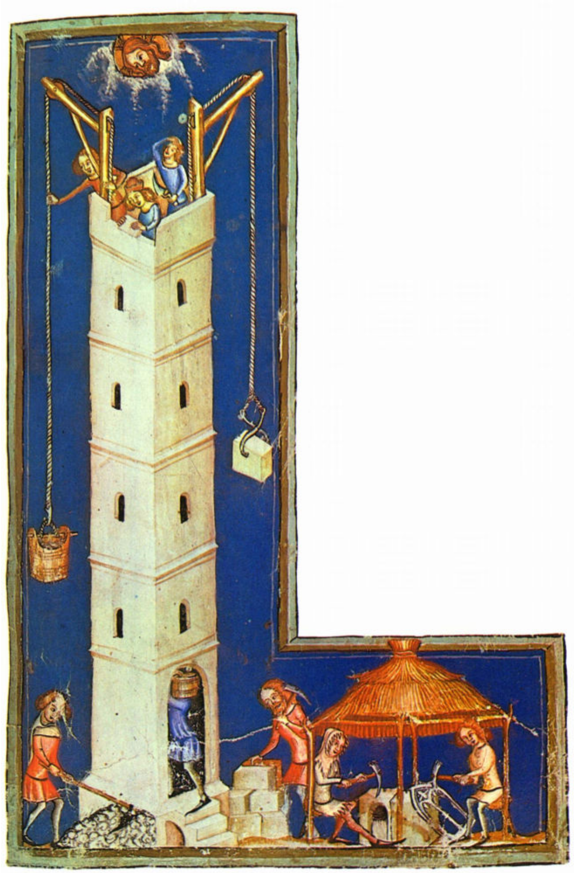
RPC-DHCAL
Event Builder
Event Display

Jacob Smith

UT Arlington

Argonne National Laboratory



Meister der Weltenchronik
(Master of World Chronicles)

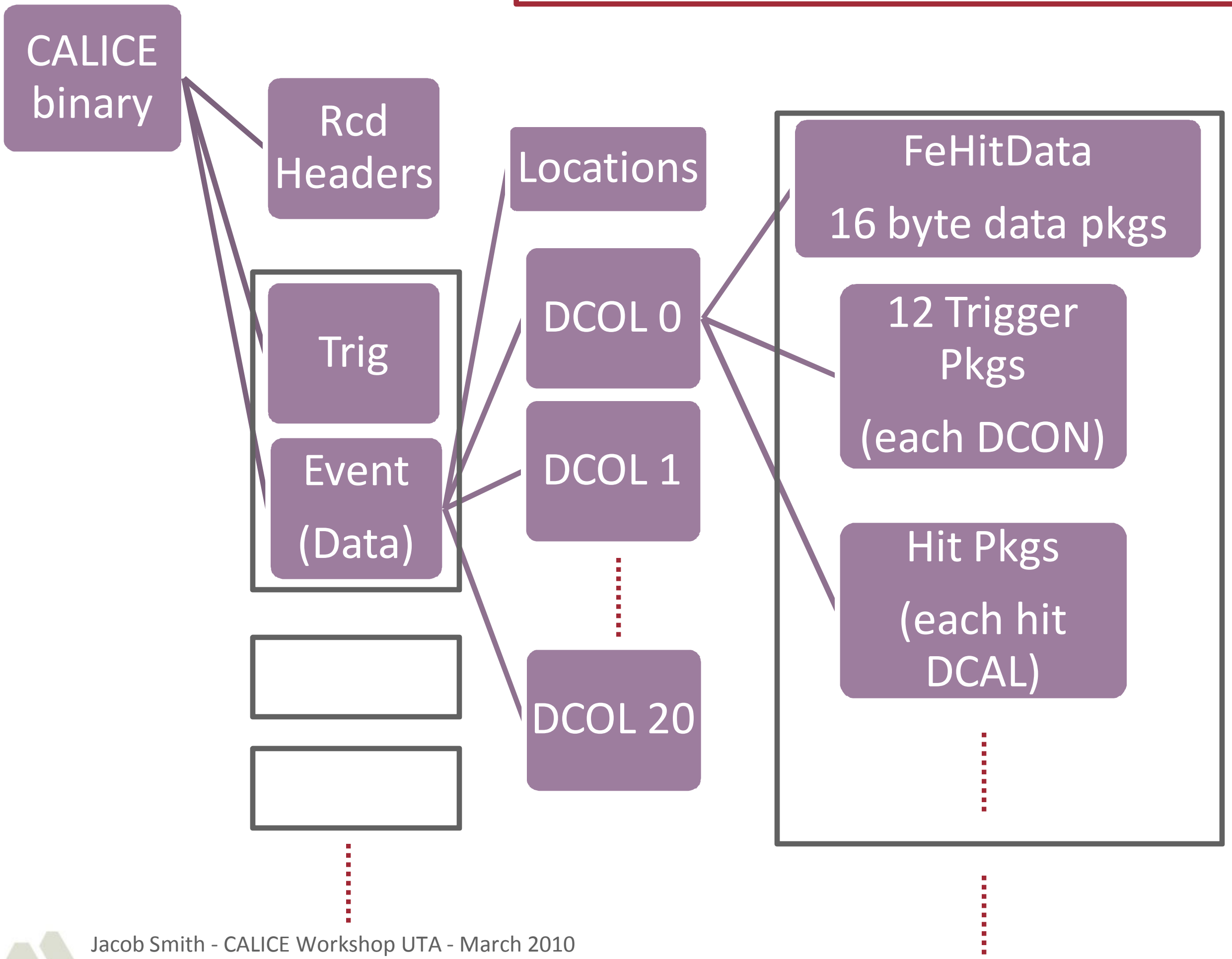
Genesis 11:1-9

Outline

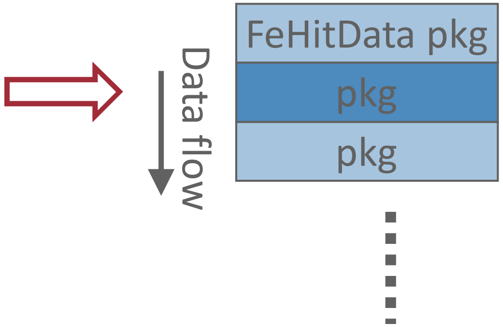
- Introduction
  - DHCAL/calice-daq Data format
- Status of Event Builder
- Frameworks for Event Builder and Event Display
- Show event display capabilities
  - Cubic meter
  - Cosmic rays
- Conclusion

DHCAL/calice-daq Data Structure

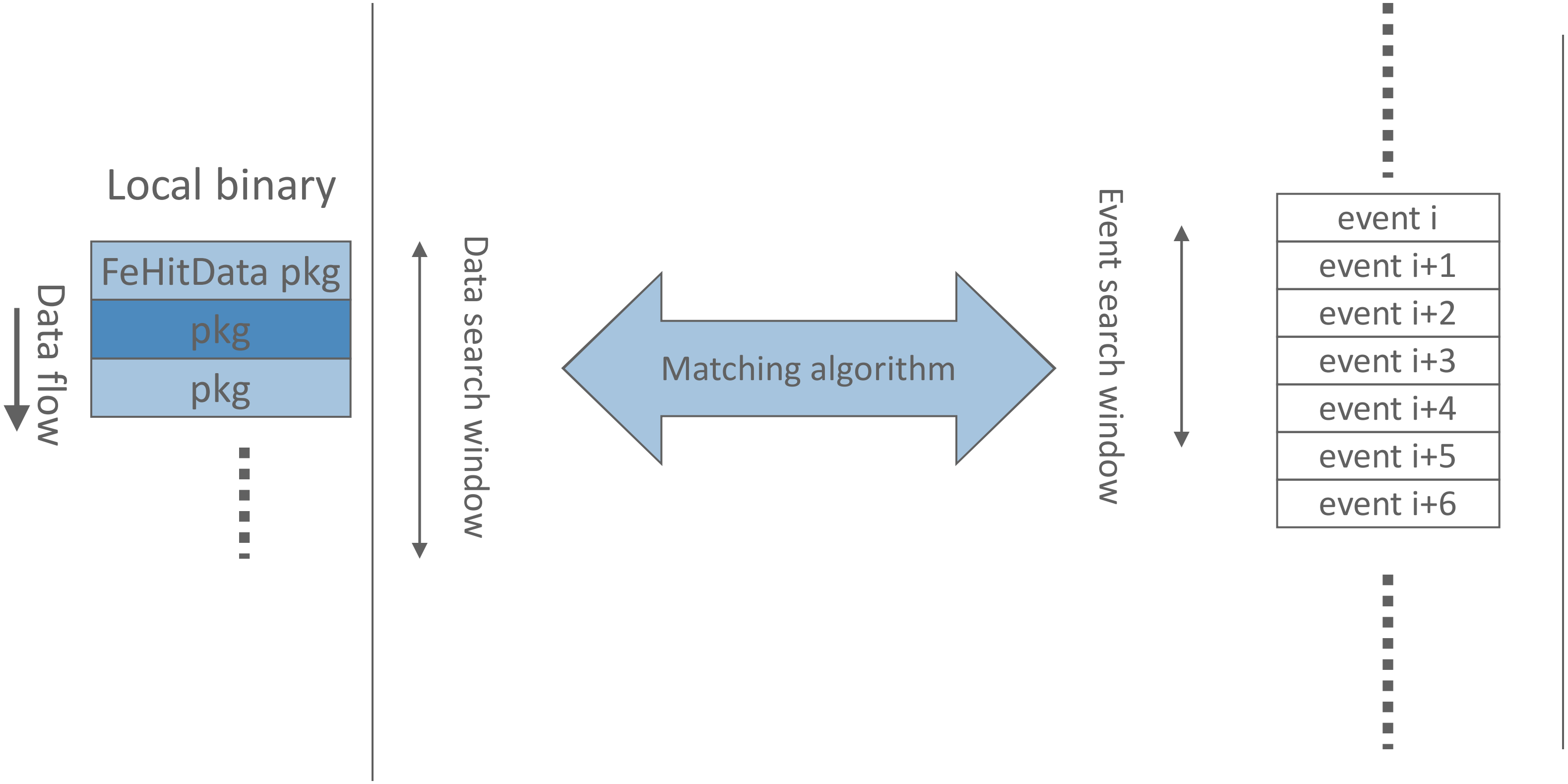
We need to define event boundaries
- Events spread through DCOLs
- Multiple events in each DCOL



VST: use a single DCOL, create new binary format, then build events



Current/Previous (java) Event Building



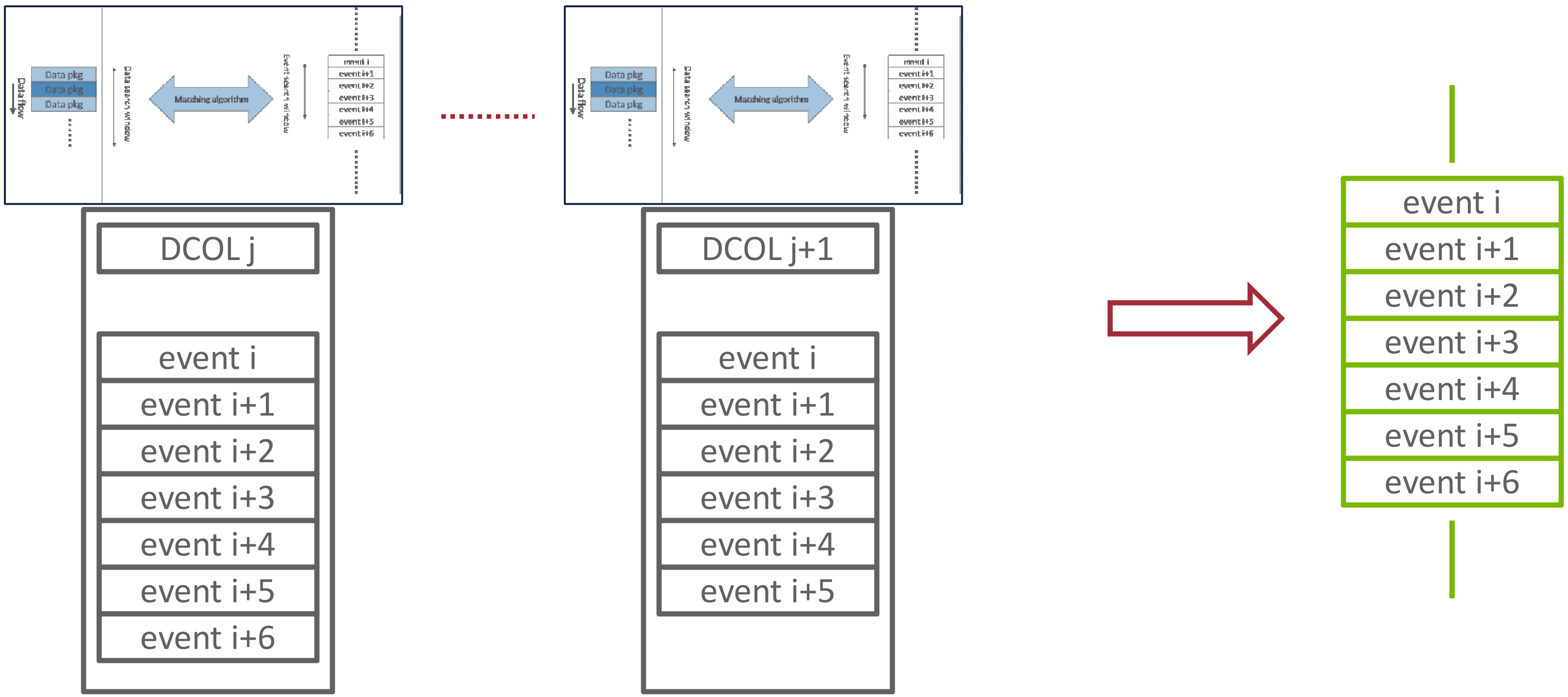
Defines boundaries well for a single DCOL, with the help of:

1. Time stamp
2. DCON trigger package
3. Neighboring packages
4. Error detection and recovery

Event Builder Design: Cubic Meter

- Each DCOL gets an "event builder"
    - C++ translated from experienced java
    - Some code overlap exists

- Timestamp matching algorithm merges data packages from all DCOLs into complete events
    - Reuse algorithms already in place



DCOL j

event i
event i+1
event i+2
event i+3
event i+4
event i+5
event i+6

DCOL j+1

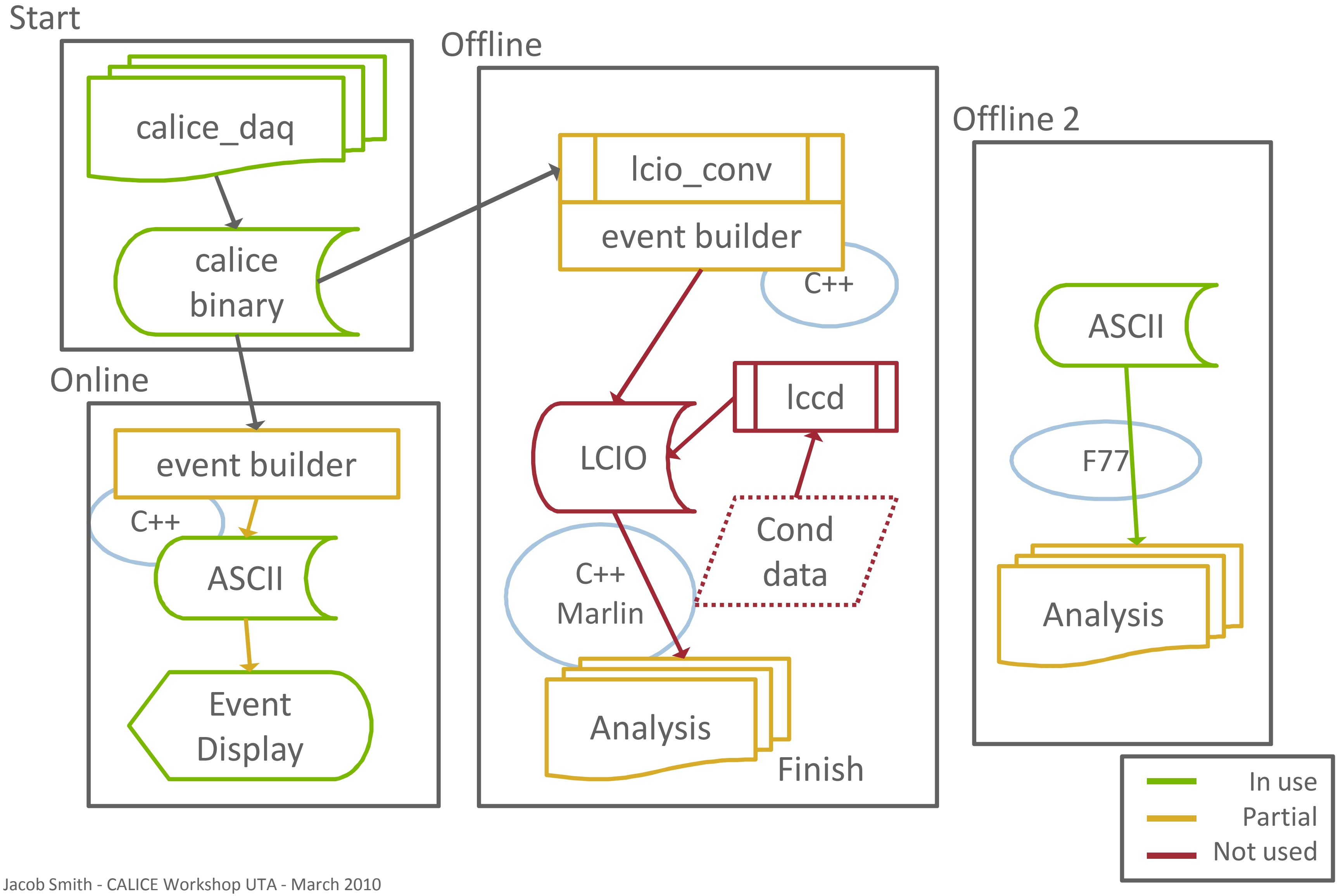
event i
event i+1
event i+2
event i+3
event i+4
event i+5

event i
event i+1
event i+2
event i+3
event i+4
event i+5
event i+6

Status: Cubic Meter Event Builder

- Framework for new event builder is finish
  - event_builder (application)
  - EbDhcEvent.hh, EbDhcFeDataHit, EbDhcFeDataTrig (EB classes)
  - DhcFeHitData.hh (calice binary)
    - chksum(), dcolad(), timestamp()
    - Jim Schlereth developed local version of calice_daq
- Finishing translation of java into C++
  - Some algorithm overlap exists between calice_daq (dhc) and java
    - Reading different binary formats
- Start testing final event builder
  - Take data with multiple data collectors
- Start developing event builder for trigger-less data
  - Finding cosmic ray events from noise runs
  - Build events without help from DCON trigger packages
- LCIO conversion with final event builder not yet started

Analysis: Cubic Meter



Start

calice_daq

calice binary

Online

event builder

C++

ASCII

Event Display

Offline

lcio_conv

event builder

C++

LCIO

lccd

Cond data

C++ Marlin

Analysis

Finish

Offline 2

ASCII

F77

Analysis

🟩 In use

🟨 Partial

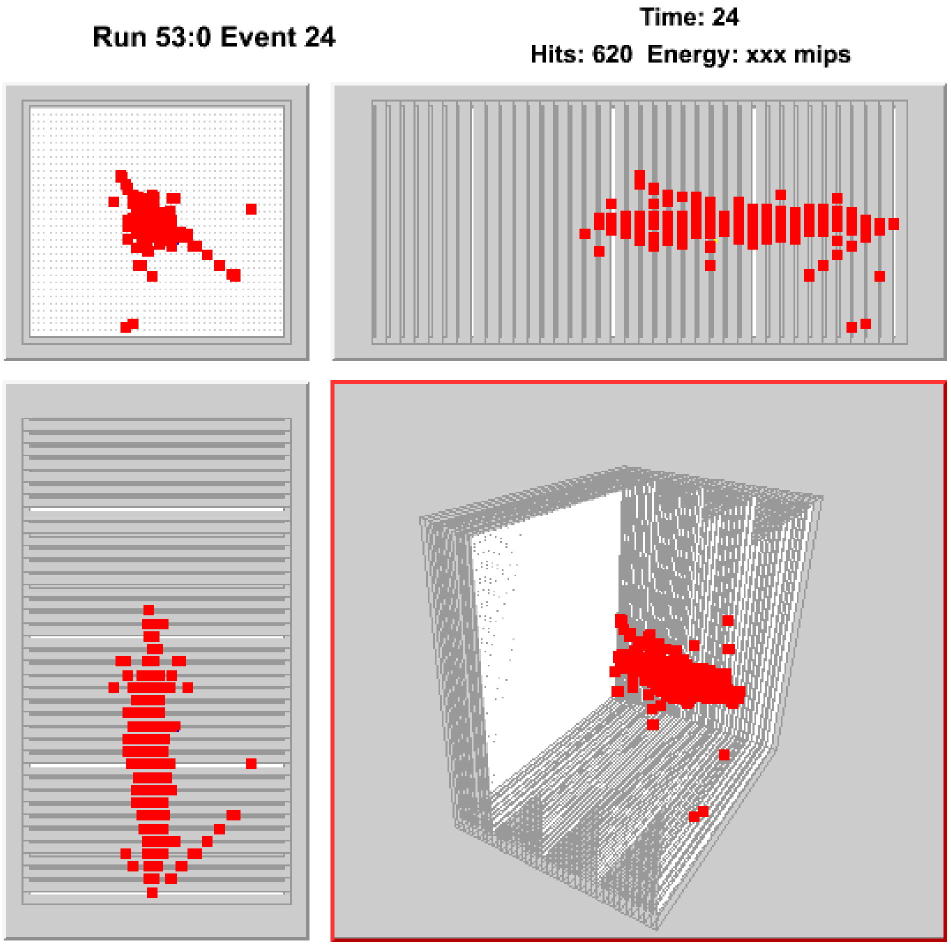
🟥 Not used

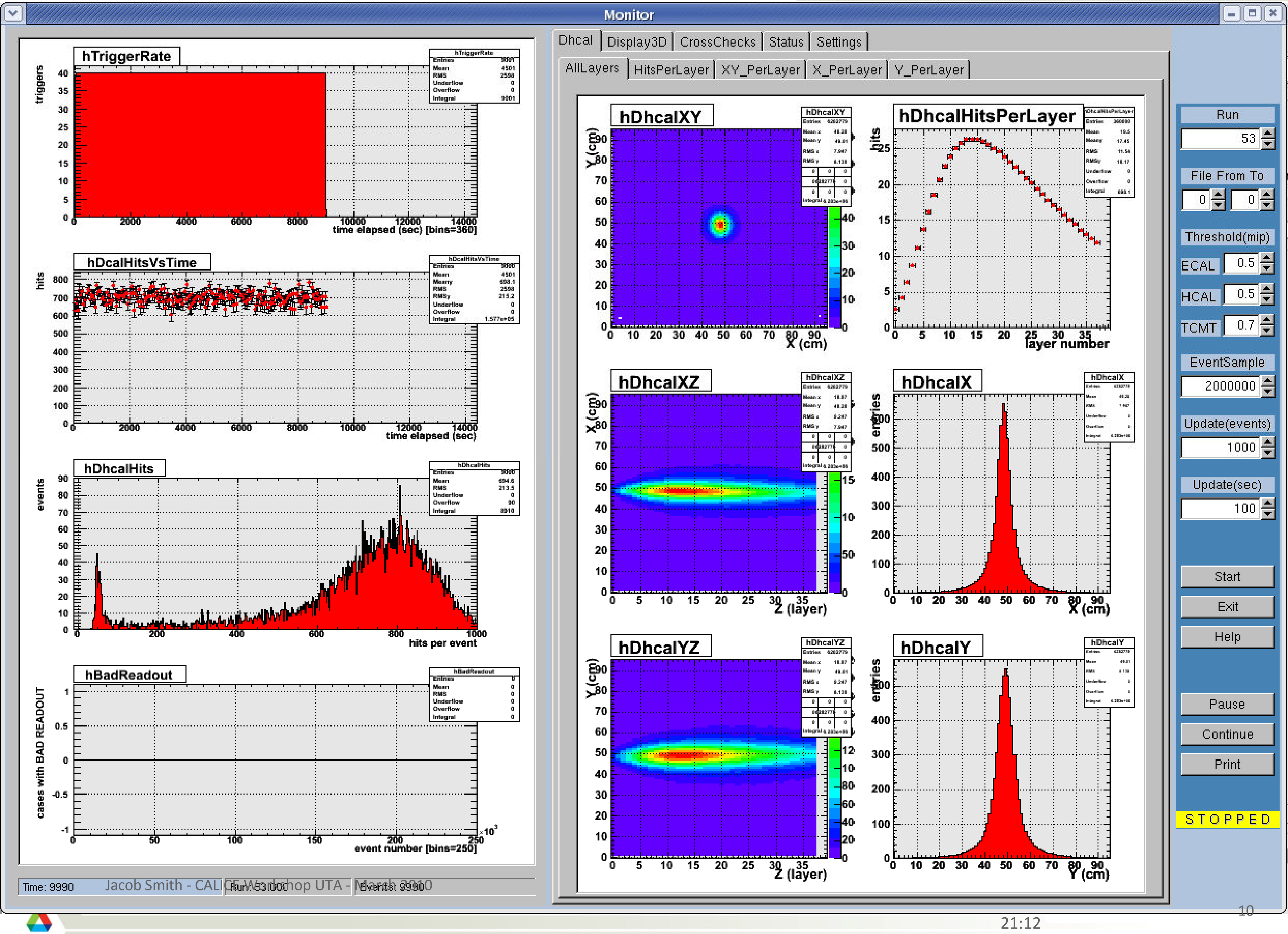
Strategy: Event Builder and Event Display

- Very short term plan (days)
  - Event Builder/Display for RPC commissioning in Cosmic Ray Test Stand
    - Part of RPC construction procedure
    - Finish Event Builder translation
    - Start testing with multiple DCOLs (on-hand)
    - An application produces ASCII files using Event Builder code
    - Event Display works from ASCII file
    - Event Builder for trigger-less data
- Short term plan (weeks)
  - EB and ED for cassette performance tests
    - Finish Event Builder testing and commissioning for analysis
    - Begin implementation with lcio_conv for offline analysis
  - C++ based analysis for cosmic tests
    - Daniel Trojand is developing c++ based analysis
  - Development of EB for trigger-less data
- Test beam plan (months)
  - Will use LCIO converter implemented with Event Builder code for offline analysis
  - Will also use ASCII + F77 as a second analysis path
  - Event Display samples events during data taking

Event Display - Cubic Meter Simulations

- George Mavromanolakis developed the event display for the VST
  - Based on CALICE online display program
  - Reads in ASCII data format
  - Kurt Francis updated for cubic meter
- Monte Carlo 60 GeV pions
  - Currently use Geant4 and RPCsim
  - Kurt Francis is developing RPC simulation with Mokka





Run: 53

File From To: 0 0

Threshold(mip)

ECAL: 0.5

HCAL: 0.5

TCMT: 0.7

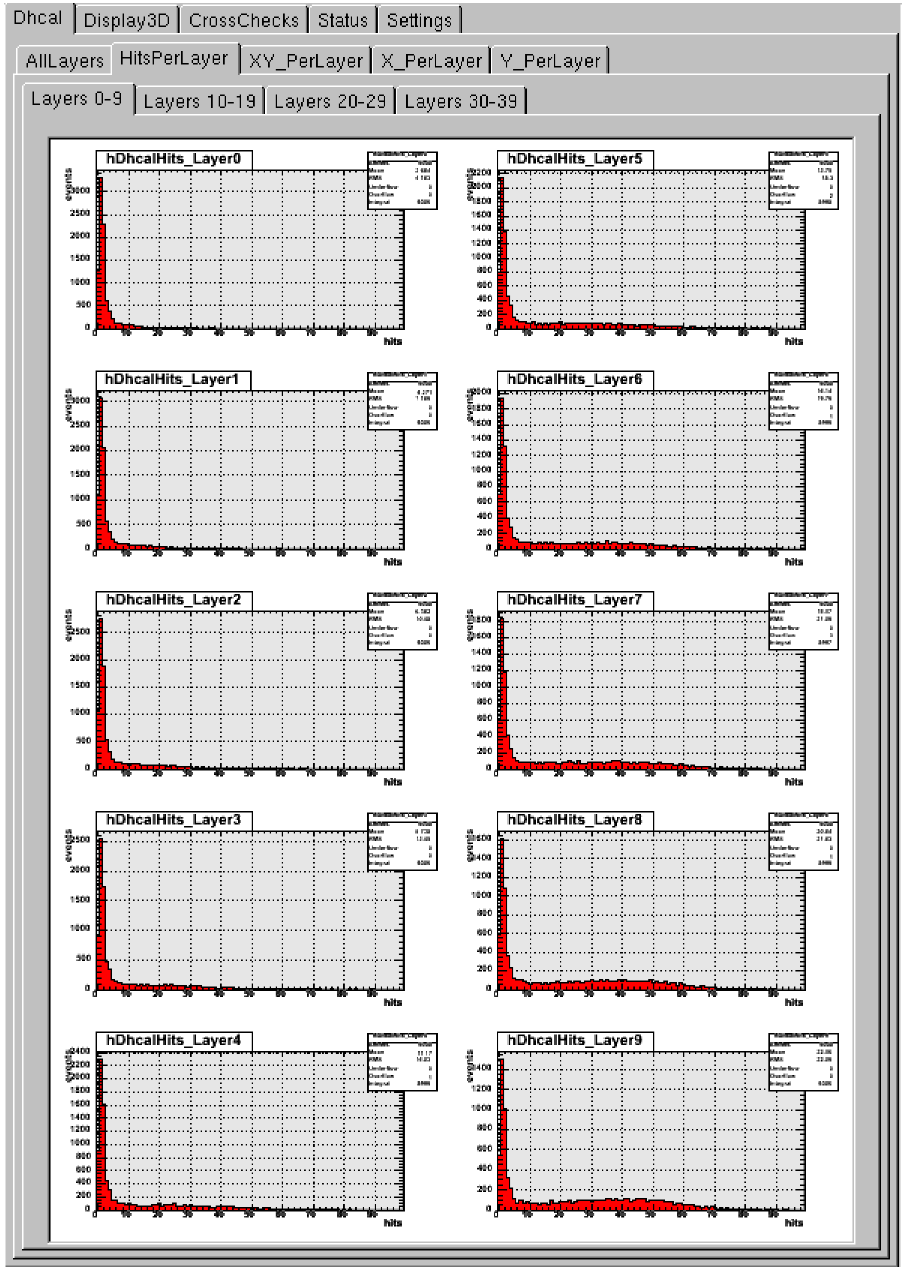
EventSample: 2000000

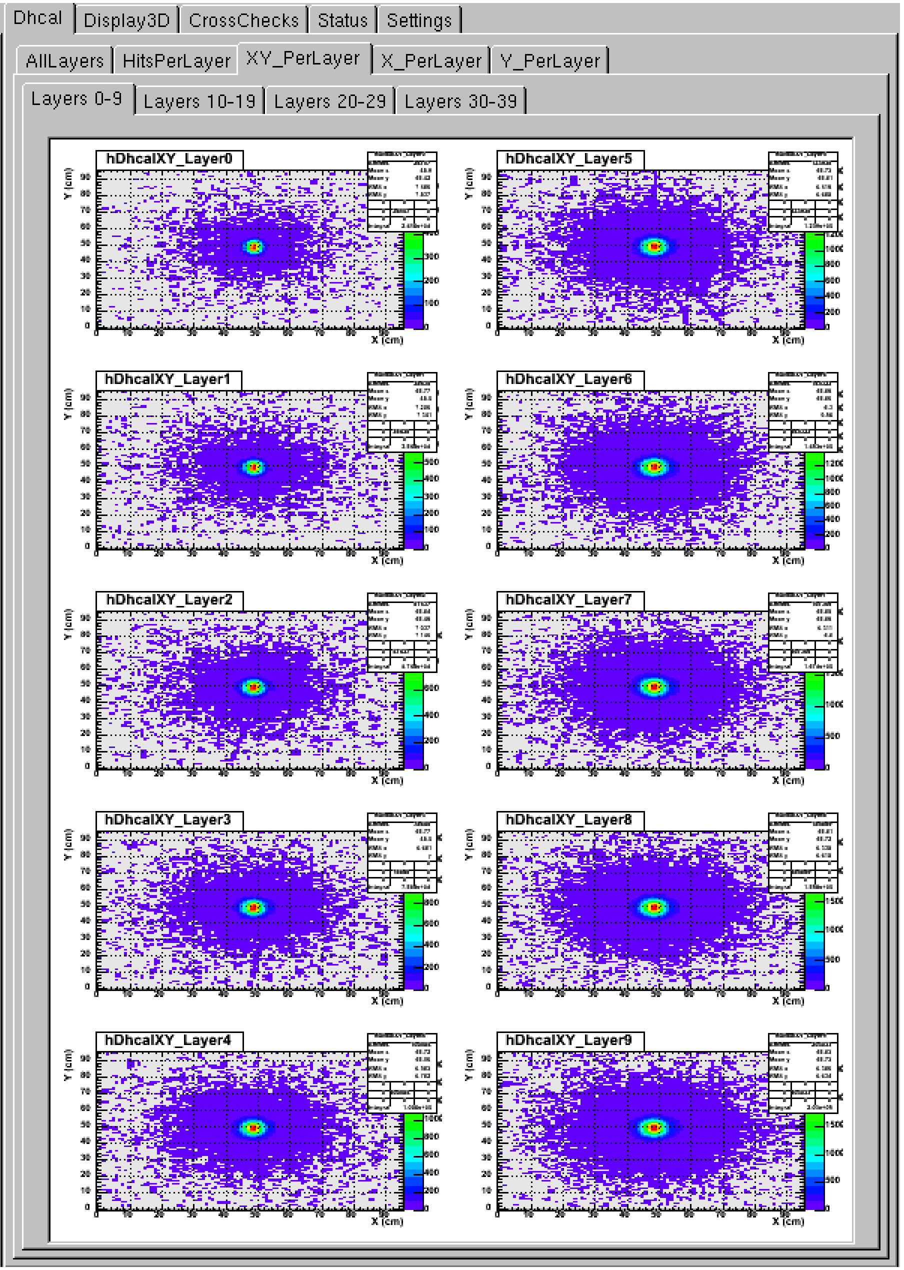
Update(events): 1000

Update(sec): 100

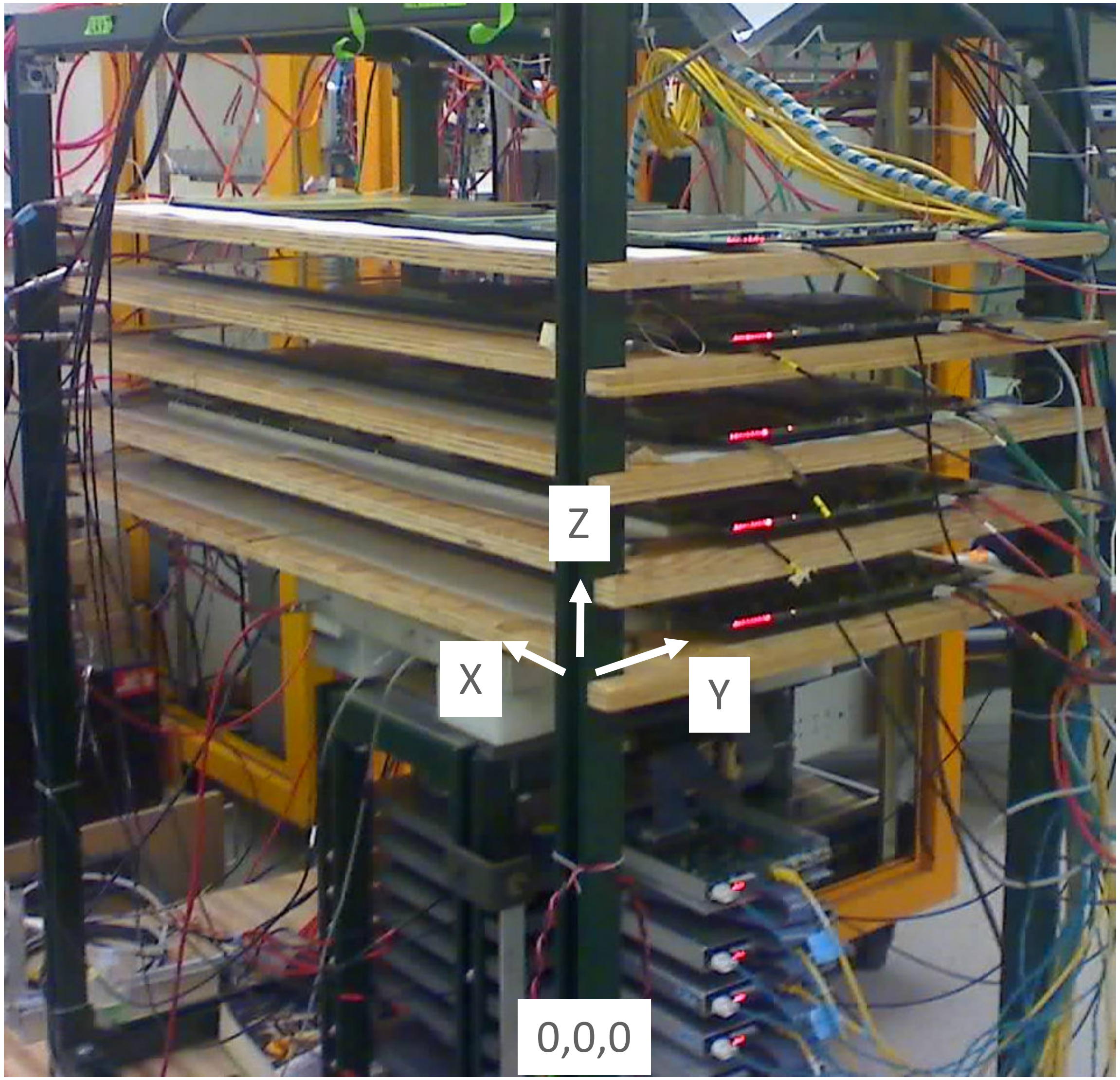
STOPPED

Time: 9990 | Run: 53000 | Events: 9990

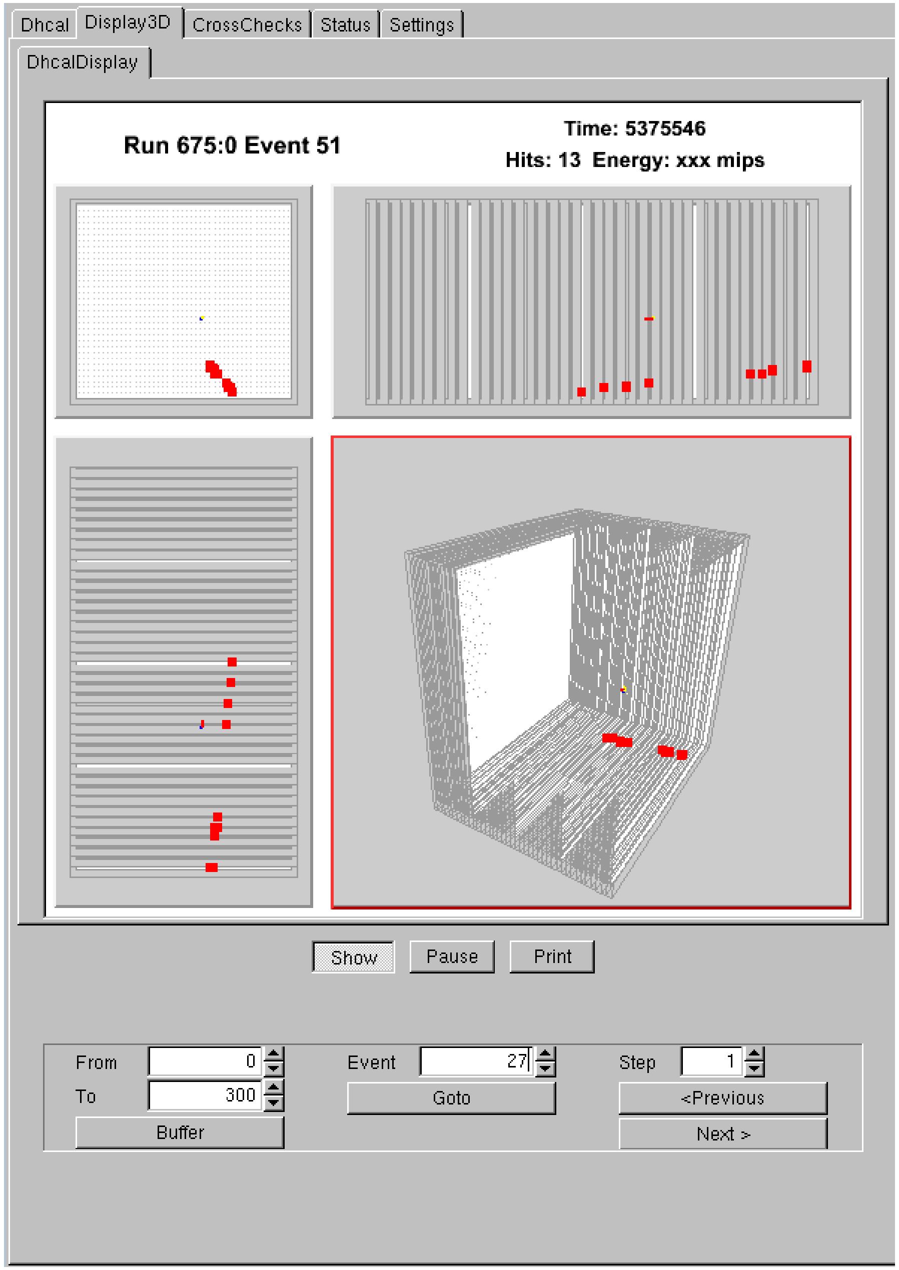

Dhcal
Display3D
CrossChecks
Status
Settings
AllLayers
HitsPerLayer
XY_PerLayer
X_PerLayer
Y_PerLayer
Layers 0-9
Layers 10-19
Layers 20-29
Layers 30-39
hDhcalHits_Layer0
hDhcalHits_Layer1
hDhcalHits_Layer2
hDhcalHits_Layer3
hDhcalHits_Layer4
hDhcalHits_Layer5
hDhcalHits_Layer6
hDhcalHits_Layer7
hDhcalHits_Layer8
hDhcalHits_Layer9
events
hits


Dhcal
Display3D
CrossChecks
Status
Settings
AllLayers
HitsPerLayer
XY_PerLayer
X_PerLayer
Y_PerLayer
Layers 0-9
Layers 10-19
Layers 20-29
Layers 30-39
hDhcalXY_Layer0
hDhcalXY_Layer1
hDhcalXY_Layer2
hDhcalXY_Layer3
hDhcalXY_Layer4
hDhcalXY_Layer5
hDhcalXY_Layer6
hDhcalXY_Layer7
hDhcalXY_Layer8
hDhcalXY_Layer9
Y (cm)
X (cm)

Event Display with CRTS



3D Display



Run 675:0 Event 51

Time: 5375546

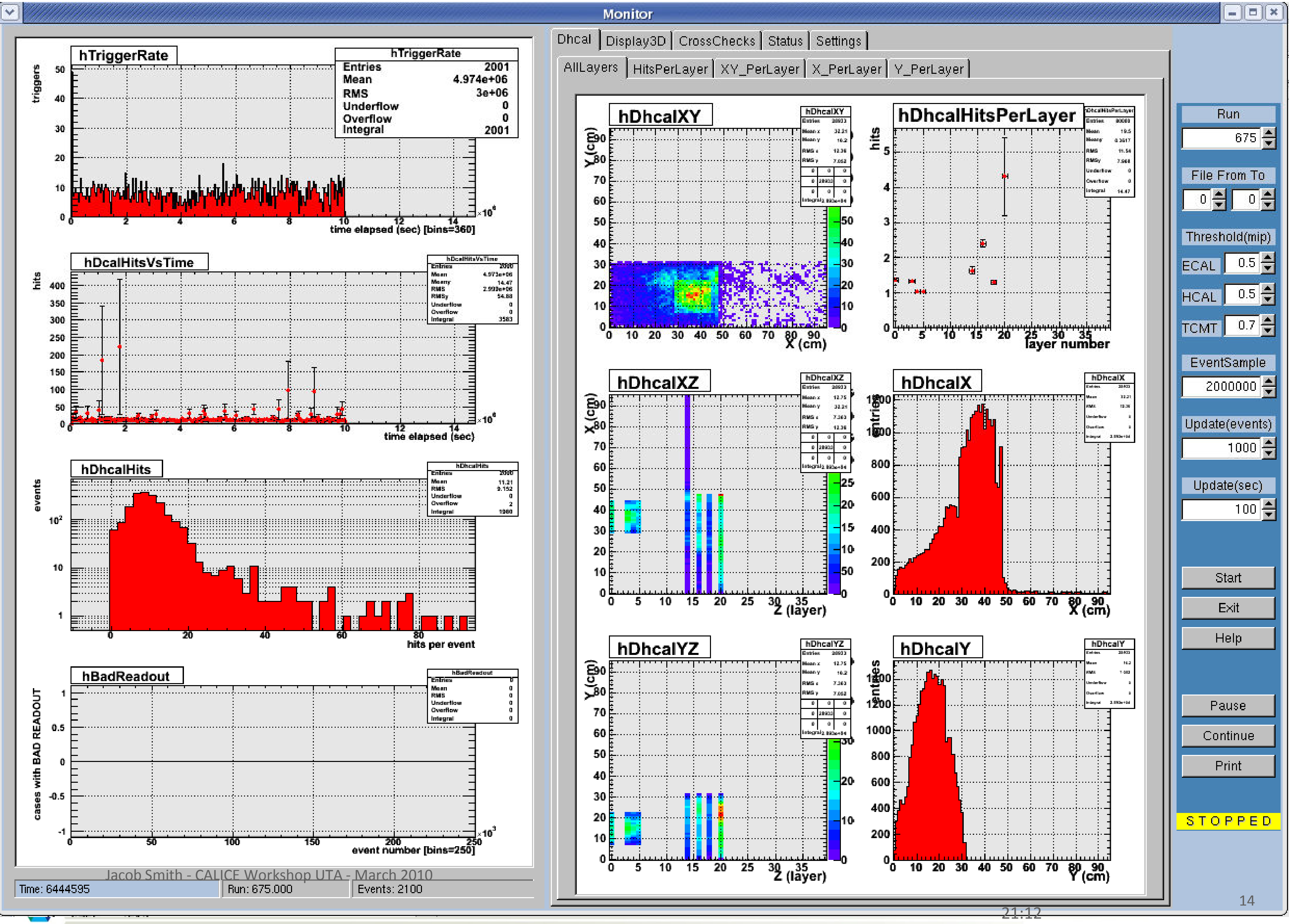
Hits: 13 Energy: xxx mips

From 0

To 300

Event 27

Step 1



hTriggerRate
Entries 2001
Mean 4.974e+06
RMS 3e+06
Underflow 0
Overflow 0
Integral 2001

hDcalHitsVsTime
Entries 2000
Mean 4.973e+06
Meany 14.47
RMS 2.900e+06
RMSy 54.88
Underflow 0
Overflow 0
Integral 3583

hDhcalHits
Entries 2000
Mean 11.21
RMS 9.152
Underflow 0
Overflow 2
Integral 1980

hBadReadout
Entries 0
Mean 0
RMS 0
Underflow 0
Overflow 0
Integral 0

Run: 675

File From To: 0 0

Threshold(mip)
ECAL 0.5
HCAL 0.5
TCMT 0.7

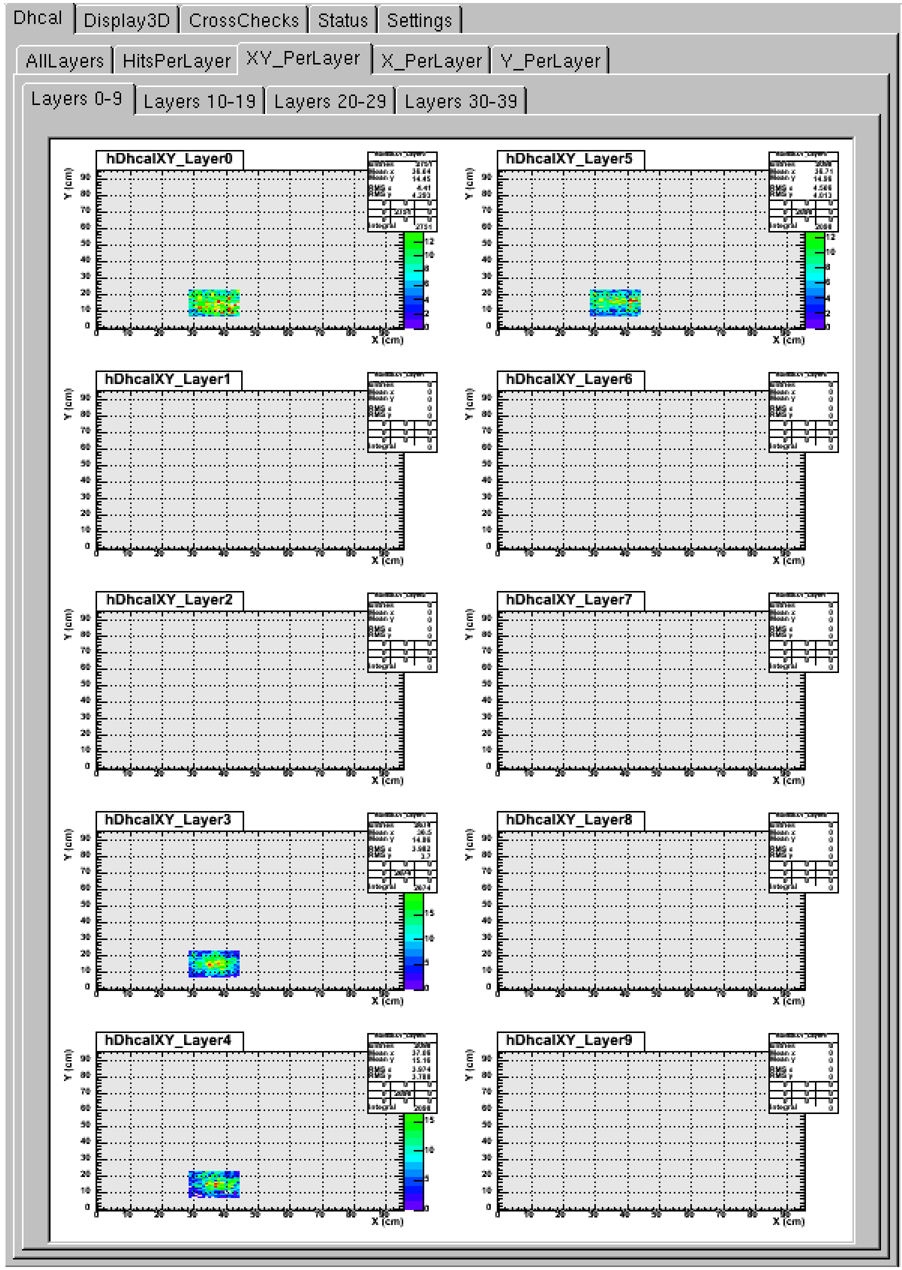
EventSample: 2000000

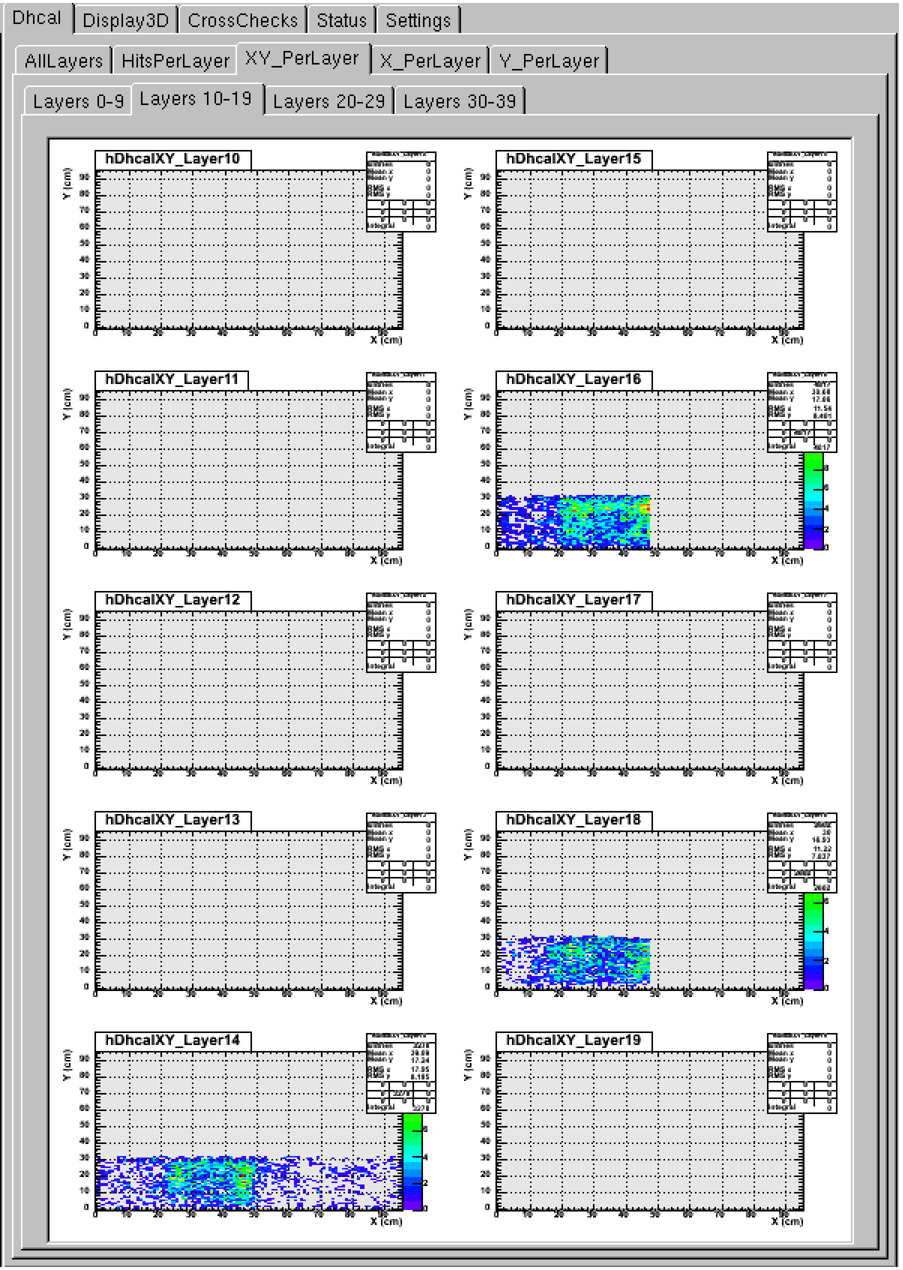
Update(events): 1000

Update(sec): 100

STOPPED

Time: 6444595 | Run: 675.000 | Events: 2100


Dhcal
Display3D
CrossChecks
Status
Settings
AllLayers
HitsPerLayer
XY_PerLayer
X_PerLayer
Y_PerLayer
Layers 0-9
Layers 10-19
Layers 20-29
Layers 30-39
hDhcalXY_Layer0
hDhcalXY_Layer5
hDhcalXY_Layer1
hDhcalXY_Layer6
hDhcalXY_Layer2
hDhcalXY_Layer7
hDhcalXY_Layer3
hDhcalXY_Layer8
hDhcalXY_Layer4
hDhcalXY_Layer9
Y (cm)
X (cm)

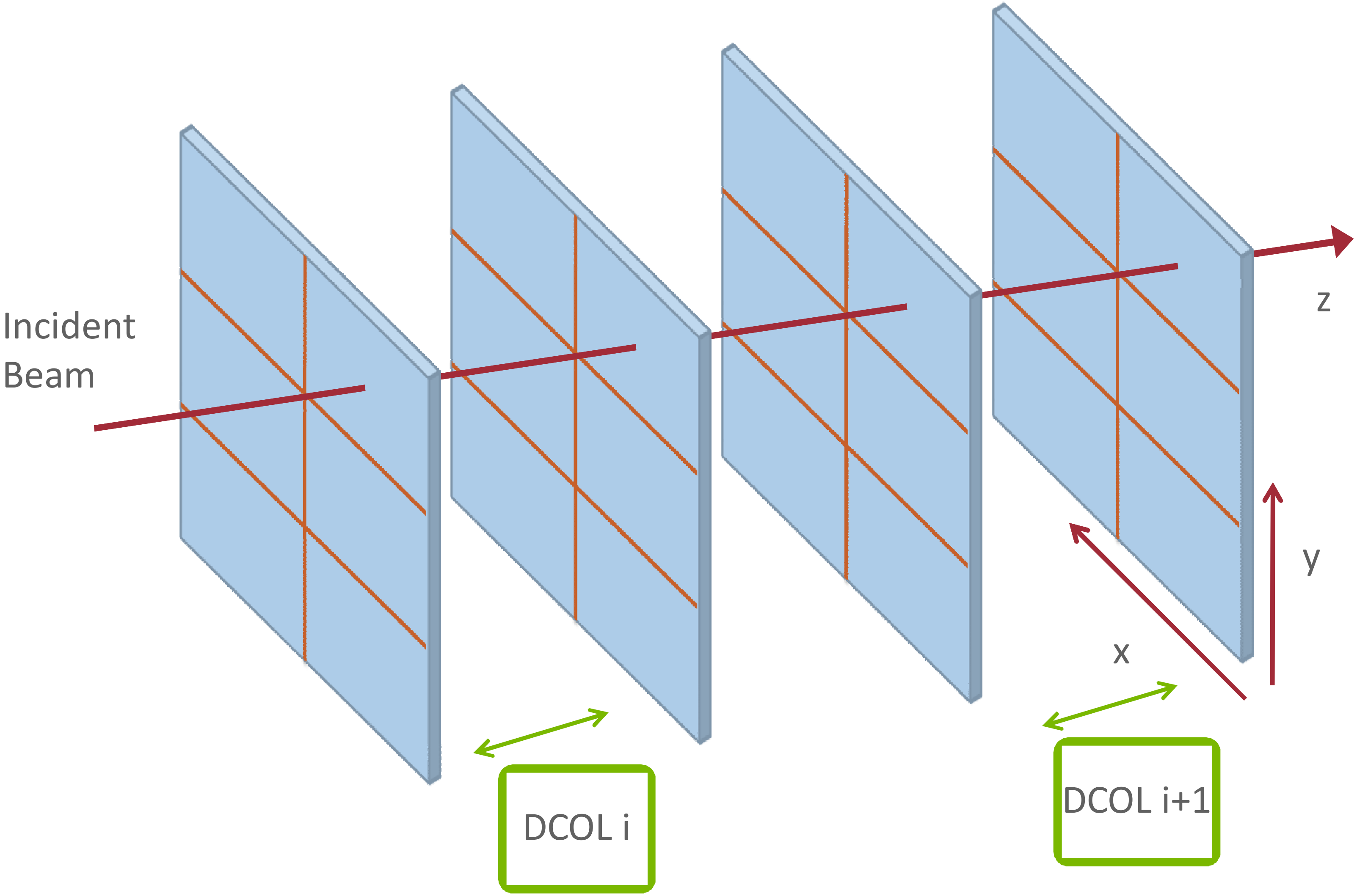

Dhcal
Display3D
CrossChecks
Status
Settings
AllLayers
HitsPerLayer
XY_PerLayer
X_PerLayer
Y_PerLayer
Layers 0-9
Layers 10-19
Layers 20-29
Layers 30-39
hDhcalXY_Layer10
hDhcalXY_Layer15
hDhcalXY_Layer11
hDhcalXY_Layer16
hDhcalXY_Layer12
hDhcalXY_Layer17
hDhcalXY_Layer13
hDhcalXY_Layer18
hDhcalXY_Layer14
hDhcalXY_Layer19
Y (cm)
X (cm)

Conclusions

- Cubic Meter Event Builder design is complete and ready for testing soon
  - Basic (java-based) event builder has history of quality performance
  - Translation from java to c++ currently under way
  - Will start Event Builder testing with multiple DCOLs as soon as possible
    - DCOLs are on hand
- Event Display is finished and ready for…
  - Individual RPC chamber performance testing
  - Cassette testing
  - test beam geometry
- LCIO conversion with event builder codes is on the ToDo List
  - Help with LCIO conversion from experts will be appreciated

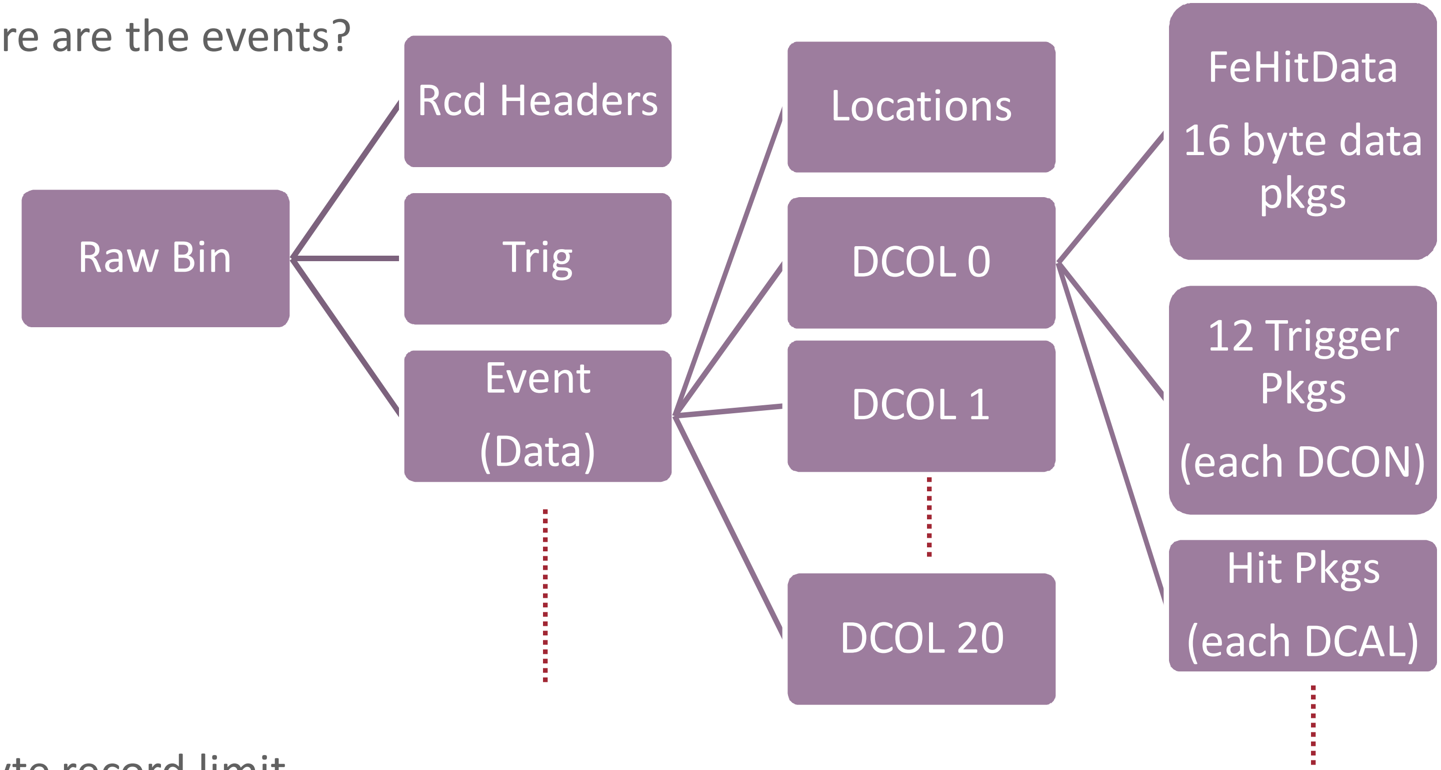
Go Sox!

Backup


Incident Beam
z
y
x
DCOL i
DCOL i+1

DHCAL/calice-daq Data Structure

Where are the events?



- 64kbyte record limit
  - 4000 (16byte) data pkgs spread through 20 dcol
- 240 data pkgs/record for trg pkgs always
- Up to 3760 data pkgs/record available for hit pkgs
- 3760*64 = 240,640 hits (i.e. half the cubic meter)

Events spread through DCOLs
Multiple events in each DCOL