

DAQ Calice

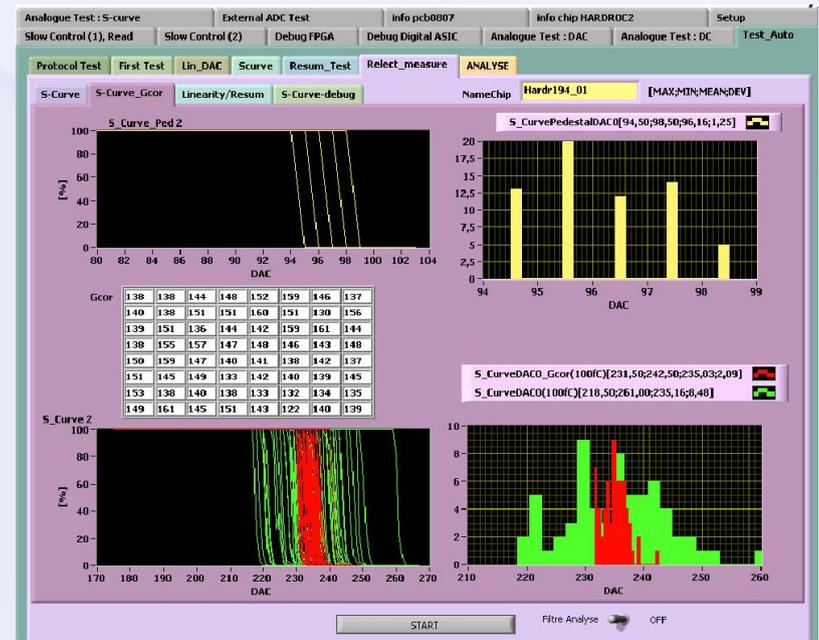
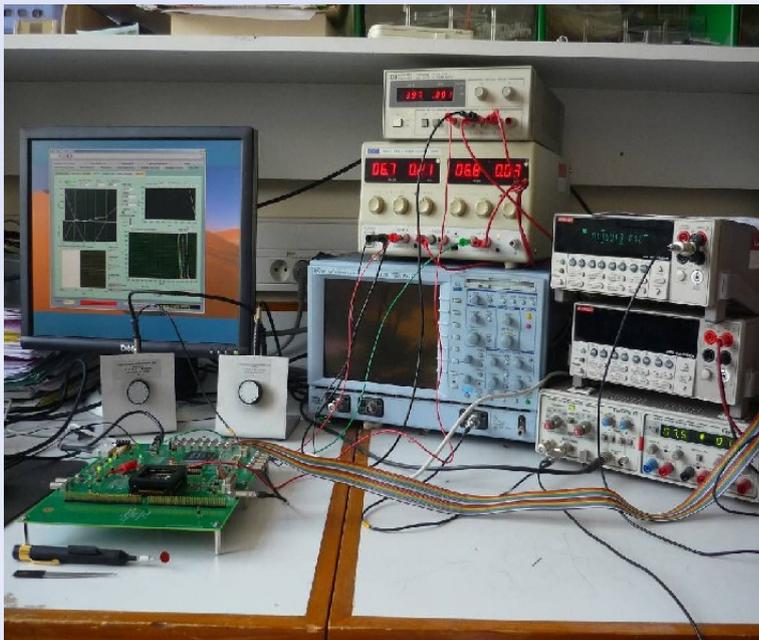
Christophe Combaret (IPNL)
David Decotigny, Vincent Boudry (LLR)



DAQs existantes pour Calice

Bancs de tests

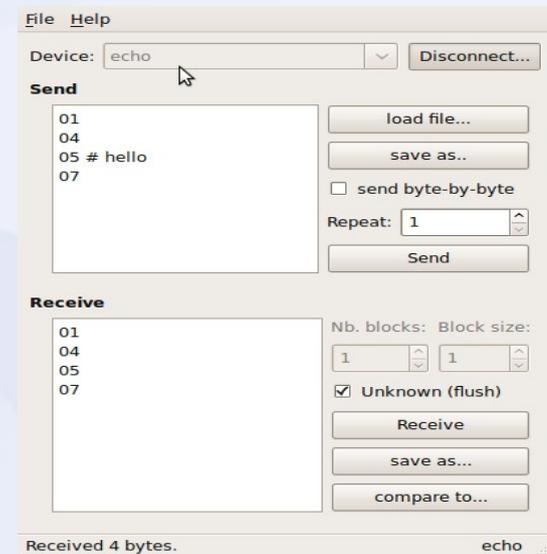
- Bancs de tests chips
 - Labview pour tests ROC @LAL (R. Della Negra/IPNL): tests 5000 chips



DAQs existantes pour Calice

Bancs de tests

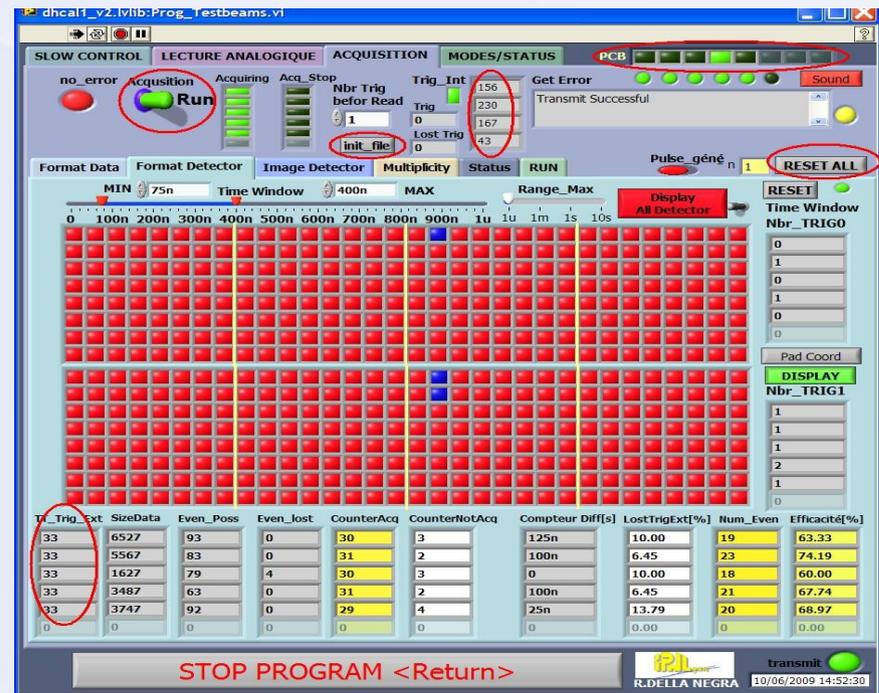
- Banc de test electronique
 - C++/python pour debug Calice DAQ2 @LLR



DAQs existantes pour Calice

Bancs de tests

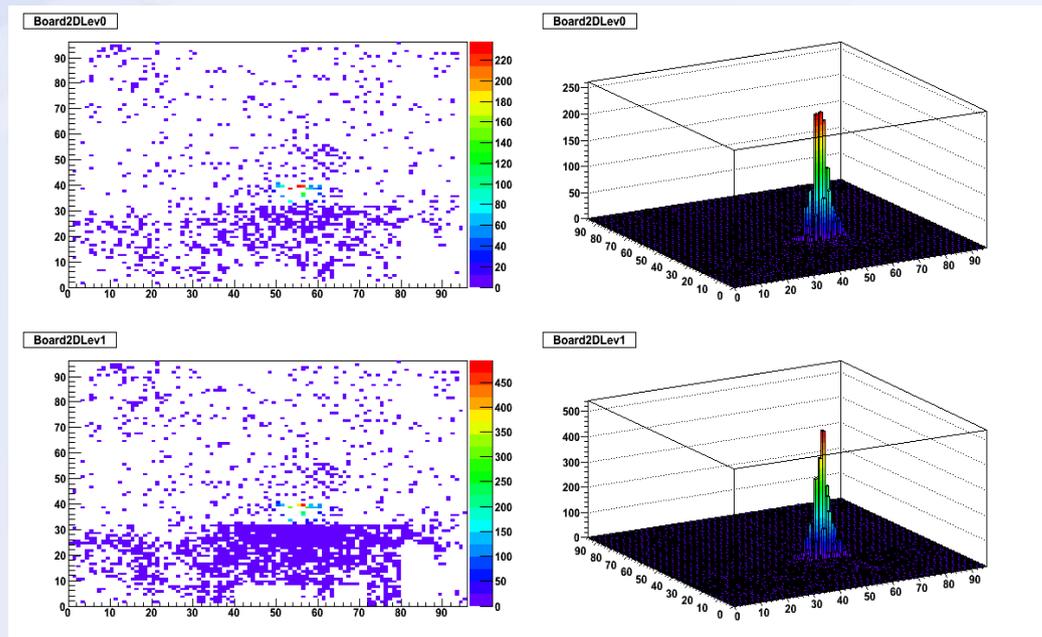
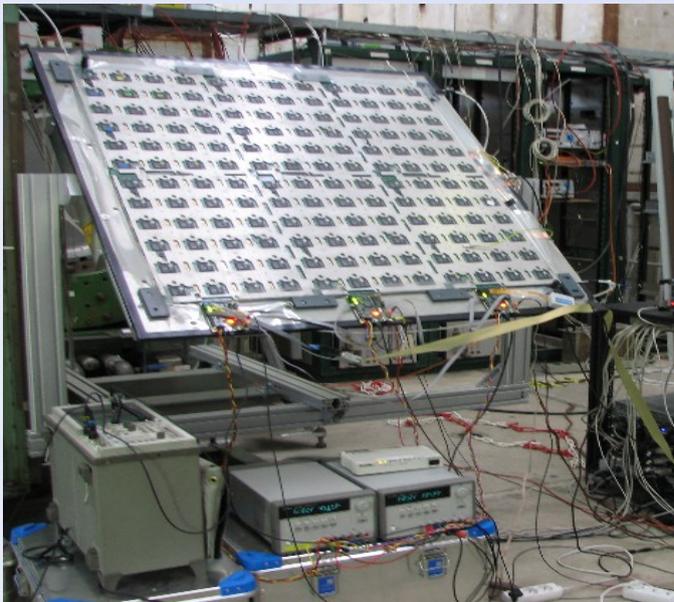
- Banc de test SDHCAL GRPC @IPNL (R. Della Negra)
 - Labview + HardROC + USB



DAQs existantes pour Calice

Bancs de tests

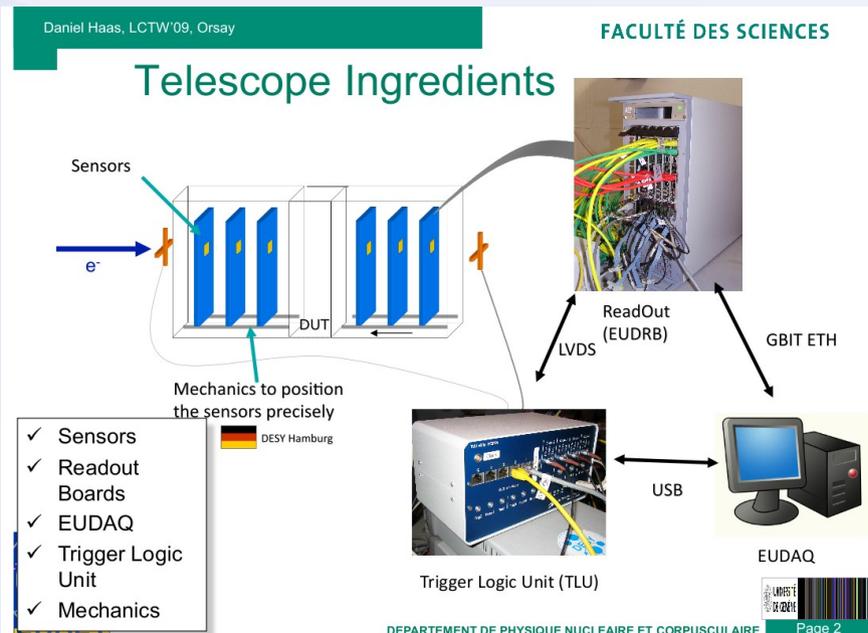
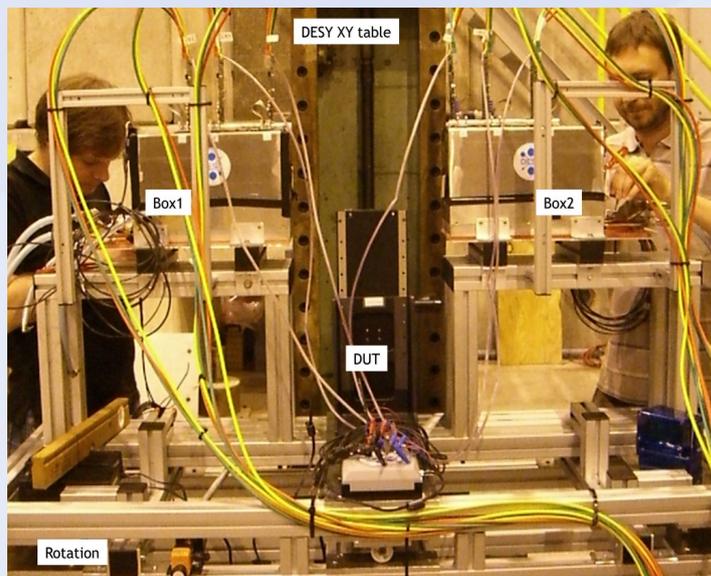
- Banc de test SDHCAL GRPC m2 @IPNL
(C. Combaret / R. Kieffer)
 - Xdaq + HardROC2 + USB + ROOT



DAQs existantes pour Calice

Systemes integres

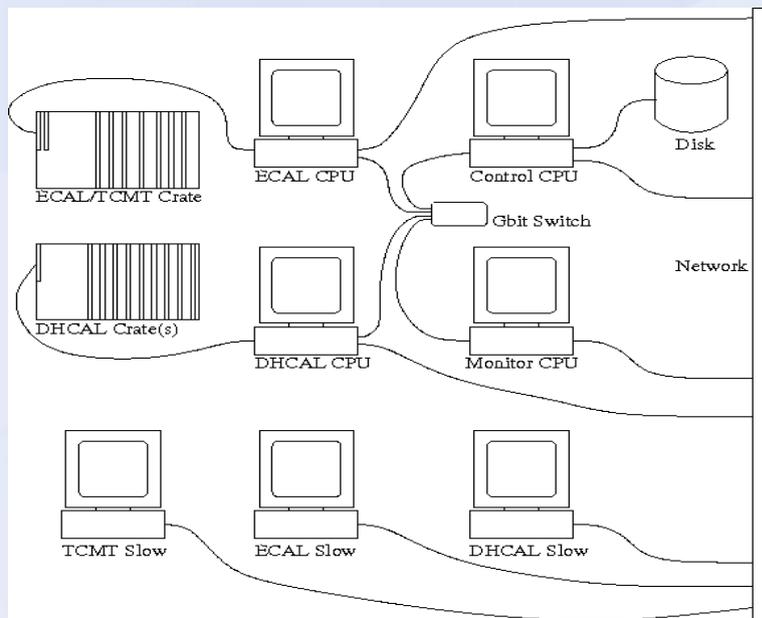
- DAQ Telescope EUDET : “EUDAQ” (Univ. Geneve)
 - C++ multi-plateformes
 - Utilisee par ~20 tests, dont Calice
 - Base commune pour les echanges/stockages de donnees



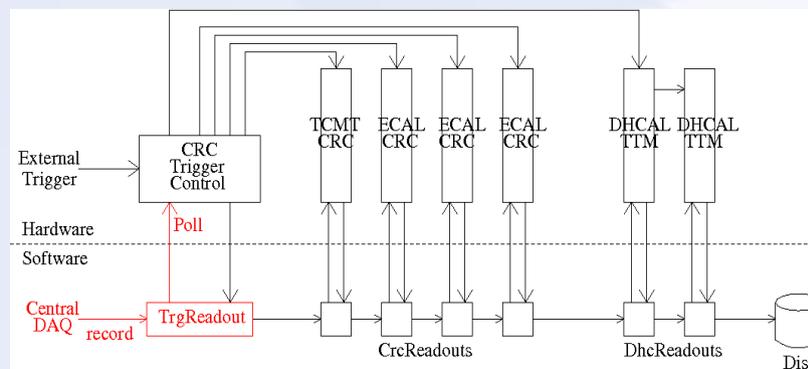
DAQs existantes pour Calice

Systemes integres

- DHCAL “DAQ1” (P. Dauncey @ Imperial)
 - C++ / CERN HAL + VME / Eth
 - Utilisee en combine avec ECAL + TCMT (FNAL)
 - Plus utilisee en France



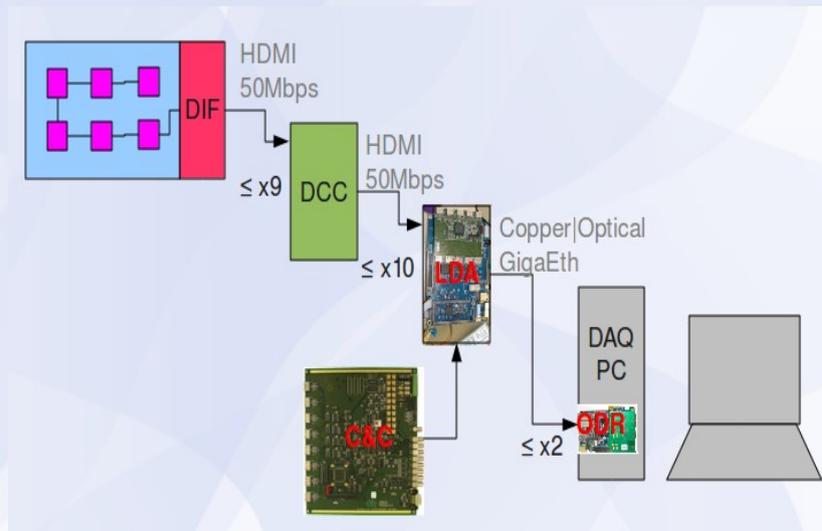
DAQ readout combine
par passage de jeton :



DAQs existantes pour Calice

Systemes integres

- “DAQ2” (UCL/RHUL/Manchester/LLR)
 - Pour DHCAL/ECAL
 - Topologie type reseau (~ Ethernet)
 - C++ + Xdaq ou Tango + EUDAQ



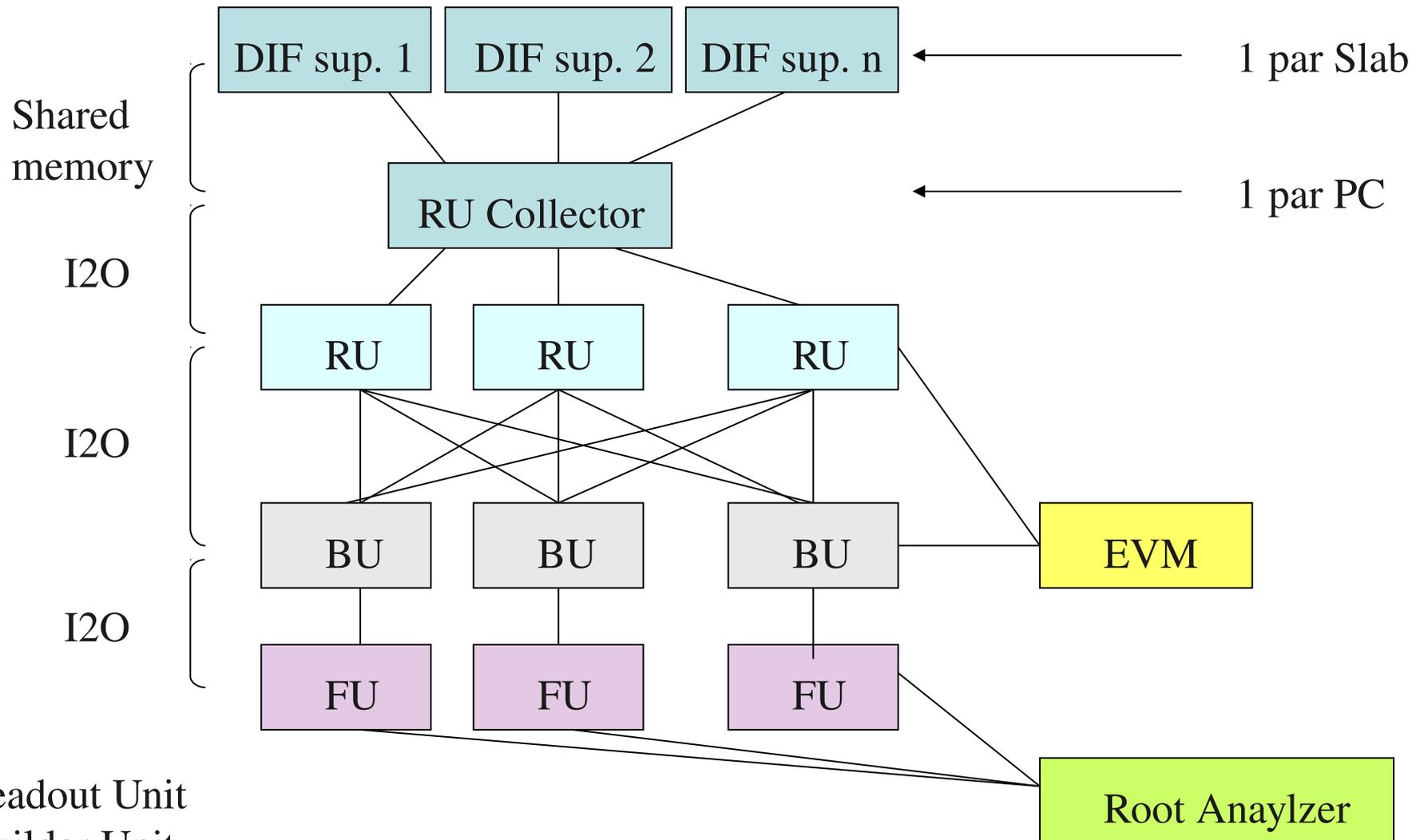
1m2 GRPC Acquisition System with Xdaq

C. Combaret, IPNL

Nouveautés

1. Intégration de l'évent builder natif de XDAQ
(L. Mirabito)
2. Analyse online (L. Mirabito)
3. Lancement de l'acquisition par la machine d'états principale.
4. Tests de la carte Hardroc 2 : Carte fonctionnelle,
Xdaq OK pour Hardroc 2. Calibration en cours.

Intégration de l'event builder natif de Xdaq



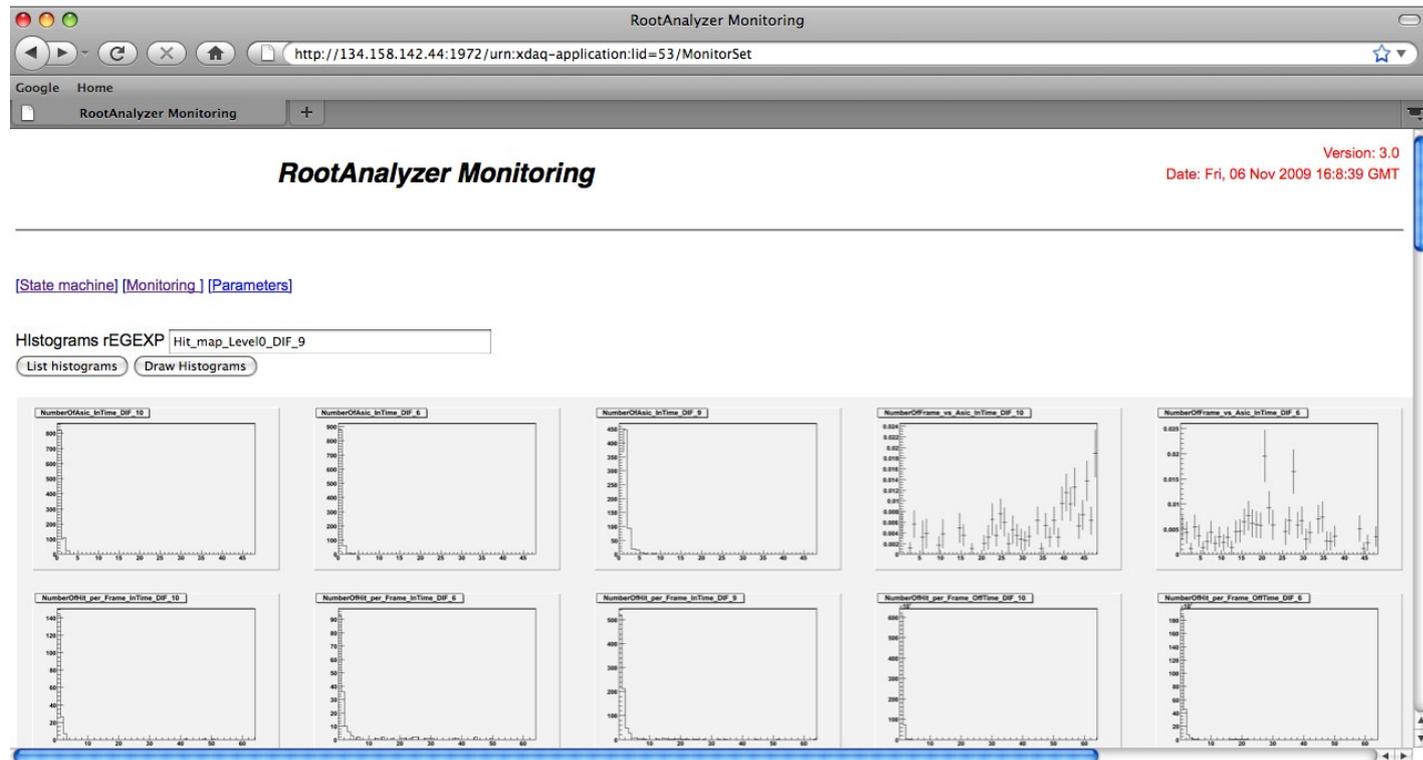
RU: Readout Unit
BU: Builder Unit
FU: Filter Unit

9/11/2009

Analyse root online (L. Mirabito)

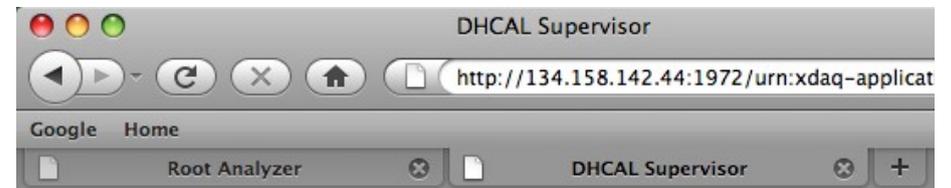
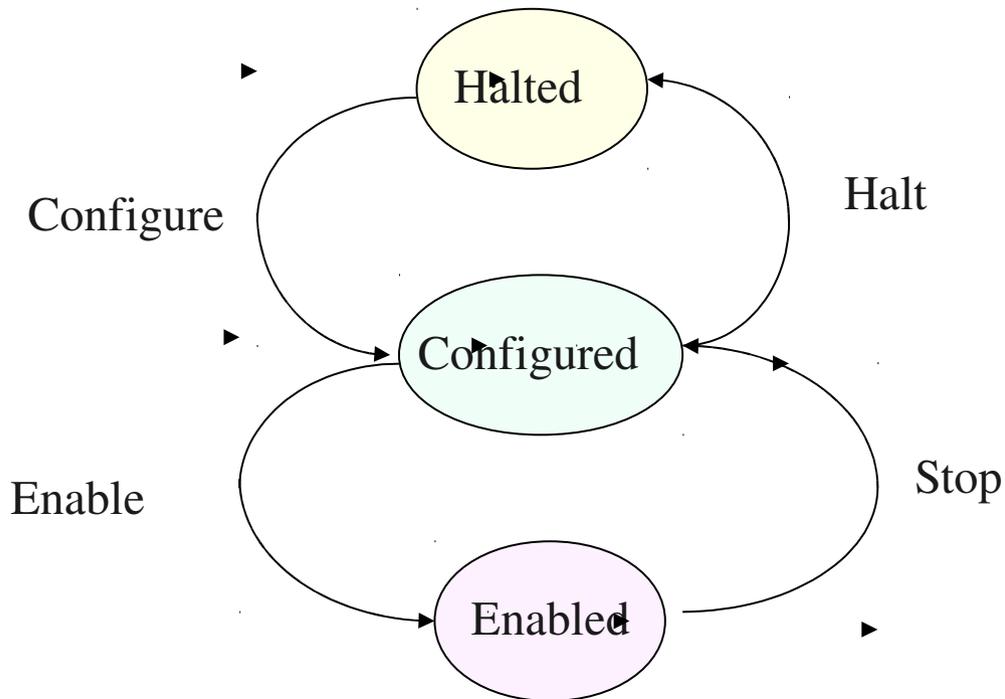
Pour chaque DIF :

NumberOfAsic_InTime	Hit_map_Level0/1
NumberOfFrame_vs_Asic_InTime	Hit_map_OffTime_Level0/1
NumberOfHit_per_Frame_InTime	LastHit_map_Level0/1
NumberOfHit_per_Frame_OffTime	LastHit_map_OffTime_Level0/1
NumberOfHit_vs_Asic_InTim	NumberOfFrame_vs_Asic



Utilisation de la FSM principale

Chaque application Xdaq peut être piloté par la FSM principale. Il suffit que sa propre FSM comporte les états **Halted**, **Configured** et **Enabled** et qu'elle implémente les transitions **Configure**, **Enable**, **Halt** et **Stop**.



Main DHCAL Supervisor

Date: Sun, 08 Nov 2009:

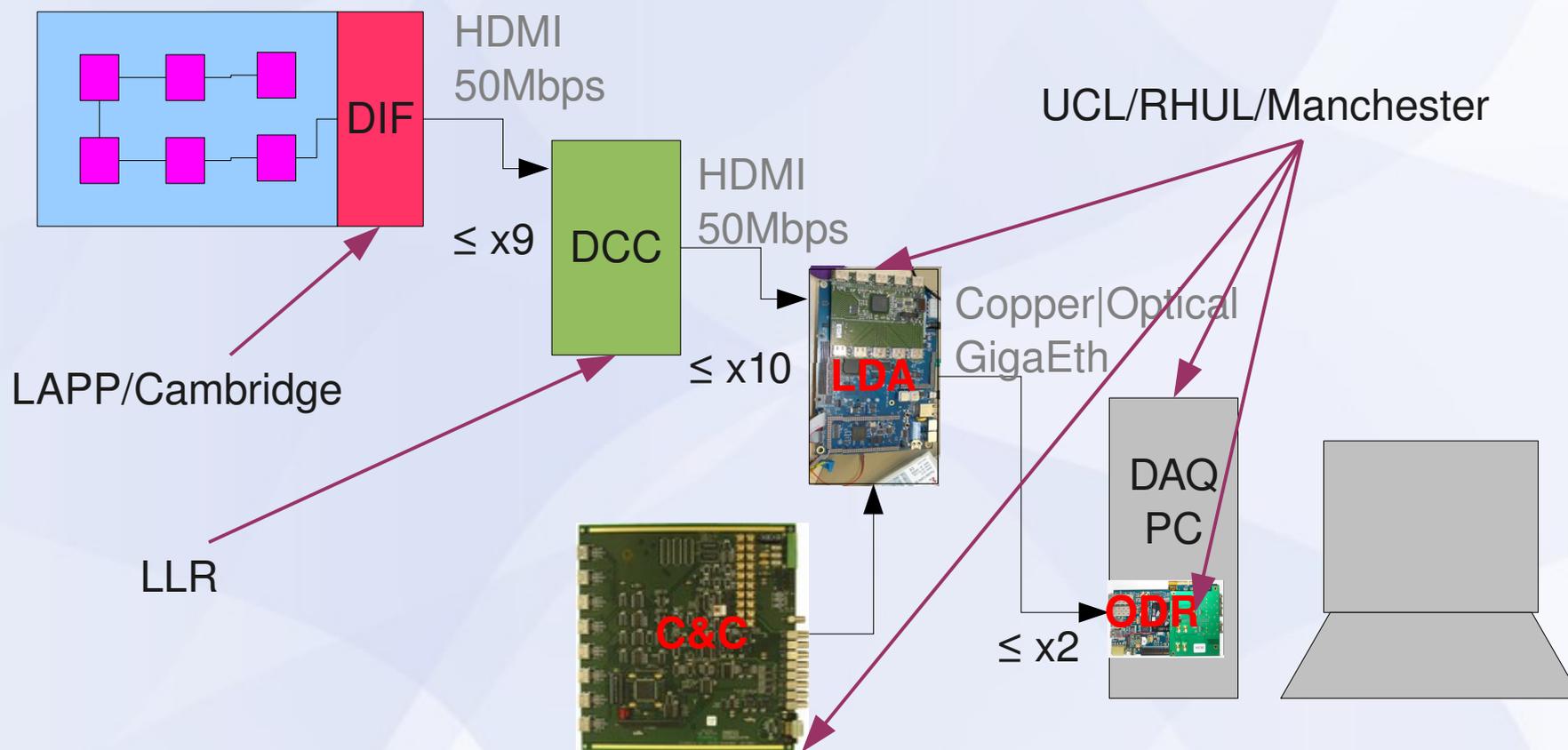
[\[State machine\]](#) [\[Parameters\]](#)

DHCAL Supervisor Finite State Machine

Enabled				
<input type="button" value="Configure"/>	<input type="button" value="Enable"/>	<input type="button" value="Halt"/>	<input type="button" value="Initialise"/>	<input type="button" value="Stop"/>

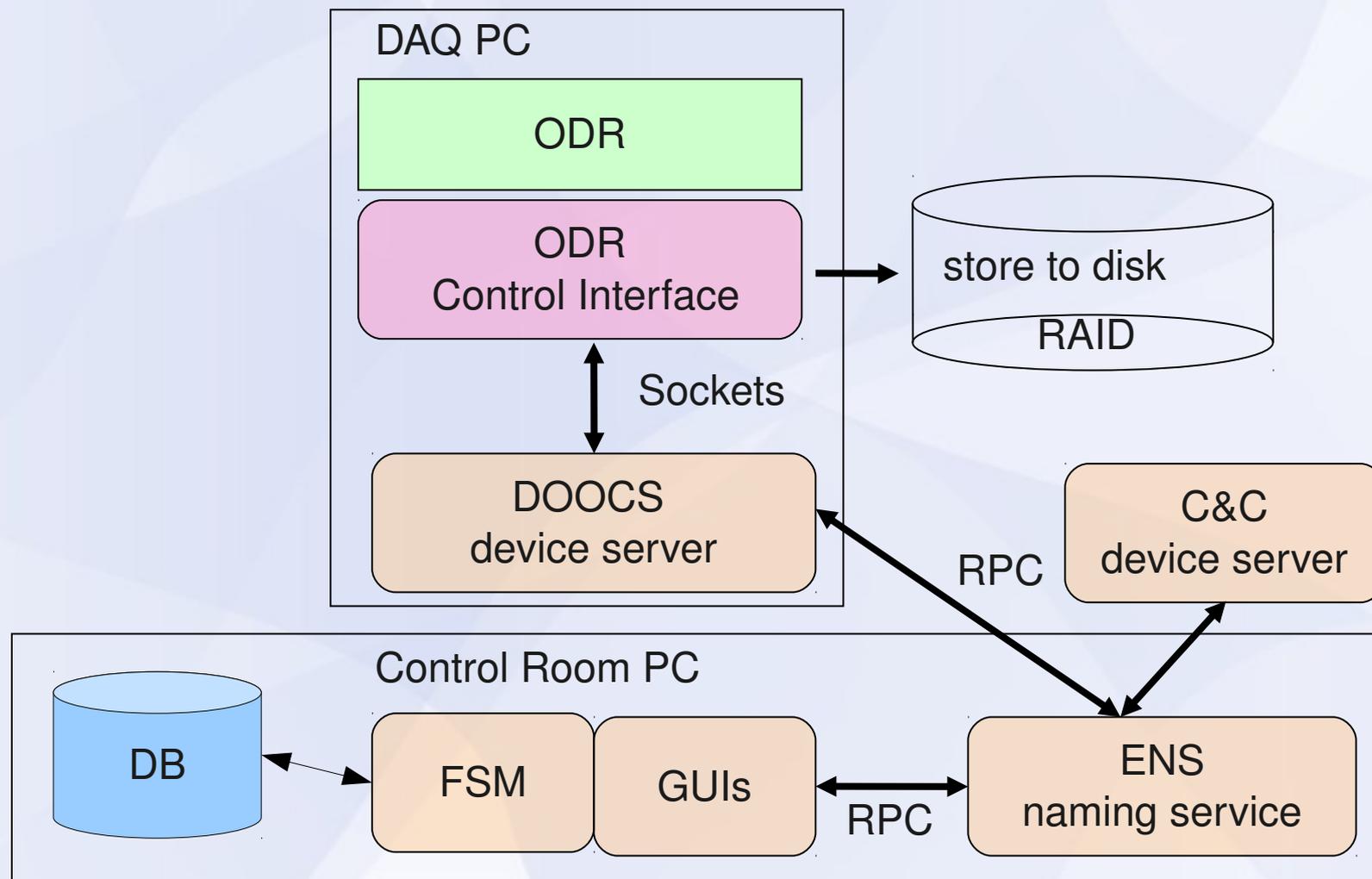
“DAQ2” - Status & Perspectives

“DAQ2” Calice : aperçu général



Topologie type “réseau”
Le hardware pousse les données

DAQ2 SW architecture



+ Simulation algo reassemblage (Read-out)

DAQ2 : Perspectives et besoins

- Début 2010 :
 - Integration/debug de la DAQ anglaise + DCC
 - Programmation des DIFs, DCC
 - Mesurer les performances de l'algo de réassemblage
- Mi-2010 et suite :
 - Soft DAQ réutilisable
 - Xdaq ou Tango pour slow-control
 - EUDAQ pour read-out/stockage/integration avec autres DAQ
 - Intégration dans une DAQ commune (HW+SW) :
 - Intégration avec ILCSoft
 - Gestion trigger unique : TLU + C&C
 - Récupération caractéristiques faisceau : “BIF” ?
 - Récupération caractéristiques machine

DAQ2 : Perspectives et besoins

- mi-2010
 - Installation d'un banc DAQ2 à l'IPNL et/ou au LAPP pour test/debugging (mat donné par UK)
- Q3 2010
 - test en faisceau avec plusieurs plans

Prévision des besoins

- LLR :
 - Matériels : PC, carte ethernet optique
 - Efforts de développement :
 - Courant 2010 :
 - debug/mise au point du coeur de la chaîne avec les électroniciens
 - slow-control/read-out simple (Xdaq/Tango, EUDAQ/maison)
 - À partir de fin 2010 : **renfort manpower souhaité**
 - intégration avec ILCSoft
 - Intégration avec les autres détecteurs (testbeams combinés ; ex. Support CCC+TLU)
 - Récupération des informations de la machine
 - Récupération des caractéristiques faisceau (“Beam InterFace card” ?)

Prévision des besoins

- IPNL
 - ??
- LAPP
 - Production des DIF (ANR)

Bonus slides

Slow-control : Choix retenus

- Embarras du choix pour les SCADA pour HEP:
 - DAQ1, Xdaq, Tango, EPICS, Labview, PVSS, MIDAS, home-made, ...
- Plusieurs utilisés dans ILC
- Pour Calice:
 - Xdaq+Labview par C. Combaret à l'IPNL
 - DAQ2: Évaluation Xdaq + Tango (ESRF)

Data-readout : choix retenus

- Un peu moins de frameworks disponibles:
 - EUDaq, DAQ1, Xdaq, ICE, Narval, home-made (file/socket-based), ...
- Plusieurs utilisés dans ILC:
 - Exemples: Xdaq, EUDaq, DAQ1, Narval, home-made
- Pour Calice:
 - Xdaq par C. Combaret à l'IPNL
 - DAQ1: “simple serialization” par P. Dauncey
 - DAQ2: EUDAQ pour interfaçage avec les autres ? Ou maison
 - Tenter d'aller jusqu'à sauvegarder en LCIO (éval.)

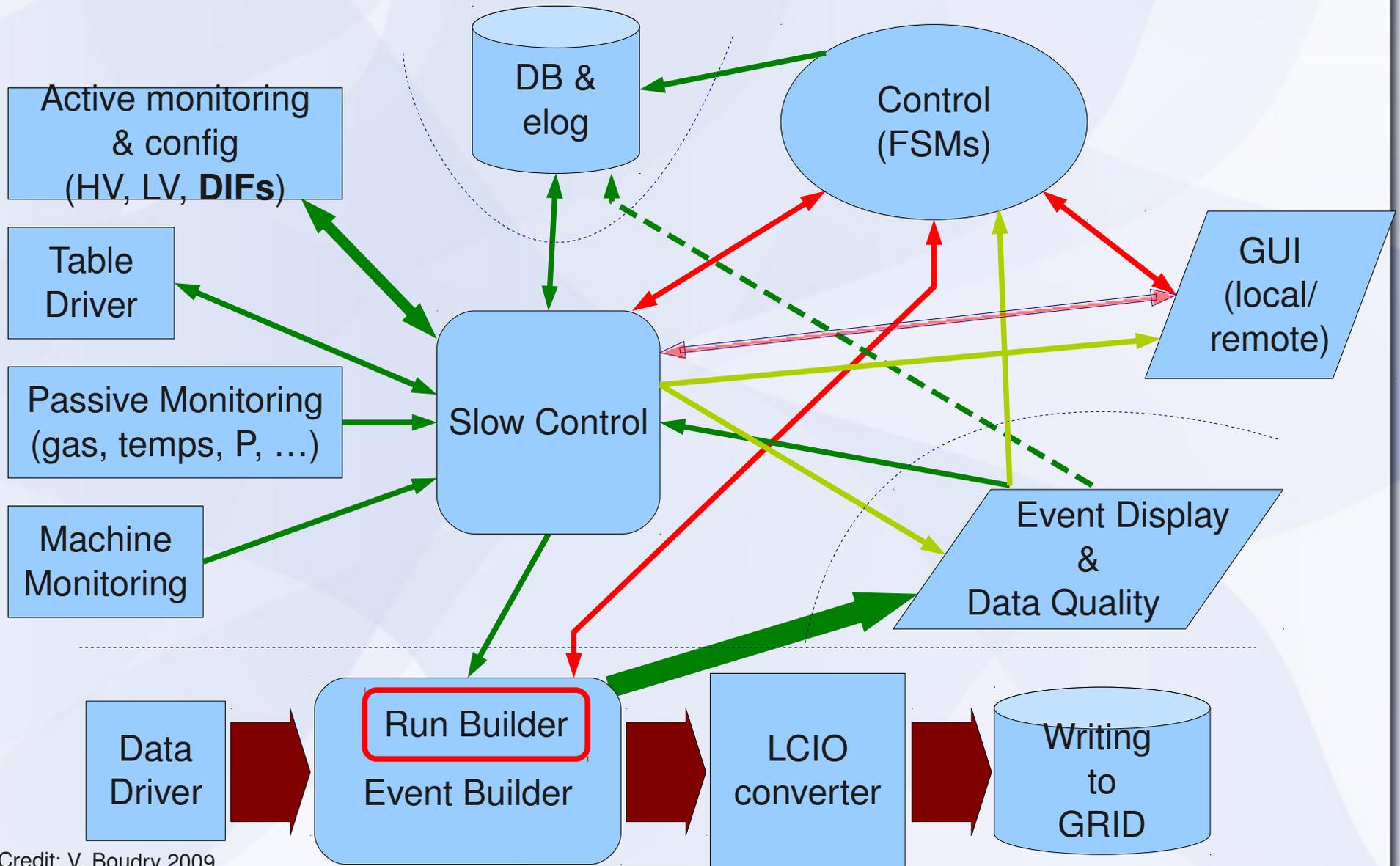
Prévision des besoins

- Matériels : PC, carte ethernet optique
- Efforts de développement :
 - Courant 2010 :
 - debug/mise au point du coeur de la chaîne avec les électroniciens
 - slow-control/read-out simple (Xdaq/Tango, EUDAQ/maison)
 - À partir de fin 2010 : **renfort manpower souhaité**
 - intégration avec ILCSoft
 - Intégration avec les autres détecteurs (testbeams combinés ; ex. Support CCC+TLU)
 - Récupération des informations de la machine
 - Récupération des caractéristiques faisceau (“Beam InterFace card” ?)

Status Software

- Prototype tournant sous DOOCS
 - ODR kernel driver par A. Misiejuk (Manchester)
 - Capable d'enregistrer les paquets de l'ODR sur disque
 - Interface CCC pour DOOCS
 - Accès registres
 - Software DAQ par V. Bartsch (UCL)
 - GUIs de contrôle de ODR + CCC
 - FSM de contrôle des device servers
 - Squelette DB pour les configs

Perspectives : mi 2010 et après



Credit: V. Boudry 2009

Slow-control/supervision

Preparing for integration issues

- (Some other) slow-control subsystems to consider, same integration problems:
 - Configuration
 - System dynamics (FSM)
 - Error handling, logging
 - Online monitoring: rendering with ROOT ?
FROG ? Other ?
 - Trending database
 - ...

Data-readout path

Preparing for integration issues

- (Some) other aspects to consider, new integration issues:
 - Correctly identify readout data coming from different subdetectors as belonging to the same physical event
 - Uniquely identify each trigger ? Specialized hardware (TLU/CCC ?)
 - P. Dauncey: order the readout phases...
 - Need to evaluate this solution for network-oriented DAQ

Data-readout path

Preparing for integration issues

- (Some) other aspects to consider, new integration issues (ctd.):
 - Data volumes: single machine able to store ?
 - Specific hardware needed ?
 - “Intelligent” load-balancing switches to transfer data to PC clusters ?
 - Zero-suppression, selective readout, on-the-wire lossless compression ? Other ?
 - Not really a concern for Calice... how about the other subdetectors ?

– Storage data format ?

Monitoring the beam

- To determine the accurate actual characteristics of the beam
- Usually: readout of Scintillators, Cerenkov, MWPC, fiber hodoscopes, etc. specific and/or supported by the hosting testbeam facility
 - Need to adapt the setup for each testbeam location
- “Beam Interface Card” (BIF): common hardware to monitor the beam
 - Proposals by V. Boudry, R. Cornat, F. Gastaldi (LLR) and J. Prast (LAPP)
 - Hardware card + associated driver software
 - Compatible with the ILC acquisition modes
 - Including auto-triggered: would allow to have an accurate insight into the beam characteristics for each recorded event
- Readout of a wide range of signals used by the devices above

A BIF proposal

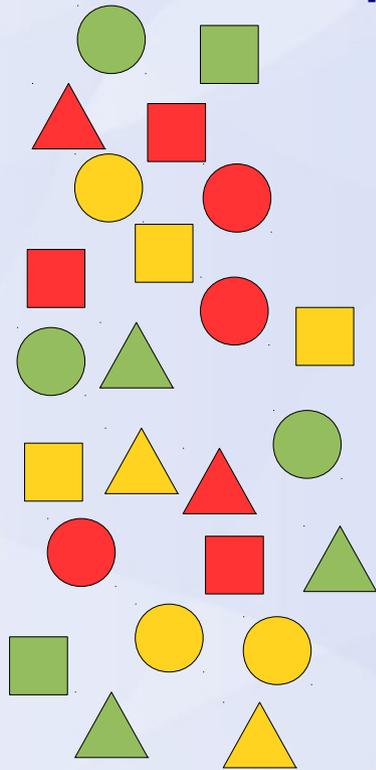
- A ROC-based solution:
 - 1 or 2 ROC receive the external signals and store them
 - Pros/cons:
 - ROC Number of channels seem large enough (eg. 10 scint, 16 C PM, 8 MWPC)
 - SPIROC stores time+amplitude: needed for MWPC
 - Memory size might be too short for DHCAL
 - use 1 HR for digital signals (Scint, C PM), 1 SPIROC for MWPC ?
 - Trigger logic provided by the outside world
 - Readout by a standard DIF
 - More demanding in hardware design manpower

Another BIF proposal

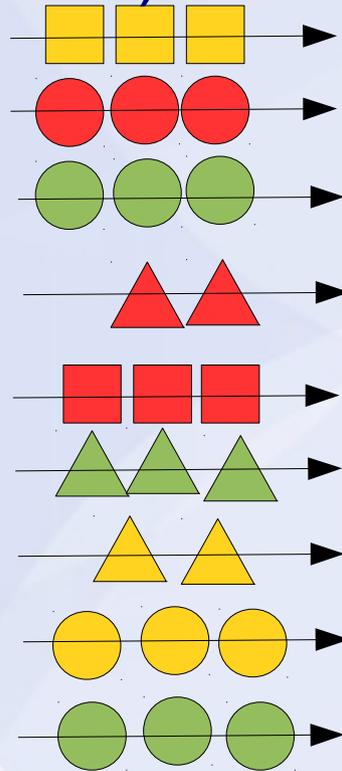
- A DIF-based solution
 - Adapter board + modified DIF
 - Existing hardware, FPGA-based
 - Pros/cons:
 - Cheap+simple adaptation board (~ connectors)
 - Large memory needed onboard FPGAs
 - No amplitude recording
 - “Easily” reconfigurable trigger logic
 - More demanding in VHDL manpower

Réassemblage de paquets

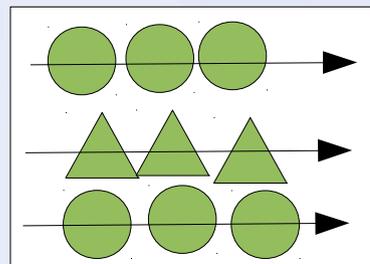
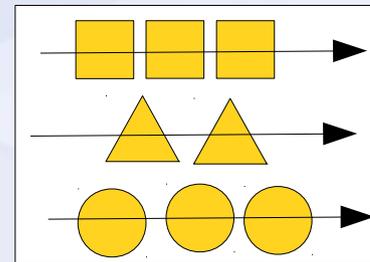
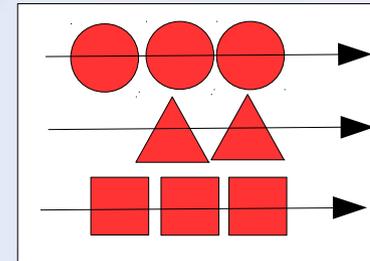
- Simulation réassemblage (“event builder” primitif)



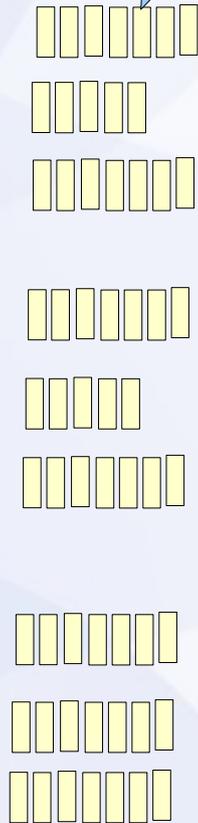
ODR
pkts



ROC
chains



Events



ROC
Data

1 ROC
“f” frames

LCIO

Interfacing with the machine

Monitoring the environment

→ To get get knowledge about:

- beam energy & particle type (\leftrightarrow magnets current & upstream collimators position)
 - beam intensity ("counting devices")
 - final collimators position
 - beam profile (monitoring chambers histograms)
 - Cherenkov pressure
 - and others: existing environmental measures (P, t, humidity), ...
- Have a common API for all the machines ?
 - All this with fast access ($\leq 30s$)
- An open question ?

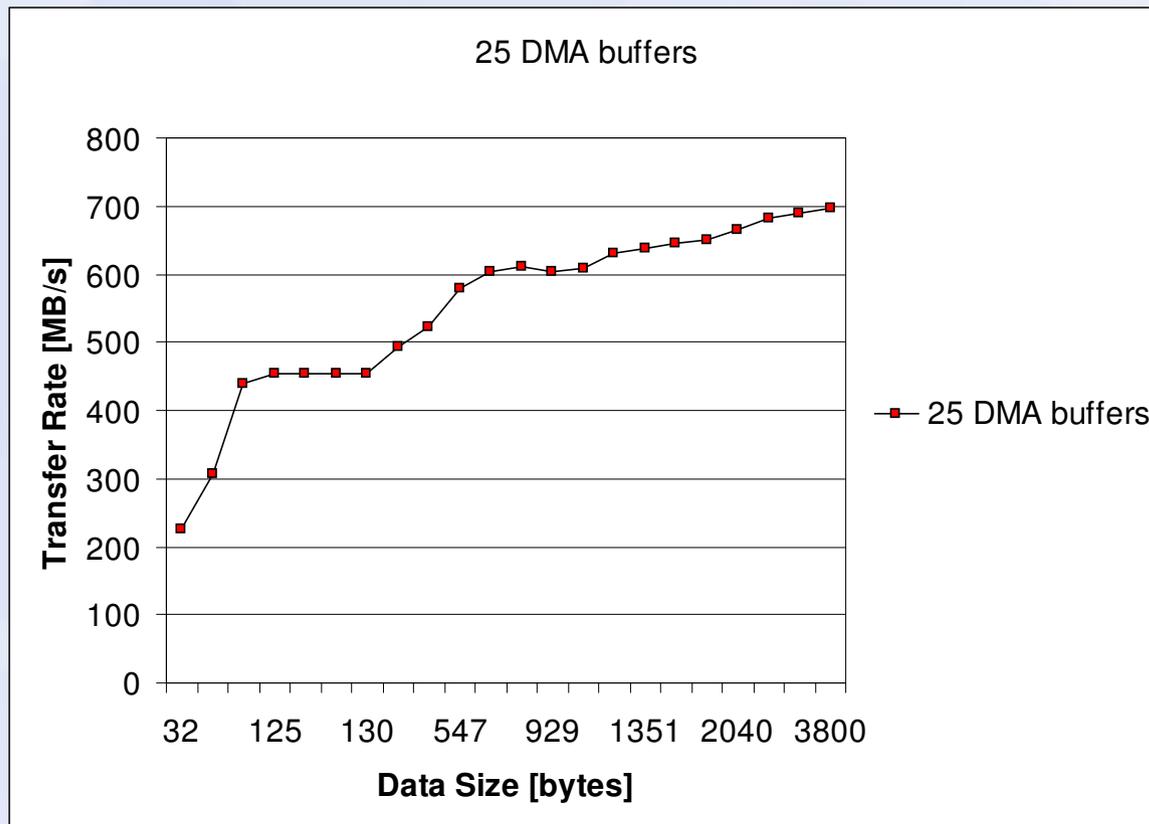
DAQ Framework: Tango ?

- <http://www.tango-controls.org/>
- Used by many synchrotron exp. (ESRF, DESY, Soleil...)
- But: the base is NOT synchrotron-specific
- Generic DAQ distributed system (slow-control)
 - Distributed configuration stored in DB
 - Service interaction through an ORB (~ RPC bus)
- Nothing really technically original, but more modern, nicer, more mature, less DESY-centric than Doocs

DAQ Framework: Tango ?

- The things that made me enthusiastic:
 - A real 3rd-party blob: we are users of the framework, we don't go *inside* the framework
 - Simple generic & composable message data types between services
 - Reasonably language-agnostic: C++, java, python
 - Can use jddd (Doocs GUI builder)
 - Management can be achieved via GUIs
 - A fast “device server design” approach (GUI code-generator)

DAQ Prototype: Performances



Evaluation by V. Bartsch/T. Wu/UCL/Manchester