HPAD meeting: DAQ and Control

C.Youngman for the XFEL DAQ and control group (self and S.Esenov)

christopher.youngman@desy.de

Outline

- DAQ architecture
- Train builder status
- DAQ software
- Position advert
- Conclusions

DAQ architecture: preamble

- DAQ architecture = from detectors to archive silo
- Only limited understanding of data rates and sizes, maximums:
  - **HPAD: ~400 frames / train @ 2MB / frame**
  - **LPD and DEPFET: ≤512 frames @ 2MB / frame**
  - **No useable number on frame/train reject or size reduction algorithms**
- DAQ multi layer architecture required, ingredients:
  - **All layers must handle full bandwidth**
  - **A data cache must sink data before commit to archive silo**
  - **Should foresee data rejection and size reduction processing in all layers**
    - **in pre data cache layers in real time, and**
    - **“offline” at the data cache to taking advantage of expt. off time**
  - **Must be scalable for 2, 4 … Mpixel detectors**
  - **Final layers must be tuneable in size and performance**

DAQ architecture: protocols

- Measurement of UDP % frame losses
  - **4 client-server pairs sending UDP frames concurrently**
  - **all clients and servers on separate host**
  - **each host connected point-to-point to Cisco switch**
  - **1GE links**

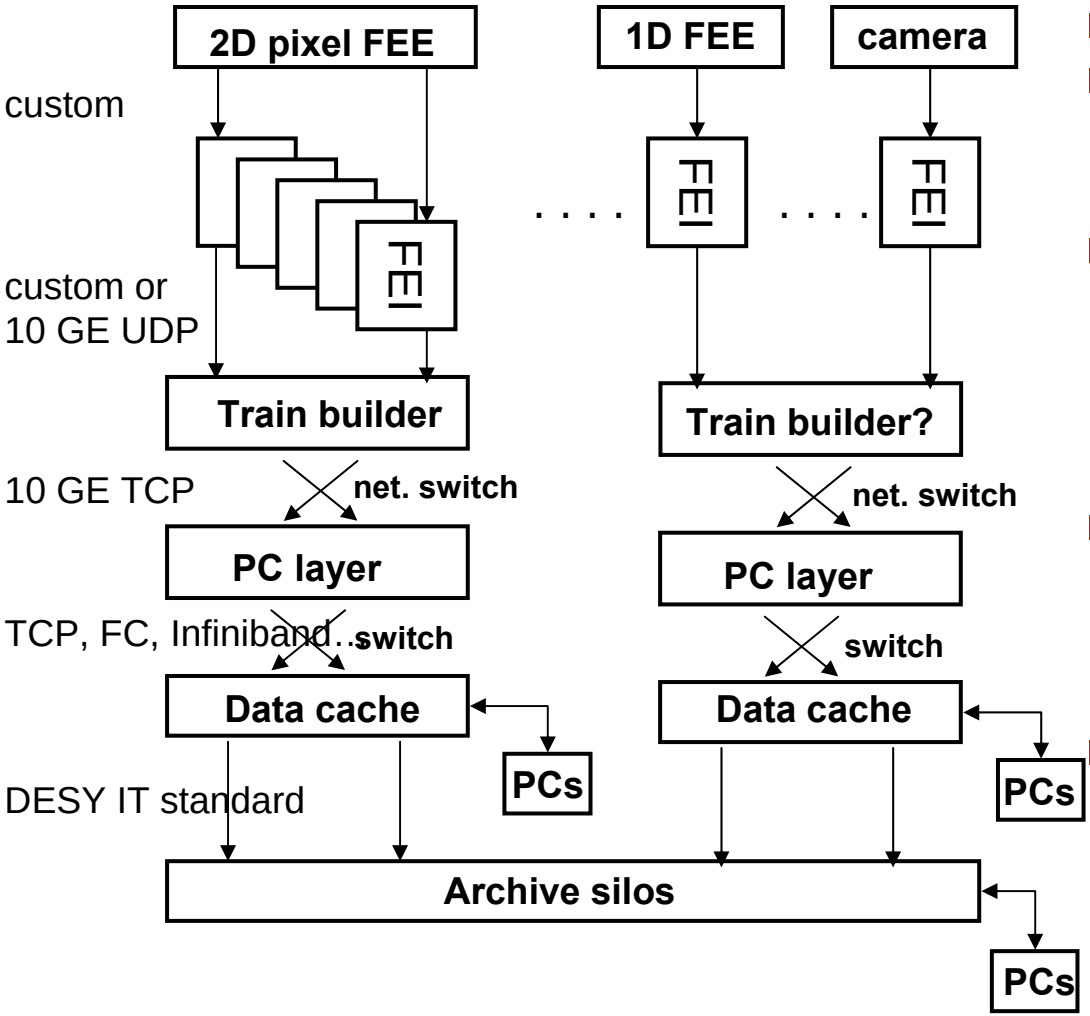
Pair:	1	2	3	4
Test config 1	0.0013	0.0068	0.0067	0.0040
Test config 2	0.0	0.0021	0.0	0.0053
Test config 3	0.0	0.0001	0.0056	0.0

% frame loss

- Observations
  - **client-server bandwidth = 95.7% of 1GE**
  - **user/system CPU usage %, client = 11/19, server = 17/22**
- 0.002% frame loss = 1 frame in 10 has missing data somewhere.

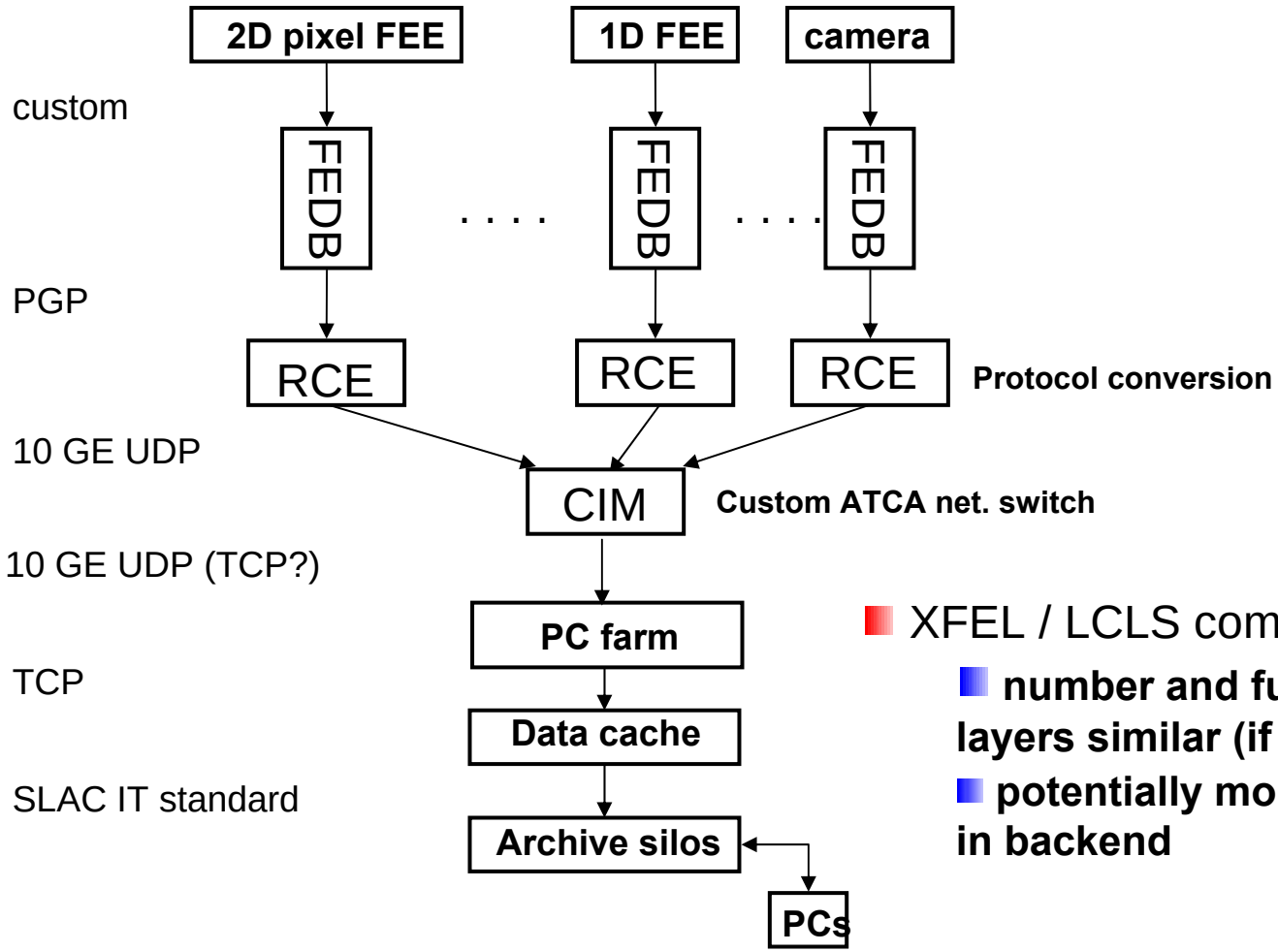
The measurements may be simplistic, but it's why I do not like UDP input over switches.

DAQ architecture: XFEL solution



- Front End Electronics (FEE)
- Front End Interface (FEI)
  - interface to Train Builder.
  - maybe not needed by 2D
- Train builder layer
  - builds trains
  - simple data processing
  - maybe not needed by 1D and cameras.
- PC layer
  - interface to cache
  - additional train building ?
  - more complex data process
- Data cache
  - hold, analyze, reduce and reject data
  - post processing commit to silo

DAQ architecture: LCLS solution



- XFEL / LCLS comparison
  - **number and functionality of layers similar (if not identical)**
  - **potentially more resource sharing in backend**

Train builder status:

- XDAC committee recommend global backend solution
  - HPAD proposed design: time sliced UDP to PC farm frame builder.
  - LPD EoI design: ATCA frame builder, TCP(?) out
  - DEPFET proposed design: custom into ePCI ATLAS PC boards, TCP out

- Workshop developments
  - DAQ and control group not happy with large number of connections from PC farm solution (originally 96 1GE links per 1Mpixel) if 4-8 Mpixel detectors eventually appear. Prefer a hardware switch builder (like ATCA) extended to train building.
  - Workshop specified 10GE as standard.
  - RAL group asked to prepare a straw man design (J.Coughlan) of train builder
  - Discuss straw man design when ready and find a global solution
- Straw man design now ready
  - discussion of HPAD, LPD, DEPFET and other interested parties at 25.6.2008 meeting at DESY.

Train builder status: Straw man design

- Quick look at Straw man design
  - Can just handle 512 (-few percent) 2MB frames /train (power worries, etc.)
  - Design has 10 inputs and outputs, 8 or 16 better match to FEEs
  - Could remove the 2D FEI if detector FEE inputs can be matched to TB

Processing mezzanine

10 out

Crosspoint switch

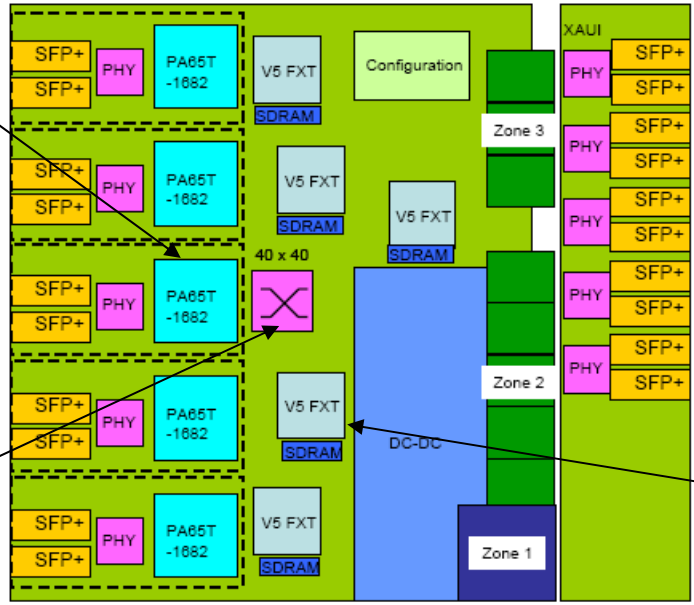


Figure 1 Straw-man ATCA Train Builder with Data Processing Engines

10 in

FPGA input processing

Use meeting to readdress common signal interface – initial discussions inconclusive.

DAQ software: using DOOCS at XFEL

- March 2008 workshop baseline: use DOOCS for control software.

- Have tested simple DOOCS data server (RC, FSM, monitoring…) – setup:
  - **started from scratch with our choice OS (OpenSUSE not Solaris or Debian.)**
  - **not a (simple) binary installation = build from sources.**
  - **installed own Oracle DB and populated from main DOOCS DESY DB.**
  - **performed run control transitions and monitoring using RC GUI.**
  - **much help received from V.Rybnikov (DOOCS RC/data server expert)**

- Experience gained – not quite what we expected:
  - **code management: CVS used, but no tags(?) = head only source builds.**
  - **configuration: RC from DB, services location from text files = not easy.**
  - **Significant expert interaction and time required to get test running.**

- Conclusion – tedious and expert intensive, what to do:
  - **spend next 4-5 weeks working out what implementation we want**
  - **then discuss with DOOCS how to proceed.**

DAQ software: ideas being looked at

- Code – DOOCS is a mixture of C++ plus Java.
  - **Use Java and interface to non Java applications using defined APIs**
  - **Use packages and tools, writing as little in house code as possible**
- Management – standard issues
  - **Use IDEs where possible.**
  - **Use SVN (or CVS) with tags for version control.**
  - **Use defined backup policy (Tivoli).**
- Packages and tools to look at – initial list
  - **Framework = Spring (object container with XML object wiring, DI, AOP…)**
  - **Logging = Log4J**
  - **DB = JDBC, Hibernate, iBATIS or JDO, and plain XML**
  - **Naming = LDAP or RMI over JNDI**
  - **Messaging = JMS, SOAP…**
  - **GUI = Matisse, FLEX…**
  - **Glassfish = application server**
- Observations – expect frequent changes
  - **Many of these packages are new to us**
  - **Objective is simplicity and user friendliness**

Position advert:

- Need extra manpower in DAQ and Control group
  - **Currently 2, myself and S.Esenov.**
  - **Now looking for a more hardware related individual with following profile**
    - **Enjoys working at the software / hardware interface**
    - **Has lots of expeience**
    - **Works long hours efficiently**
    - **… If you know a candidate please tell me !**

Conclusions:

- A preliminary DAQ architecture design now exists.
- Try to agree to train builder solution at 25.6.2008 meeting
- Address the common control signal (timing, clocks, vetos…) at meeting.
- Have built and tested DOOCS RC system.
- Need to prepare and make concrete requests to DOOCS group about future development.
- Additional manpower needed
- Need to review the HPAD / DAQ commitments schedule w.r.t. slow control type (HV and LV) currently scheduled for 2009.