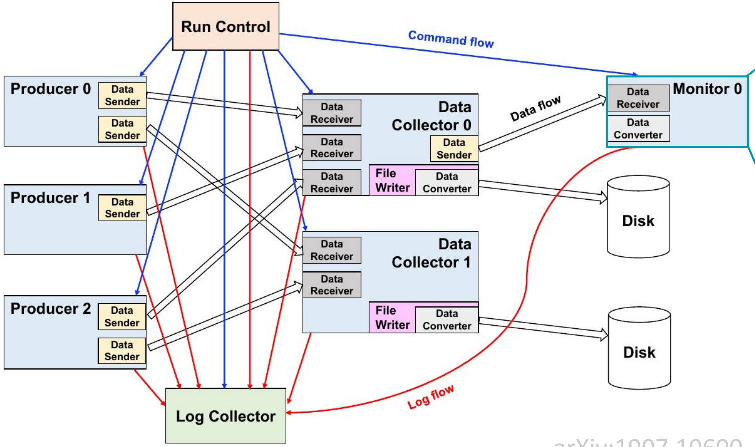
EUDAQ2 [repository, manual]

- Data Acquisition Control System
  - ○ Provides central DAQ control, data handling and storage, log collection
  - ○ Distributes all commands among all modules
- Architecture based on runtime-instances
  - ○ Producer is interface between custom DAQ for DUT and EUDAQ2
  - ○ DataCollector for merging/synchronising data
  - ○ DataConverter from raw data to StdEvent

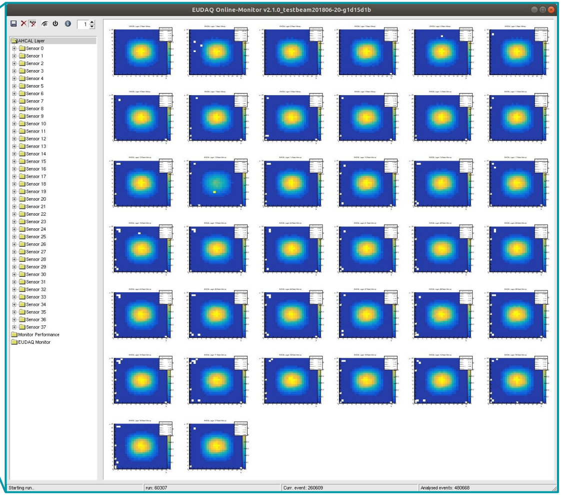


arXiv:1907.10600

Monitoring with EUDAQ2



arXiv:1907.10600



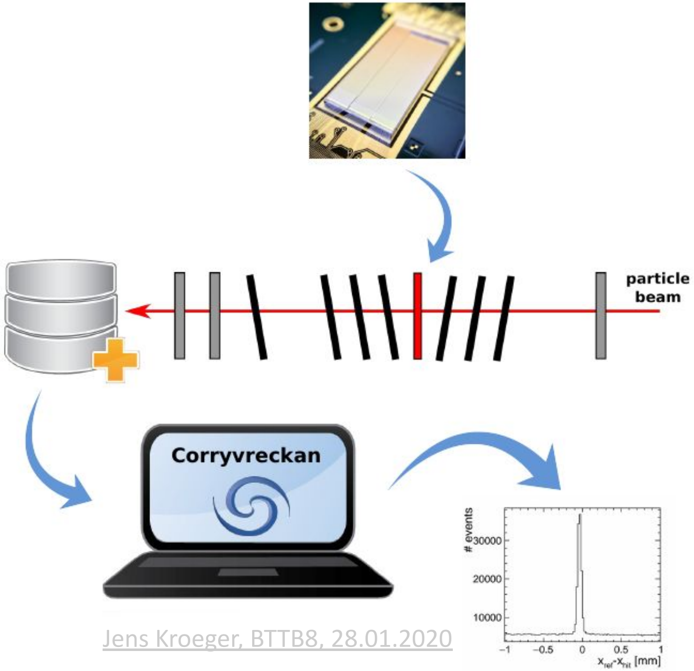
Starting run.. | run: 60307 | Curr. event: 260609 | Analysed events: 480668

- No flexible event building
- No information about detector geometry
- No advanced reconstruction
- But widely used software corryvreckan provides everything we need
  - ➢ **Idea: Combine corryvreckan with EUDAQ2 for monitoring**

Corryvreckan: Introduction

- Widely used track reconstruction and analysis software for test beam data
- Developed by pixel community, but not limited to pixel sensors
- Versatile, highly configurable with modular structure
  - Lightweight framework core
  - Modules for specific tasks, e.g.:
    - Event Loading
    - Clustering
    - Tracking
    - Offline alignment
- Graphical user interface for monitoring of reconstruction progress
- Modern C++, minimal external dependencies
- Extensive documentation

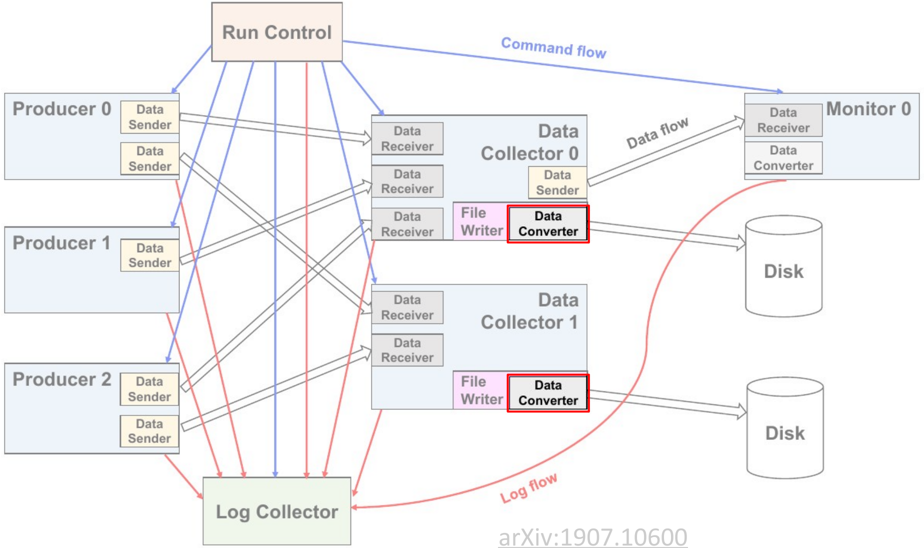
BTTB12 Corryvreckan Tutorial



Jens Kroeger, BTTB8, 28.01.2020

Corryvreckan: A Perfect Candidate

- Direct compatibility with EUDAQ2
    - Load EUDAQ2 raw events via EventLoaderEUDAQ2 module
    - Requires eudaq::StdEventConverter

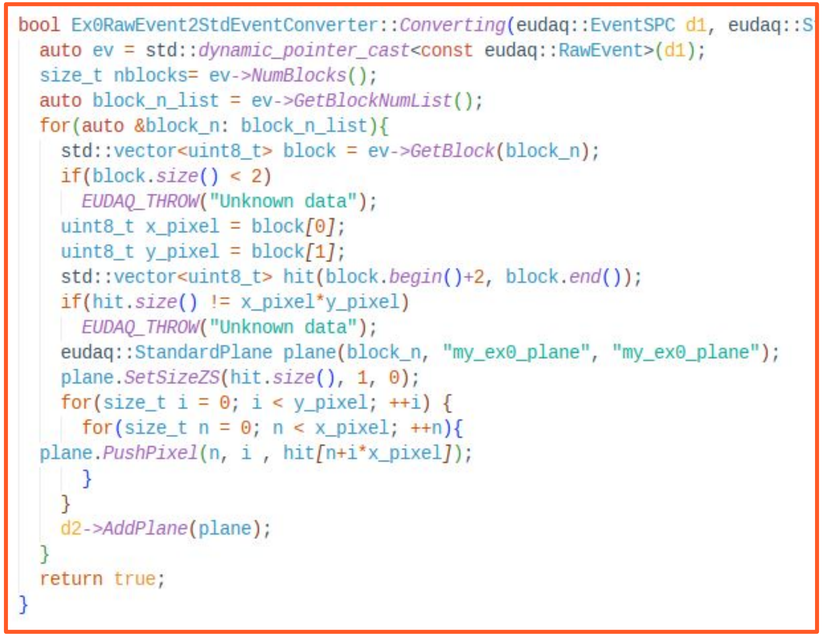


arXiv:1907.10600

Corryvreckan: A Perfect Candidate

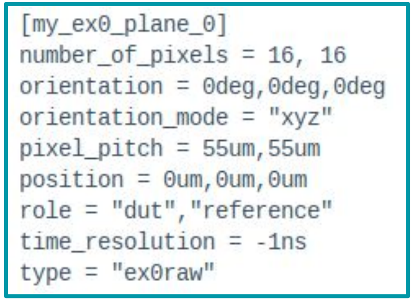
- Direct compatibility with EUDAQ2
  - Load EUDAQ2 raw events via EventLoaderEUDAQ2 module
  - Requires eudaq::StdEventConverter
- Geometry information through .geo file
  - Nomenclature for pixels, but can be used generically
  - Mapping to pixels and planes done in StdEventConverter

Example eudaq::RawEvent2StdEventConverter



```
bool Ex0RawEvent2StdEventConverter::Converting(eudaq::EventSPC d1, eudaq::S
  auto ev = std::dynamic_pointer_cast<const eudaq::RawEvent>(d1);
  size_t nblocks= ev->NumBlocks();
  auto block_n_list = ev->GetBlockNumList();
  for(auto &block_n: block_n_list){
    std::vector<uint8_t> block = ev->GetBlock(block_n);
    if(block.size() < 2)
      EUDAQ_THROW("Unknown data");
    uint8_t x_pixel = block[0];
    uint8_t y_pixel = block[1];
    std::vector<uint8_t> hit(block.begin()+2, block.end());
    if(hit.size() != x_pixel*y_pixel)
      EUDAQ_THROW("Unknown data");
    eudaq::StandardPlane plane(block_n, "my_ex0_plane", "my_ex0_plane");
    plane.SetSizeZS(hit.size(), 1, 0);
    for(size_t i = 0; i < y_pixel; ++i) {
      for(size_t n = 0; n < x_pixel; ++n){
  plane.PushPixel(n, i , hit[n+i*x_pixel]);
      }
    }
    d2->AddPlane(plane);
  }
  return true;
}
```

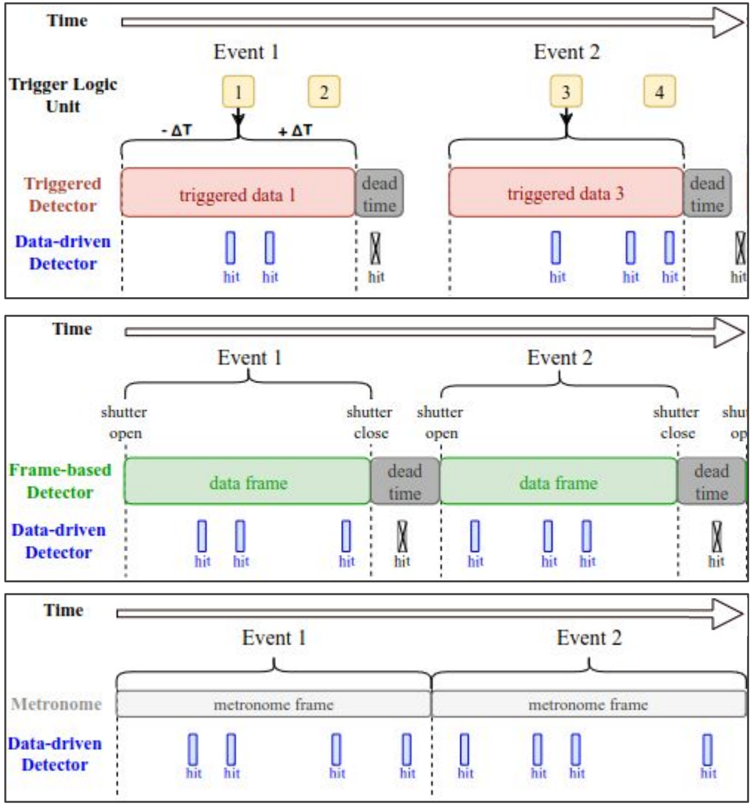
Geometry file



```
[my_ex0_plane_0]
number_of_pixels = 16, 16
orientation = 0deg,0deg,0deg
orientation_mode = "xyz"
pixel_pitch = 55um,55um
position = 0um,0um,0um
role = "dut","reference"
time_resolution = -1ns
type = "ex0raw"
```

Corryvreckan: A Perfect Candidate

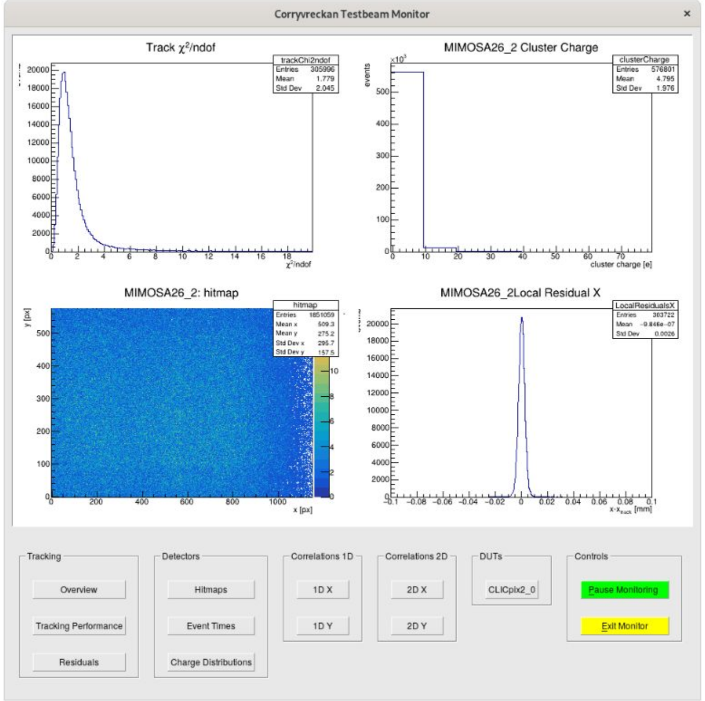
- Direct compatibility with EUDAQ2
  - Load EUDAQ2 raw events via EventLoaderEUDAQ2 module
  - Requires eudaq::StdEventConverter
- Geometry information through .geo file
  - Nomenclature for pixels, but can be used generically
  - Mapping to pixels and planes done in StdEventConverter
- Flexible event building
  - Trigger-based, frame-based, data-driven



arXiv:2011.12730

Corryvreckan: A Perfect Candidate

- Direct compatibility with EUDAQ2
  - Load EUDAQ2 raw events via EventLoaderEUDAQ2 module
  - Requires eudaq::StdEventConverter
- Geometry information through .geo file
  - Nomenclature for pixels, but can be used generically
  - Mapping to pixels and planes done in StdEventConverter
- Flexible event building
  - Trigger-based, frame-based, data-driven
- Monitoring of reconstruction chain (event loading, clustering, tracking)



Corryvreckan Testbeam Monitor

trackChi2ndof
Entries 305996
Mean 1.779
Std Dev 2.045

clusterCharge
Entries 576801
Mean 4.795
Std Dev 1.976

hitmap
Entries 1851059
Mean x 509.3
Mean y 275.2
Std Dev x 295.7
Std Dev y 157.5

LocalResidualsX
Entries 303722
Mean -9.846e-07
Std Dev 0.0026

arXiv:2011.12730

Corryvreckan: A Perfect Candidate

- Direct compatibility with EUDAQ2
    - Load EUDAQ2 raw events via EventLoaderEUDAQ2 module
    - Requires eudaq::StdEventConverter
- Geometry information through .geo file
    - Nomenclature for pixels, but can be used generically
    - Mapping to pixels and planes done in StdEventConverter
- Flexible event building
    - Trigger-based, frame-based, data-driven
- Monitoring of reconstruction chain (event loading, clustering, tracking)
- Only need to figure out how to seamlessly integrate into EUDAQ2: **CorryMonitor**

CorryMonitor

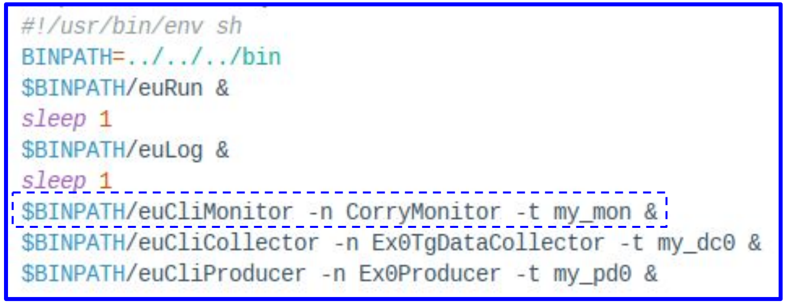
Lennart Huth, AIDAinnova 2nd Annual Meeting, 27.04.2023

Maximise user convenience (while maintaining flexibility):

- Fully integrated into EUDAQ2
  - ○ Start and stop from RunControl
- Require minimal user input to get monitoring running
- Corryvreckan only capable of *semi*-online monitoring
  - ○ Events not sent directly but read from file
  - ➢ Automatically finds correct files to monitor and pass on start-up
- Able to read files on different machines

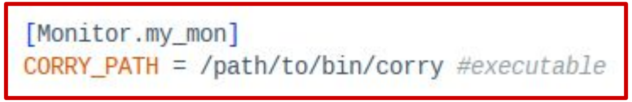
How to CorryMonitor

1. Startup



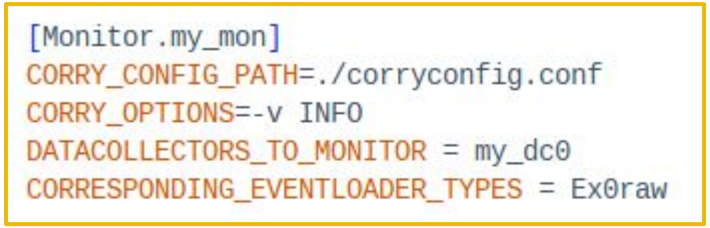
```
#!/usr/bin/env sh
BINPATH=../../../bin
$BINPATH/euRun &
sleep 1
$BINPATH/euLog &
sleep 1
$BINPATH/euCliMonitor -n CorryMonitor -t my_mon &
$BINPATH/euCliCollector -n Ex0TgDataCollector -t my_dc0 &
$BINPATH/euCliProducer -n Ex0Producer -t my_pd0 &
```

2. EUDAQ2 .ini file



```
[Monitor.my_mon]
CORRY_PATH = /path/to/bin/corry #executable
```

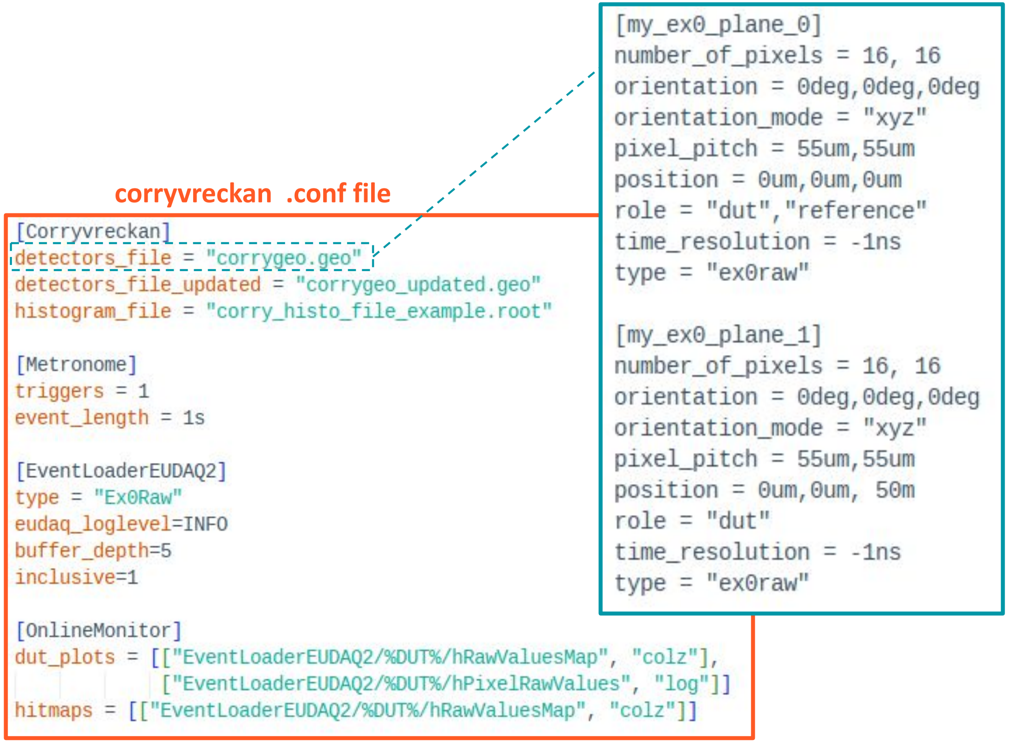
3. EUDAQ2 .conf file



```
[Monitor.my_mon]
CORRY_CONFIG_PATH=./corryconfig.conf
CORRY_OPTIONS=-v INFO
DATACOLLECTORS_TO_MONITOR = my_dc0
CORRESPONDING_EVENTLOADER_TYPES = Ex0raw
```

corryvreckan .conf file

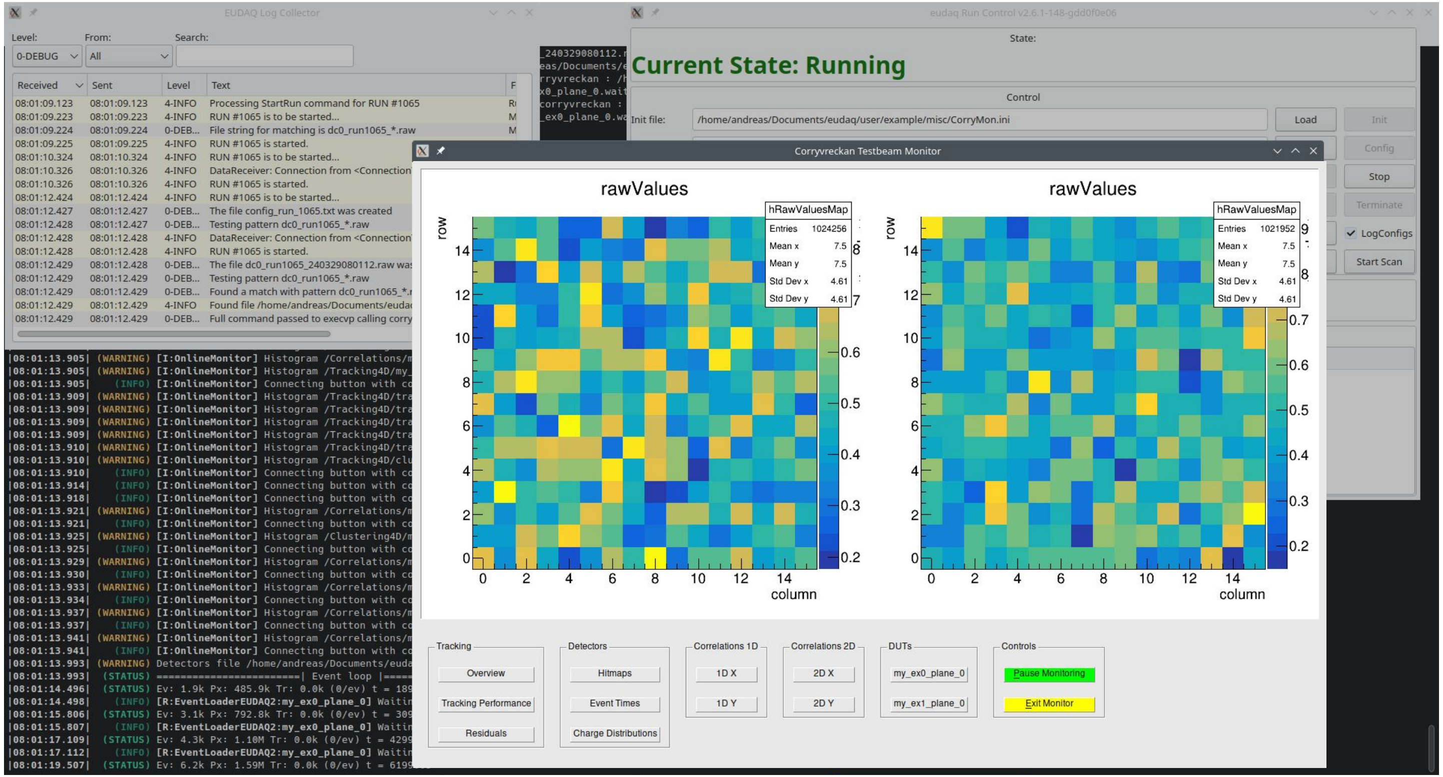
corrygeo.geo



```
[Corryvreckan]
detectors_file = "corrygeo.geo"
detectors_file_updated = "corrygeo_updated.geo"
histogram_file = "corry_histo_file_example.root"

[Metronome]
triggers = 1
event_length = 1s

[EventLoaderEUDAQ2]
type = "Ex0Raw"
eudaq_loglevel=INFO
buffer_depth=5
inclusive=1

[OnlineMonitor]
dut_plots = [["EventLoaderEUDAQ2/%DUT%/hRawValuesMap", "colz"],
             ["EventLoaderEUDAQ2/%DUT%/hPixelRawValues", "log"]]
hitmaps = [["EventLoaderEUDAQ2/%DUT%/hRawValuesMap", "colz"]]
```

```
[my_ex0_plane_0]
number_of_pixels = 16, 16
orientation = 0deg,0deg,0deg
orientation_mode = "xyz"
pixel_pitch = 55um,55um
position = 0um,0um,0um
role = "dut","reference"
time_resolution = -1ns
type = "ex0raw"

[my_ex0_plane_1]
number_of_pixels = 16, 16
orientation = 0deg,0deg,0deg
orientation_mode = "xyz"
pixel_pitch = 55um,55um
position = 0um,0um, 50m
role = "dut"
time_resolution = -1ns
type = "ex0raw"
```

CorryMonitor “in Action”



EUDAQ Log Collector

Level: 0-DEBUG From: All Search:

Received	Sent	Level	Text	F
08:01:09.123	08:01:09.123	4-INFO	Processing StartRun command for RUN #1065	Ri
08:01:09.223	08:01:09.223	4-INFO	RUN #1065 is to be started...	M
08:01:09.224	08:01:09.224	0-DEB...	File string for matching is dc0_run1065_*.raw	M
08:01:09.225	08:01:09.225	4-INFO	RUN #1065 is started.	
08:01:10.324	08:01:10.324	4-INFO	RUN #1065 is to be started...	
08:01:10.326	08:01:10.326	4-INFO	DataReceiver: Connection from <Connection	
08:01:10.326	08:01:10.326	4-INFO	RUN #1065 is started.	
08:01:12.424	08:01:12.424	4-INFO	RUN #1065 is to be started...	
08:01:12.427	08:01:12.427	0-DEB...	The file config_run_1065.txt was created	
08:01:12.428	08:01:12.427	0-DEB...	Testing pattern dc0_run1065_*.raw	
08:01:12.428	08:01:12.428	4-INFO	DataReceiver: Connection from <Connection	
08:01:12.428	08:01:12.428	4-INFO	RUN #1065 is started.	
08:01:12.429	08:01:12.428	0-DEB...	The file dc0_run1065_240329080112.raw was	
08:01:12.429	08:01:12.429	0-DEB...	Testing pattern dc0_run1065_*.raw	
08:01:12.429	08:01:12.429	0-DEB...	Found a match with pattern dc0_run1065_*.r	
08:01:12.429	08:01:12.429	4-INFO	Found file /home/andreas/Documents/eudac	
08:01:12.429	08:01:12.429	0-DEB...	Full command passed to execvp calling corry	

```
_240329080112.r
eas/Documents/e
rryvreckan : /h
x0_plane_0.wait
corryvreckan :
_ex0_plane_0.wa
```

```
|08:01:13.905| (WARNING) [I:OnlineMonitor] Histogram /Correlations/m
|08:01:13.905| (WARNING) [I:OnlineMonitor] Histogram /Tracking4D/my_
|08:01:13.905|    (INFO) [I:OnlineMonitor] Connecting button with co
|08:01:13.909| (WARNING) [I:OnlineMonitor] Histogram /Tracking4D/tra
|08:01:13.909| (WARNING) [I:OnlineMonitor] Histogram /Tracking4D/tra
|08:01:13.909| (WARNING) [I:OnlineMonitor] Histogram /Tracking4D/tra
|08:01:13.909| (WARNING) [I:OnlineMonitor] Histogram /Tracking4D/tra
|08:01:13.910| (WARNING) [I:OnlineMonitor] Histogram /Tracking4D/tra
|08:01:13.910| (WARNING) [I:OnlineMonitor] Histogram /Tracking4D/clu
|08:01:13.910|    (INFO) [I:OnlineMonitor] Connecting button with co
|08:01:13.914|    (INFO) [I:OnlineMonitor] Connecting button with co
|08:01:13.918|    (INFO) [I:OnlineMonitor] Connecting button with co
|08:01:13.921| (WARNING) [I:OnlineMonitor] Histogram /Correlations/m
|08:01:13.921|    (INFO) [I:OnlineMonitor] Connecting button with co
|08:01:13.925| (WARNING) [I:OnlineMonitor] Histogram /Clustering4D/m
|08:01:13.925|    (INFO) [I:OnlineMonitor] Connecting button with co
|08:01:13.929| (WARNING) [I:OnlineMonitor] Histogram /Correlations/m
|08:01:13.930|    (INFO) [I:OnlineMonitor] Connecting button with co
|08:01:13.933| (WARNING) [I:OnlineMonitor] Histogram /Correlations/m
|08:01:13.934|    (INFO) [I:OnlineMonitor] Connecting button with co
|08:01:13.937| (WARNING) [I:OnlineMonitor] Histogram /Correlations/m
|08:01:13.937|    (INFO) [I:OnlineMonitor] Connecting button with co
|08:01:13.941| (WARNING) [I:OnlineMonitor] Histogram /Correlations/m
|08:01:13.941|    (INFO) [I:OnlineMonitor] Connecting button with co
|08:01:13.993| (WARNING) Detectors file /home/andreas/Documents/euda
|08:01:13.993|  (STATUS) ========================| Event loop |=====
|08:01:14.496|  (STATUS) Ev: 1.9k Px: 485.9k Tr: 0.0k (0/ev) t = 189
|08:01:14.498|    (INFO) [R:EventLoaderEUDAQ2:my_ex0_plane_0] Waitin
|08:01:15.806|  (STATUS) Ev: 3.1k Px: 792.8k Tr: 0.0k (0/ev) t = 309
|08:01:15.807|    (INFO) [R:EventLoaderEUDAQ2:my_ex0_plane_0] Waitin
|08:01:17.109|  (STATUS) Ev: 4.3k Px: 1.10M Tr: 0.0k (0/ev) t = 4299
|08:01:17.112|    (INFO) [R:EventLoaderEUDAQ2:my_ex0_plane_0] Waitin
|08:01:19.507|  (STATUS) Ev: 6.2k Px: 1.59M Tr: 0.0k (0/ev) t = 6199
```

eudaq Run Control v2.6.1-148-gdd0f0e06

State:

Current State: Running

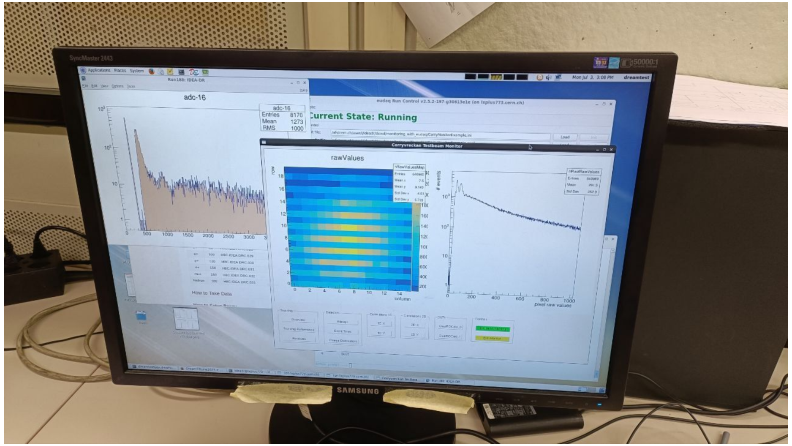
Control

Init file: /home/andreas/Documents/eudaq/user/example/misc/CorryMon.ini

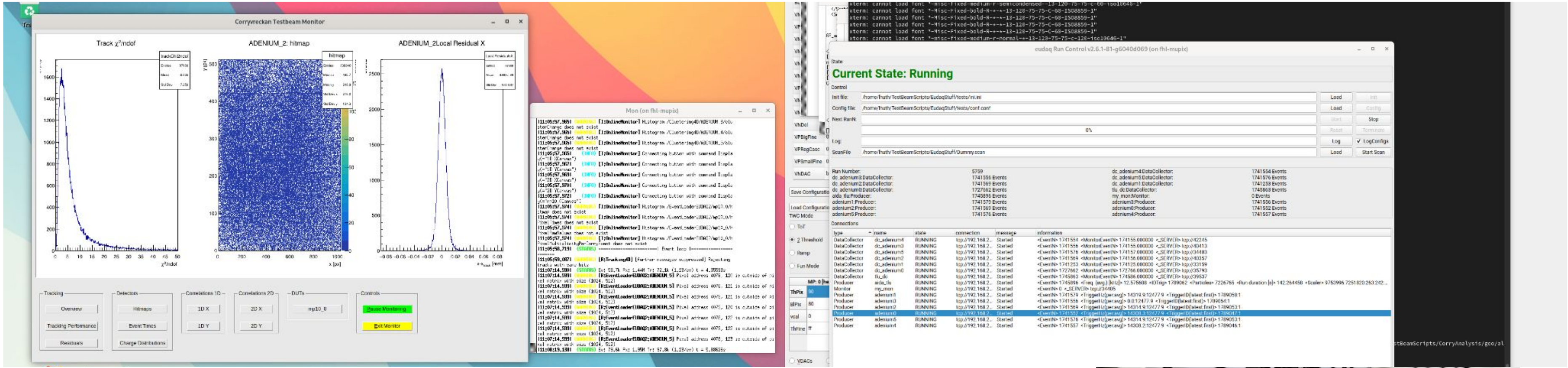
Tests with actual data

- Dual-readout EM prototype
  - 160 Scintillation, 160 Cherenkov fibres individually read out by SiPMs
  - No longitudinal segmentation, i.e. no “layers”
  - But could define Scintillation and Cherenkov channels as own layers or high gain and low gain

courtesy: INFN-PV Lab



TelePix2 + ALPIDE Telescope Monitoring



```
|11:05:57.965| (WARNING) [I:OnlineMonitor] Histogram /Clustering4D/ADENIUM_3/clusterCharge does not exist
|11:05:57.965| (WARNING) [I:OnlineMonitor] Histogram /Clustering4D/ADENIUM_4/clusterCharge does not exist
|11:05:57.965| (WARNING) [I:OnlineMonitor] Histogram /Clustering4D/ADENIUM_5/clusterCharge does not exist
|11:05:57.965|    (INFO) [I:OnlineMonitor] Connecting button with command Display(="1D XCanvas")
|11:05:57.967|    (INFO) [I:OnlineMonitor] Connecting button with command Display(="1D YCanvas")
|11:05:57.969|    (INFO) [I:OnlineMonitor] Connecting button with command Display(="2D XCanvas")
|11:05:57.970|    (INFO) [I:OnlineMonitor] Connecting button with command Display(="2D YCanvas")
|11:05:57.972|    (INFO) [I:OnlineMonitor] Connecting button with command Display(="mp10_0Canvas")
|11:05:57.974| (WARNING) [I:OnlineMonitor] Histogram /EventLoaderEUDAQ2/mp10_0/hitmap does not exist
|11:05:57.974| (WARNING) [I:OnlineMonitor] Histogram /EventLoaderEUDAQ2/mp10_0/hPixelTimes does not exist
|11:05:57.974| (WARNING) [I:OnlineMonitor] Histogram /EventLoaderEUDAQ2/mp10_0/hPixelRawValues does not exist
|11:05:57.974| (WARNING) [I:OnlineMonitor] Histogram /EventLoaderEUDAQ2/mp10_0/hPixelMultiplicityPerCorryEvent does not exist
|11:05:58.719|  (STATUS) ======================| Event loop |=======================
|11:05:59.087| (WARNING) [R:Tracking4D] [further messages suppressed] Rejecting tracks with same hits
|11:07:14.590|  (STATUS) Ev: 58.7k Px: 1.44M Tr: 72.1k (1.23/ev) t = 4.37593s
|11:07:14.599| (WARNING) [R:EventLoaderEUDAQ2:ADENIUM_5] Pixel address 4078, 120 is outside of pixel matrix with size (1024, 512)
|11:07:14.599| (WARNING) [R:EventLoaderEUDAQ2:ADENIUM_5] Pixel address 4078, 121 is outside of pixel matrix with size (1024, 512)
|11:07:14.599| (WARNING) [R:EventLoaderEUDAQ2:ADENIUM_5] Pixel address 4078, 121 is outside of pixel matrix with size (1024, 512)
|11:07:14.599| (WARNING) [R:EventLoaderEUDAQ2:ADENIUM_5] Pixel address 4078, 120 is outside of pixel matrix with size (1024, 512)
|11:07:14.599| (WARNING) [R:EventLoaderEUDAQ2:ADENIUM_5] Pixel address 4078, 122 is outside of pixel matrix with size (1024, 512)
|11:07:14.599| (WARNING) [R:EventLoaderEUDAQ2:ADENIUM_5] Pixel address 4078, 122 is outside of pixel matrix with size (1024, 512)
|11:07:14.599| (WARNING) [R:EventLoaderEUDAQ2:ADENIUM_5] Pixel address 4078, 123 is outside of pixel matrix with size (1024, 512)
|11:08:19.138|  (STATUS) Ev: 79.6k Px: 1.95M Tr: 97.8k (1.23/ev) t = 5.88626s
```

```
xterm: cannot load font "-misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso10646-1"
xterm: cannot load font "-Misc-Fixed-bold-R-*-*-13-120-75-75-C-60-ISO8859-1"
xterm: cannot load font "-Misc-Fixed-bold-R-*-*-13-120-75-75-C-60-ISO8859-1"
xterm: cannot load font "-Misc-Fixed-bold-R-*-*-13-120-75-75-C-60-ISO8859-1"
xterm: cannot load font "-Misc-Fixed-bold-R-*-*-13-120-75-75-C-60-ISO8859-1"
xterm: cannot load font "-misc-fixed-medium-r-normal-*-13-120-75-75-c-120-iso10646-1"
```

eudaq Run Control v2.6.1-81-g6040d069 (on fhl-mupix)

State

Current State: Running

Control

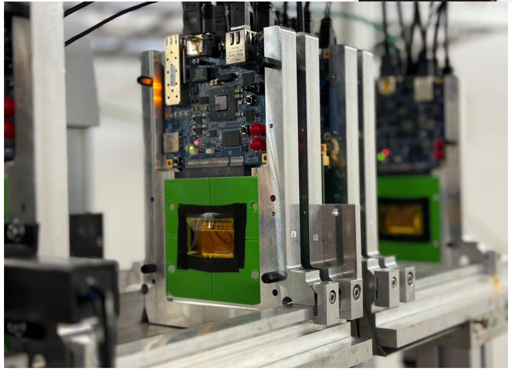
Init file:	/home/lhuth/TestBeamScripts/EudaqStuff/tests/ini.ini	Load	Init
Config file:	/home/lhuth/TestBeamScripts/EudaqStuff/tests/conf.conf	Load	Config
Next RunN:		Start	Stop
	0%	Reset	Terminate
Log:		Log	✓ LogConfigs
ScanFile	/home/lhuth/TestBeamScripts/EudaqStuff/Dummy.scan	Load	Start Scan

Run Number:	5789	dc_adenium4:DataCollector:	1741554 Events
dc_adenium3:DataCollector:	1741556 Events	dc_adenium5:DataCollector:	1741576 Events
dc_adenium2:DataCollector:	1741569 Events	dc_adenium1:DataCollector:	1741253 Events
dc_adenium0:DataCollector:	1727662 Events	tlu_dc:DataCollector:	1745863 Events
aida_tlu:Producer:	1745896 Events	my_mon:Monitor:	0 Events
adenium1:Producer:	1741579 Events	adenium3:Producer:	1741556 Events
adenium2:Producer:	1741569 Events	adenium0:Producer:	1741552 Events
adenium5:Producer:	1741576 Events	adenium4:Producer:	1741557 Events

Connections

type	name	state	connection	message	information
DataCollector	dc_adenium4	RUNNING	tcp://192.168.2...	Started	<EventN> 1741554 <MonitorEventN> 174155.000000 <_SERVER> tcp://42245
DataCollector	dc_adenium3	RUNNING	tcp://192.168.2...	Started	<EventN> 1741556 <MonitorEventN> 174155.000000 <_SERVER> tcp://40413
DataCollector	dc_adenium5	RUNNING	tcp://192.168.2...	Started	<EventN> 1741576 <MonitorEventN> 174157.000000 <_SERVER> tcp://34403
DataCollector	dc_adenium2	RUNNING	tcp://192.168.2...	Started	<EventN> 1741569 <MonitorEventN> 174156.000000 <_SERVER> tcp://40357
DataCollector	dc_adenium1	RUNNING	tcp://192.168.2...	Started	<EventN> 1741253 <MonitorEventN> 174125.000000 <_SERVER> tcp://33159
DataCollector	dc_adenium0	RUNNING	tcp://192.168.2...	Started	<EventN> 1727662 <MonitorEventN> 172766.000000 <_SERVER> tcp://35793
DataCollector	tlu_dc	RUNNING	tcp://192.168.2...	Started	<EventN> 1745863 <MonitorEventN> 174586.000000 <_SERVER> tcp://39537
Producer	aida_tlu	RUNNING	tcp://192.168.2...	Started	<EventN> 1745896 <Freq. (avg.) [kHz]> 12.575608 <IDTrig> 1789062 <Particles> 7226765 <Run duration [s]> 142.264458 <Scaler> 9753996.7251820.263.242...
Monitor	my_mon	RUNNING	tcp://192.168.2...	Started	<EventN> 0 <_SERVER> tcp://34405
Producer	adenium1	RUNNING	tcp://192.168.2...	Started	<EventN> 1741579 <TriggerHz(per:avg)> 14319.9:12477.9 <TriggerID(latest:first)> 1789058:1
Producer	adenium3	RUNNING	tcp://192.168.2...	Started	<EventN> 1741556 <TriggerHz(per:avg)> 0.0:12477.9 <TriggerID(latest:first)> 1789054:1
Producer	adenium2	RUNNING	tcp://192.168.2...	Started	<EventN> 1741569 <TriggerHz(per:avg)> 14314.9:12477.9 <TriggerID(latest:first)> 1789053:1
Producer	adenium0	RUNNING	tcp://192.168.2...	Started	<EventN> 1741552 <TriggerHz(per:avg)> 14308.3:12477.9 <TriggerID(latest:first)> 1789047:1
Producer	adenium5	RUNNING	tcp://192.168.2...	Started	<EventN> 1741576 <TriggerHz(per:avg)> 14314.9:12477.9 <TriggerID(latest:first)> 1789053:1
Producer	adenium4	RUNNING	tcp://192.168.2...	Started	<EventN> 1741557 <TriggerHz(per:avg)> 14308.2:12477.9 <TriggerID(latest:first)> 1789046:1

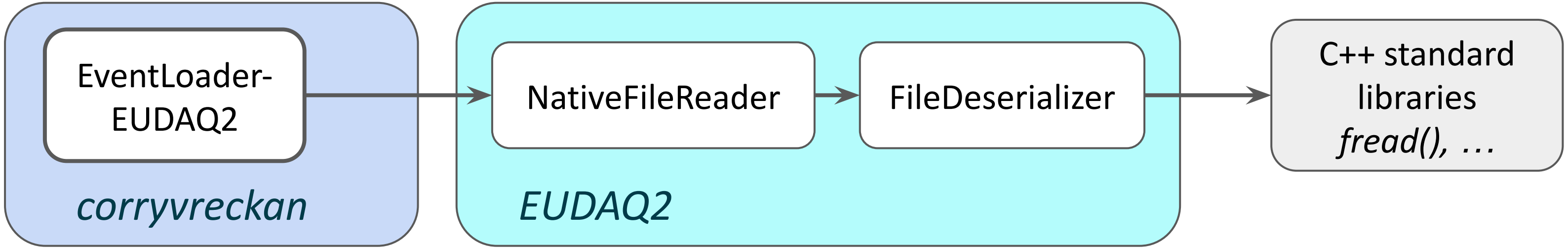
courtesy: Lennart Huth



- TelePix2 beam test with ALPIDE telescope at DESY by Lennart Huth et. al.
- Beam telescope allows to fully exploit corryvreckan functionality (correlations, tracking)

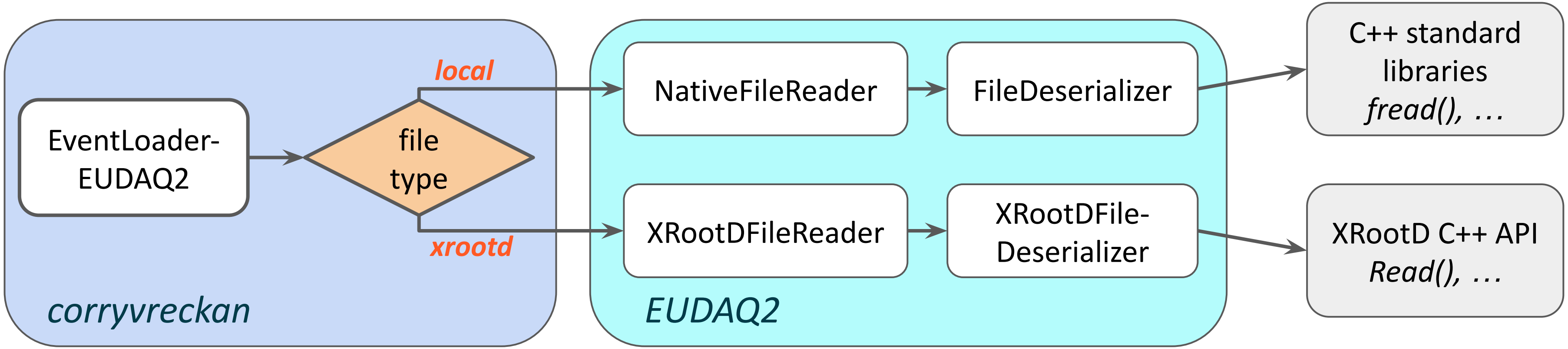
More Features: XRootD

- Want to be able to **monitor DataCollectors on different machine**
- Copying files over is not an option
- Need quick way to establish connection
- XRootD software offers solution
  - Allows fast, low latency and scalable data access
- Need to make EUDAQ2 and corryvreckan compatible with XRootD



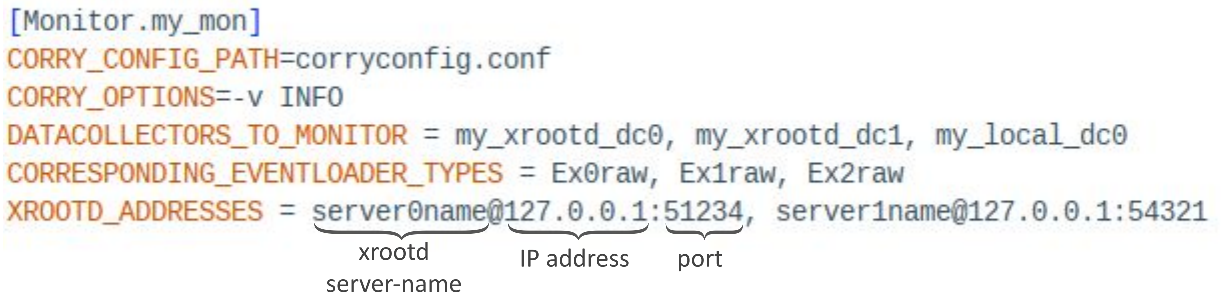
More Features: XRootD

- Want to be able to **monitor DataCollectors on different machine**
- Copying files over is not an option
- Need quick way to establish connection
- XRootD software offers solution
  - Allows fast, low latency and scalable data access
- Need to make EUDAQ2 and corryvreckan compatible with XRootD



More Features: XRootD

- User needs to start XRootD server/ensure connection
- In the .conf file pass server addresses in order (remote DataCollectors first, then local)



```
[Monitor.my_mon]
CORRY_CONFIG_PATH=corryconfig.conf
CORRY_OPTIONS=-v INFO
DATACOLLECTORS_TO_MONITOR = my_xrootd_dc0, my_xrootd_dc1, my_local_dc0
CORRESPONDING_EVENTLOADER_TYPES = Ex0raw, Ex1raw, Ex2raw
XROOTD_ADDRESSES = server0name@127.0.0.1:51234, server1name@127.0.0.1:54321
```

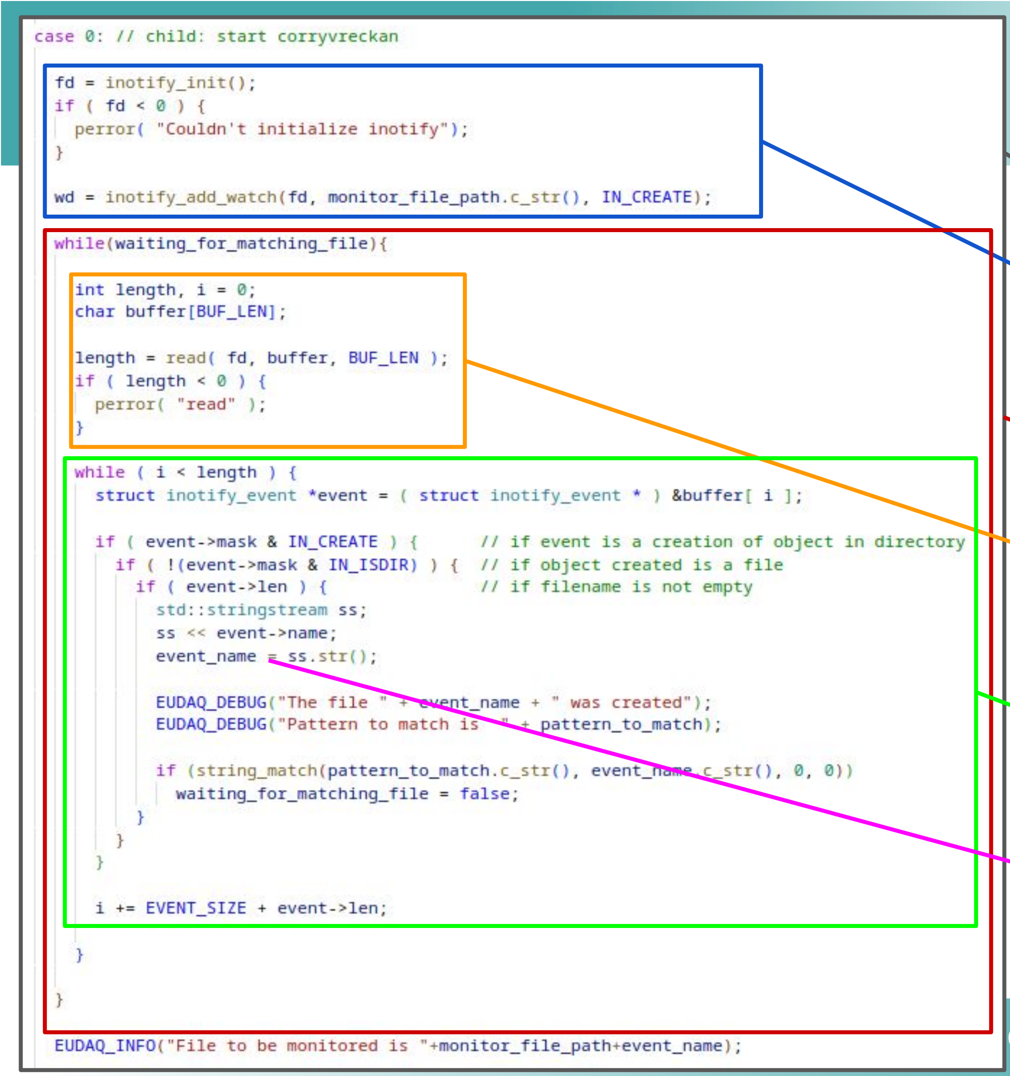
- Tested in lab setting
- Unfortunately no opportunity to test in test beam setting so far

Summary

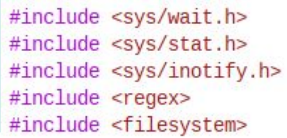
- Goal: versatile online monitoring tool usable by any test beam users
- Corryvreckan provides underlying functionality
    - Much comes for free
- Well integrated into EUDAQ2
- Have had / will have some testing opportunities
    - Reading files on different machines at test beam would be nice
- Planning to release into eudaq (and corryvreckan) repositories soon

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101004761.

Thank you for your Attention!



```
case 0: // child: start corryvreckan

  fd = inotify_init();
  if ( fd < 0 ) {
    perror( "Couldn't initialize inotify");
  }

  wd = inotify_add_watch(fd, monitor_file_path.c_str(), IN_CREATE);

  while(waiting_for_matching_file){

    int length, i = 0;
    char buffer[BUF_LEN];

    length = read( fd, buffer, BUF_LEN );
    if ( length < 0 ) {
      perror( "read" );
    }

    while ( i < length ) {
      struct inotify_event *event = ( struct inotify_event * ) &buffer[ i ];

      if ( event->mask & IN_CREATE ) {       // if event is a creation of object in directory
        if ( !(event->mask & IN_ISDIR) ) {  // if object created is a file
          if ( event->len ) {               // if filename is not empty
            std::stringstream ss;
            ss << event->name;
            event_name = ss.str();

            EUDAQ_DEBUG("The file " + event_name + " was created");
            EUDAQ_DEBUG("Pattern to match is " + pattern_to_match);

            if (string_match(pattern_to_match.c_str(), event_name.c_str(), 0, 0))
              waiting_for_matching_file = false;
          }
        }
      }

      i += EVENT_SIZE + event->len;

    }

  }
  EUDAQ_INFO("File to be monitored is "+monitor_file_path+event_name);
```



```
#include <sys/wait.h>
#include <sys/stat.h>
#include <sys/inotify.h>
#include <regex>
#include <filesystem>
```

Fork to start corry

Setting up inotify

While waiting for the correct file to be created

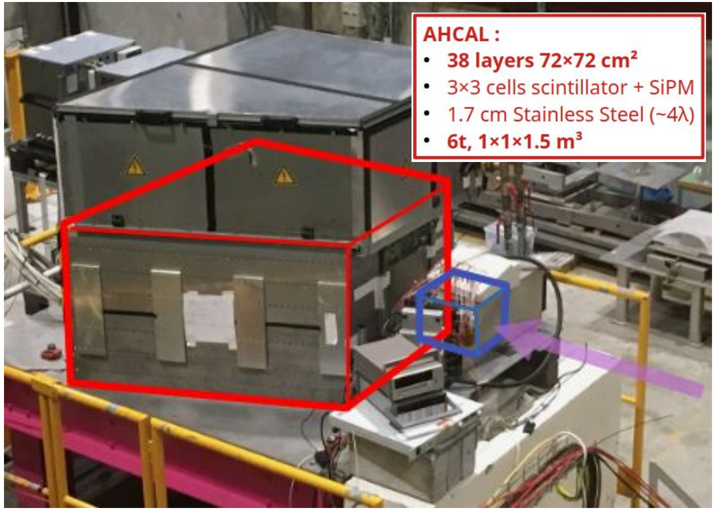
Read the "event" (change in directory)

Check of any of the events read in this instance are the creation of the desired file

Variable storing file name

AHCAL Monitoring

- After last year’s meeting: approached by Jiří Kvasnička
  - Offered CALICE AHCAL **test beam data** to test monitoring with
  - Provided with AHCAL Reader to emulate data taking

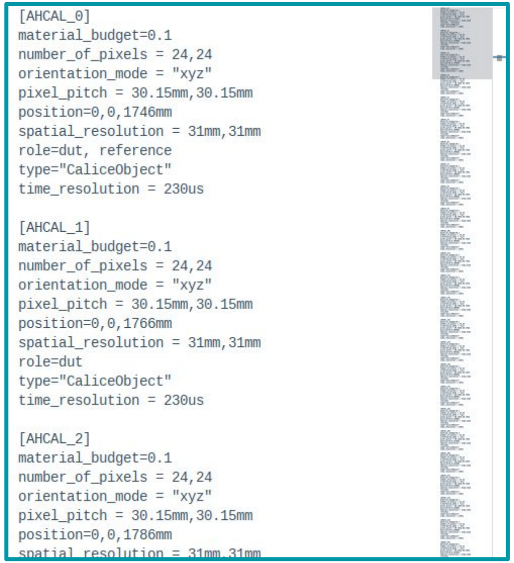


AHCAL :

- **38 layers 72×72 cm²**
- 3×3 cells scintillator + SiPM
- 1.7 cm Stainless Steel (~4λ)
- **6t, 1×1×1.5 m³**

courtesy: Vincent Boudry

corrygeo.geo

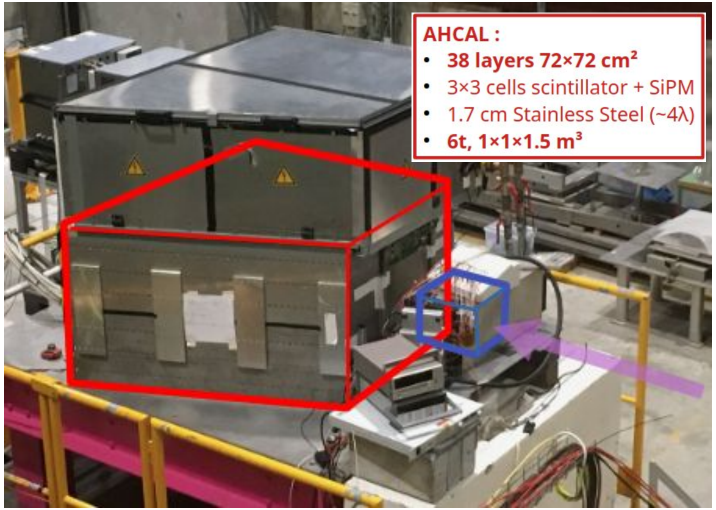


```
[AHCAL_0]
material_budget=0.1
number_of_pixels = 24,24
orientation_mode = "xyz"
pixel_pitch = 30.15mm,30.15mm
position=0,0,1746mm
spatial_resolution = 31mm,31mm
role=dut, reference
type="CaliceObject"
time_resolution = 230us

[AHCAL_1]
material_budget=0.1
number_of_pixels = 24,24
orientation_mode = "xyz"
pixel_pitch = 30.15mm,30.15mm
position=0,0,1766mm
spatial_resolution = 31mm,31mm
role=dut
type="CaliceObject"
time_resolution = 230us

[AHCAL_2]
material_budget=0.1
number_of_pixels = 24,24
orientation_mode = "xyz"
pixel_pitch = 30.15mm,30.15mm
position=0,0,1786mm
spatial_resolution = 31mm,31mm
```

AHCAL Monitoring

- After last year’s meeting: approached by Jiří Kvasnička
  - Offered CALICE AHCAL **test beam data** to test monitoring with
  - Provided with AHCAL Reader to emulate data taking

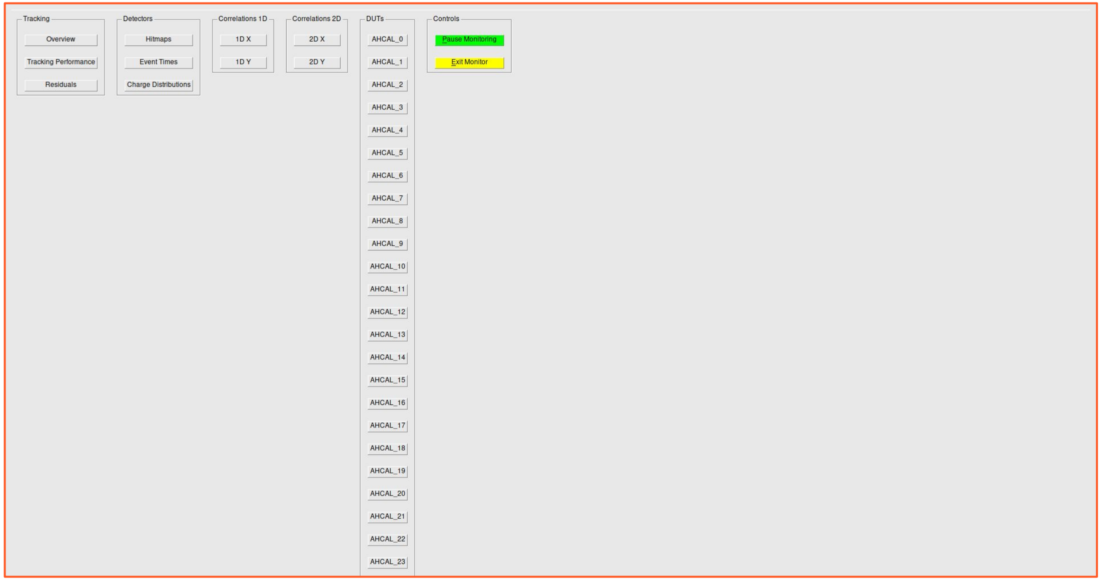
- This is the full window, not enough space for plots
- Not even enough space to fit all layers



AHCAL :

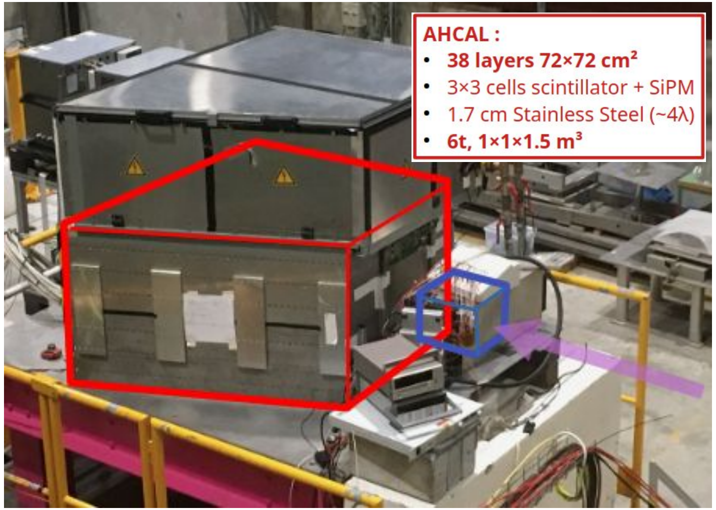
- **38 layers 72×72 cm²**
- 3×3 cells scintillator + SiPM
- 1.7 cm Stainless Steel (~4λ)
- **6t, 1×1×1.5 m³**

courtesy: Vincent Boudry



Tests with actual data

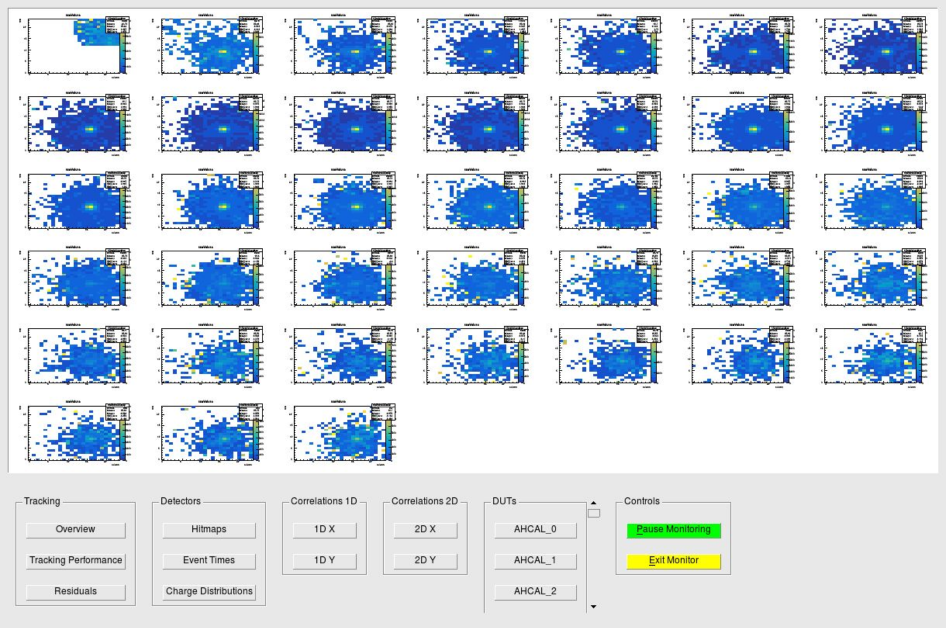
- CALICE AHCAL test beam data offered by Jiří Kvasnička
- Provided with AHCAL Reader to emulate data taking



AHCAL :

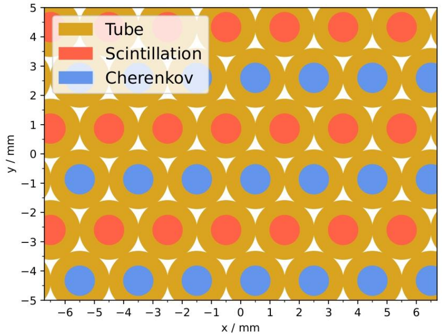
- 38 layers 72×72 cm²
- 3×3 cells scintillator + SiPM
- 1.7 cm Stainless Steel (~4λ)
- 6t, 1×1×1.5 m³

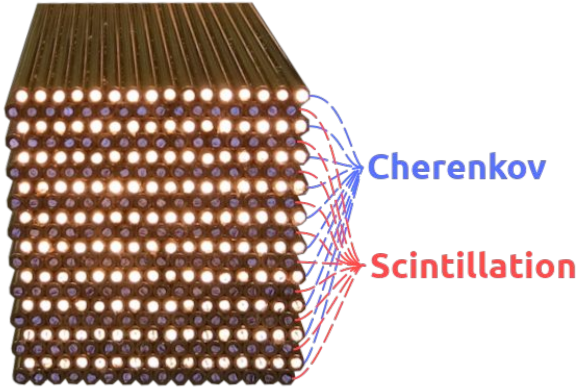
Vincent Boudry, AIDAinnova 2nd Annual Meeting, 25.04.2023



Dual-Readout Monitoring

- Test beam campaign in June/July 2023 at SPS North Area
- Dual-readout EM prototype
  - 160 Scintillation, 160 Cherenkov individually read out by SiPMs
  - No longitudinal segmentation, i.e. no “layers”
  - But could define Scintillation and Cherenkov channels as own layers or high gain and low gain





courtesy: INFN-PV Lab