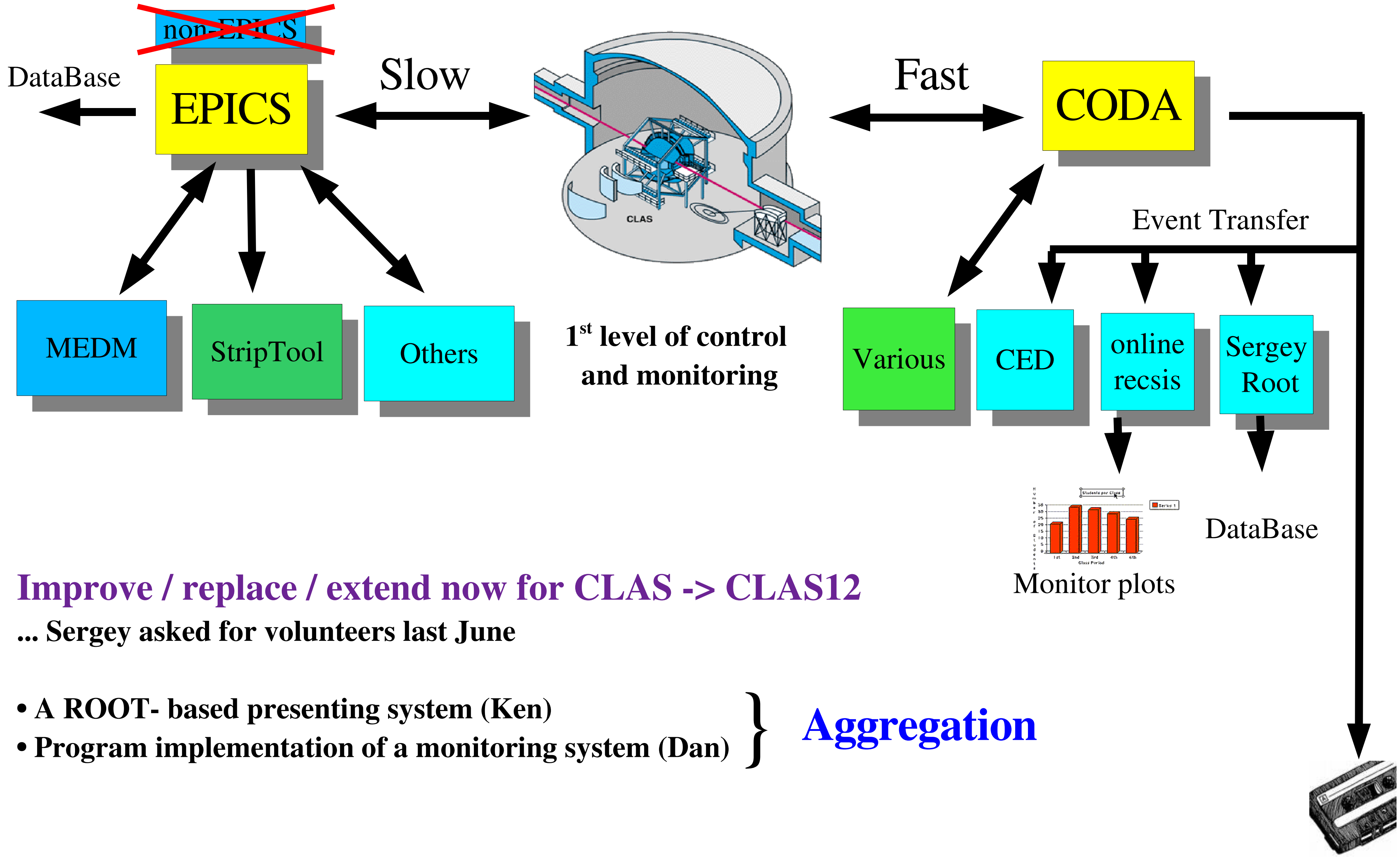
What can Glasgow do for CLAS12 software ?

	Ken Livingston	Dan Protopopescu	Others ?
Time (FTE):	~0.3	~0.2 (maybe more)	?
Experience:	ROOT analysis, tools gui, +epics EPICS device support gui UNIX sys admin, farming etc. VME, FASTBUS etc.	ROOT analysis GRID development monitoring UNIX sys admin, farming etc.	? ?

Glasgow group resources: 64-node farm, available for gsim + CLAS12 simulations,
local infrastructure for VME, EPICS etc.

We'd like to contribute to online software. What can we do?

Rough sketch of the current online tools



~~non-EPICS~~

DataBase

EPICS

Slow

CLAS

Fast

CODA

Event Transfer

MEDM

StripTool

Others

1st level of control and monitoring

Various

CED

online recsis

Sergey Root

DataBase

Monitor plots

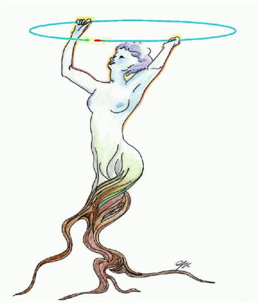
Improve / replace / extend now for CLAS -> CLAS12

... Sergey asked for volunteers last June

- A ROOT- based presenting system (Ken)
- Program implementation of a monitoring system (Dan)

} **Aggregation**

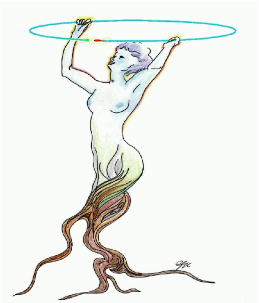
What role could ROOT play ?



MEDM	**ROOT**
Very robust	Quite flaky
Mainly for channel access	Channel access needs to be added
Easy drag & drop GUIs	A pain to make GUIs
Alarms built in	Alarms need to be added
No complex tasks, non-extensible	Do all sorts of fancy stuff. (example)
StripTool	
Very nice strip charts (EPICS)	Strip charts need to be added
CED	
Great tool, hard-wired for CLAS	Eek! Import Geant4 Geometry and display events? Look around.
Online analysis	
Sergey's ROOT tool + CLAS ET	Could be generalised in new framework
Online recsis + monitor plots	Will die with CLAS Need improving NOW

DANGER: Should not attempt to ROOTify everything – use what's most appropriate.

ROOT can aggregate data from diverse sources

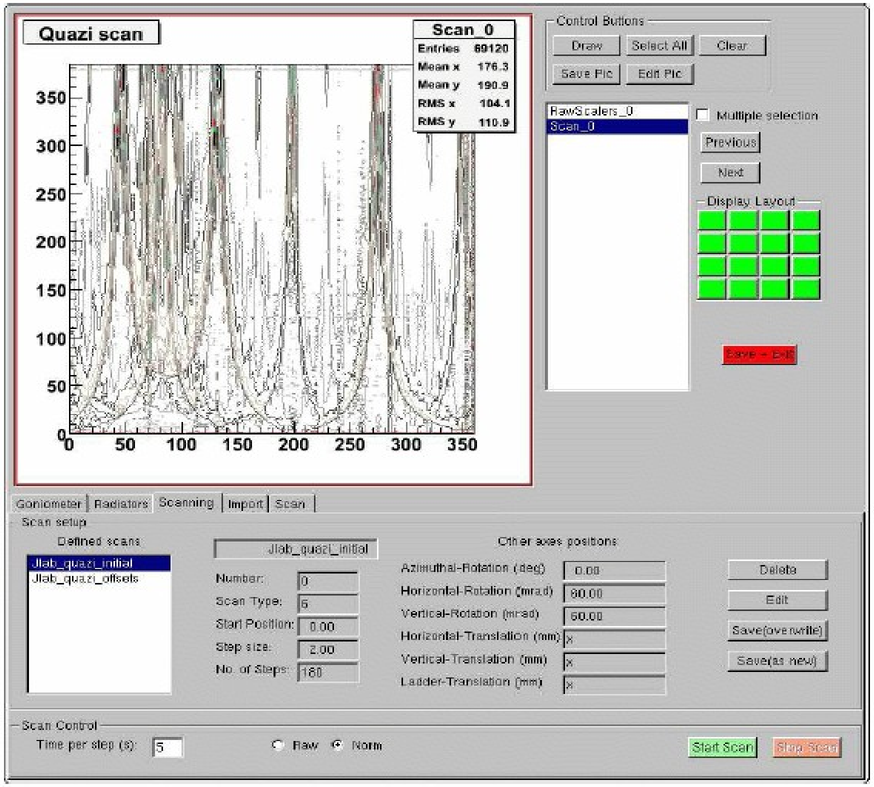


Good

Fancy plotting & analysis
Sophisticated GUIs
Access to event data (ET)
Access to EPICS (libEzCA)
Access to databases

Bad

This application required many weeks of coding and debugging!



Scan_0
Entries 69120
Mean x 176.3
Mean y 190.9
RMS x 104.1
RMS y 110.9

□ Multiple selection

Scan setup

Defined scans: Jlab_quazi_initial, Jlab_quazi_offsets

Jlab_quazi_initial

Number: 0
Scan Type: 6
Start Position: 0.00
Step size: 2.00
No. of Steps: 180

Other axes positions
Azimuthal-Rotation (deg) 0.00
Horizontal-Rotation (mrad) 60.00
Vertical-Rotation (mrad) 60.00
Horizontal-Translation (mm) x
Vertical-Translation (mm) x
Ladder-Translation (mm) x

Delete
Edit
Save(overwrite)
Save(as new)

Scan Control
Time per step (s): 5
○ Raw ● Norm

- ROOT can incorporate features of all the other tools together with its own capabilities and is extensible.

- Need to develop a framework which allows users to develop applications quickly.
  - ➔ Applications which are robust, and don't crash frequently!

Online Monitoring with MonALISA

Dan Protopopescu

MonALISA is a distributed service able to:

- collect any type of information from different systems
- analyze this information in near real time
- take automated decisions
- optimize work flows in complex environments

Dan has already set up MonaLISA in Glasgow for AliEn:
GRID Computing for PANDA project at GSI.

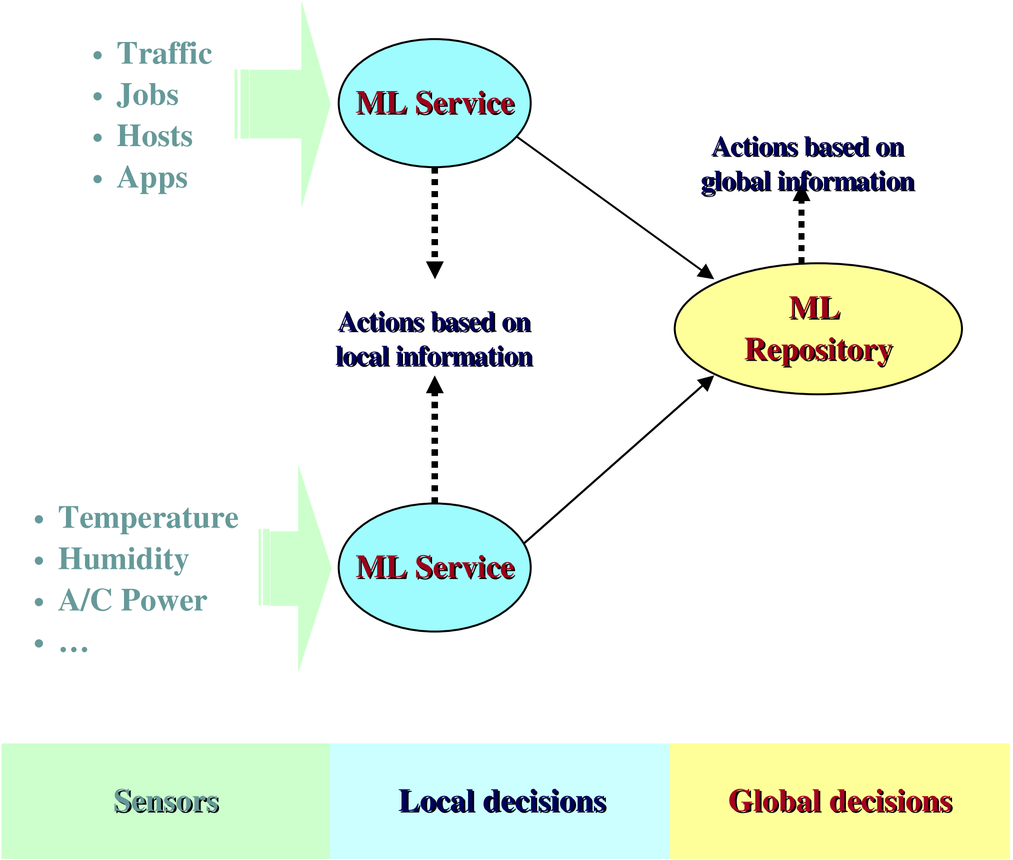
Main url: http://monalisa.caltech.edu

Lets see a demo: http://panda.gla.ac.uk:8999

Slide from talk given in Glasgow by Costin.Grigoras@cern.ch
http://nuclear.gla.ac.uk/data/MonALISA.pdf

Actions framework

- Based on monitoring information, actions can be taken in:
  - ML Service
  - ML Repository
- Actions can be triggered by:
  - Values above/below given thresholds
  - Absence/presence of values
  - Correlations between several values
- Possible actions types:
  - Alerts
    - e-mail
    - Instant messaging
  - External commands
  - Plain event logging
  - Annotation of repository charts with each event



Advantages

- MonALISA is simple to install
- ApMon daemons run on individual nodes monitoring sys. and other optional parameters via C, C++, Java, Python and Perl APIs
- Service runs on one computer which receives and aggregates all data
- Several repositories can exist (local and remote)
- WebService on server accessing DB, or elsewhere via repository
- ROOT plugin allows macros to send to MonaLISA
- Idea is fully supported by development team
- Some benchmark figures for the service:
  - ~ 800k monitored parameters at 50k updates/second
  - > 10k running (alien) jobs monitored simultaneously
  - > 100 WAN links
- ALICE uses ML for monitoring online reconstruction

Conclusion

General

- Find balance between hard-wired hacking and complete abstraction
- Use existing, robust, tools and frameworks wherever possible.
- Provide many example applications and good documentation

Proposed developments, starting as soon as possible

ROOT

- Make a tool to usefully compare the monitor plots (already half done)
- Create an EPICS library with EPICS "widgets" derived from ROOT "widgets"
  - ➔ Investigate a MEDM .adl file parser as a simple drag & drop ROOT GUI maker.
- Create a general stripChart class
- Examples with documentation using all these features + ET analysis -> MonaLISA

MonaLISA

- Set up a MonaLISA server to try it out
- Install ApMon on CLONs, begin with simple EPICS channels
- Set up some examples of actions based on aggregated data