Pyrame 3

an online framework for Calice SiW-Ecal
Frédéric Magniette on behalf of the Calice collaboration
CHEF Oct. 2017, Lyon
Pyrame in a nutshell

Hardware

Low Level Module

sequence Module

Physics Module

Central Service

Low Level Module

Low Level Module

Acquisition Chain

Scripts

GUI

- Online framework
- Collection of command modules
- Peer to peer asynchronous tcp/ip comm.
- One task per module
- Functional code in Python, C/C++/Root, R, Lua, Shell
- Open-source
- Used on different HEP experiments:
  - T2K wagasci
  - Calice SiW-Ecal
  - Harpo (TPC)
Dozens of hardware modules (drivers)

- Buses: RS232, GPIB, Ethernet, TCP, UDP, USB
- Power Supplies (Agilent, CAEN, Hameg…)
- Pattern Generators (Agilent)
- Motion controllers and probe station (Newport, Thorlabs, Signatone)
- Digital storage oscilloscopes (Lecroy)
- Particle detector chip (Omega)
- GaussMeter (LakeShore)
- Multimeter (Keithley)

It is very easy to write (adapt) a new module for your hardware
Acquisition Chain

- Multi-media
- Multi-format
- Fast: 4Gb/s
- Collect all data from all sources
- Aggregate by stream
- Extract command packets
What's new in Pyrame 3

- New online data handling mechanism
- Online decoders
- Online reconstruction
- Online Monitoring
- Unified configuration system
- Advanced scripting facilities
Dispatcher and Event Loop

- Break real-time constraints
- Fair subsampling
- Data structure publication
- Serve data to any client (multiple access)
- New run reinit mechanism
- Central index of data sources

- Ease online analysis development
- Allow chaining
- Possibility to use multiple event-loops at the same time
- Field value extraction by name function
- Only for C/C++ (ok for root)
Data format

• Basic data bricks are "events" which contains fields splitted in 3 domains
  – Space coordinates
  – Time coordinates
  – Any other data: hit, energy, quality tag…

• Events are grouped in "blocks"
  – A block can have some global properties including a unique id for the run

• Event grouping is up to the online developper
  – Beware : subsampling quality relies heavily on statistical representativity. There should be no bias in event representativity in blocks

• Pyrame can be easily adapted to any format that can embed such a structure (for example LCIO)

• SAF : a simple ascii format
  prop1,prop2!time1,time2...|space1,space2,...|data1,data2...!time1,time2...
Performance

- Event-loop decoding is better if packets are big: network overhead
- Encoding and decoding are linear in data size
- Can be improved by a binary format

On HP Zbook with i5 8 threads processor
Online converters

- A data converter can be launched by data source
- Decode data in real time + do some very basic reconstruction
- Group data by blocks corresponding to time slots
- Feed a dispatcher
Applications for SiW-Ecal : Calicoes
Online Monitoring

- Instanciate event loops connected to all decoders
- make plots from data
- Simple version just make some beam spotting
- Expert version make plots of all errors and malfunction
- Monitor beam in real time

Beam monitoring during beam-test @ DESY
New scripting facilities

- Allow any complex behavior (calibration, data driven reconfiguration...)
- All hardware accessible from script
- Central configuration file (all HW)
- All electronics from detector is reconfigurable on the fly
- Integrate offline analysis, inducing reconfiguration
- Export variables from online treatments, usable in script

Scripts are written in

```
Move table to position 32
Set high-voltage to 100v
Wait 120s
Do
    Acquire data for 300s
    Analyse data
    Output noisy channels
    Mask noisy channels
Until #noisy-channels=0
Start event builder
Acquire data
Do
    Eb export variable #full-events
    Wait 60s
Until #full-events>=1000
Stop acquisition
```
Eudaq integration

- Eudaq and Calicoes now have the same state machine
- Easy command integration for "normal run"
- Calicoes includes already a Eudaq1 producer (migration planned to Eudaq2)
- Data feeding for Eudaq will be done through a multiple event-loops client (not implemented yet)
- Question: how to format data to avoid second specific decoding inside Eudaq?
- What about some "standard format" with a generic decoder provided natively in Eudaq? extended LCIO? compressed JSON? Podio? Specific new binary format?
Summary and perspectives

- Pyrame 3 is an online framework, flexible, stable and performant
- Provides new functionalities in online monitoring and advanced scripting
- Provides advanced support to SiW-Ecal testbeams for years: see A. Irles talk tuesday

- Foreseen developments
  - High level data analysis
  - Optimize data transfert by adopting adapted new format