LCLS Photon Controls and Data Systems

SwissFEL ARAMIS Workshop

Amedeo Perazzo
June 19th, 2012

SLAC NATIONAL ACCELERATOR LABORATORY

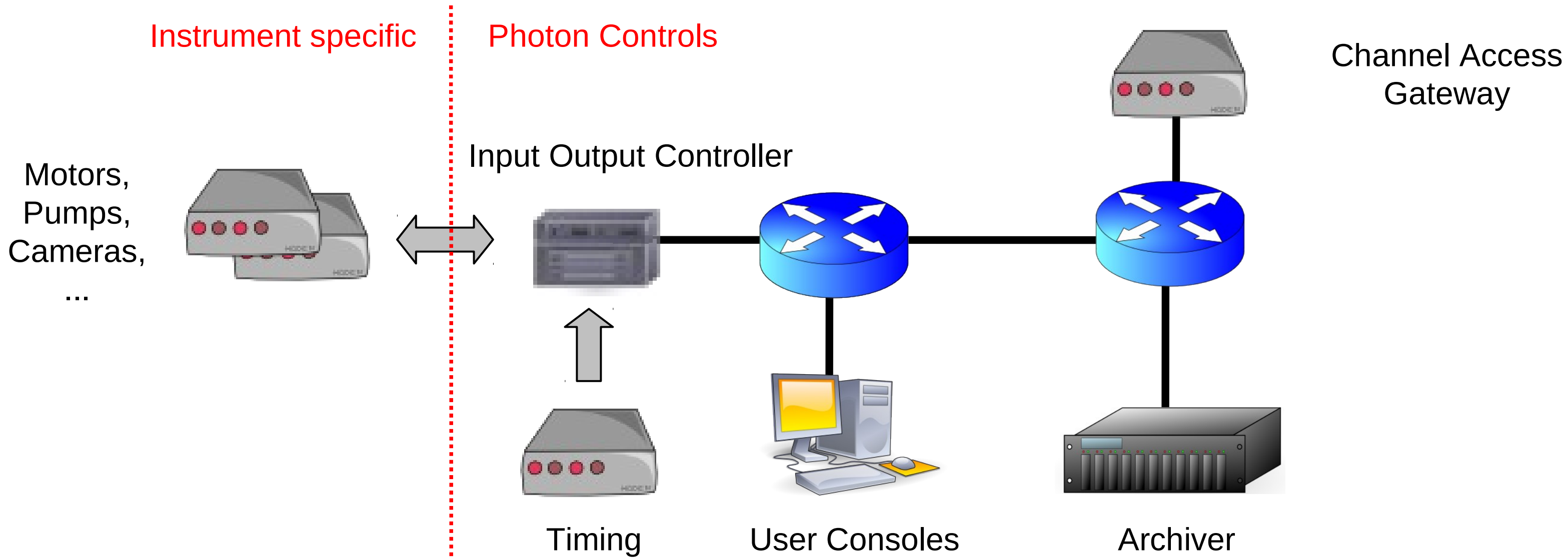
Key Challenges

- Ability to readout, event build and store more than 1GB/s data @ 120 Hz pulse rate
- Build reliable, fast, user friendly network which complies with DOE security requirements
- Allow experimenters to analyze data on-the-fly
- Flexibility to accommodate user supplied equipment
- Ability to store and analyze very large data sets

Control System

- **Each instrument has a dedicated controls network**
  - The front-end enclosure (FEE), the x-ray tunnel (XRT) and the laser system also have dedicated controls networks
- **Each network is built around:**
  - a few 1Gbps high performance edge switches (hutch)
  - operator consoles (control room)
  - variable number of IOCs (typically between 10 and 30 per hutch)
- **An additional isolated controls subnet, accessible from the console nodes, provided for user supplied equipment**
- **LCLS controls system is based on EPICS framework**

Control System Architecture



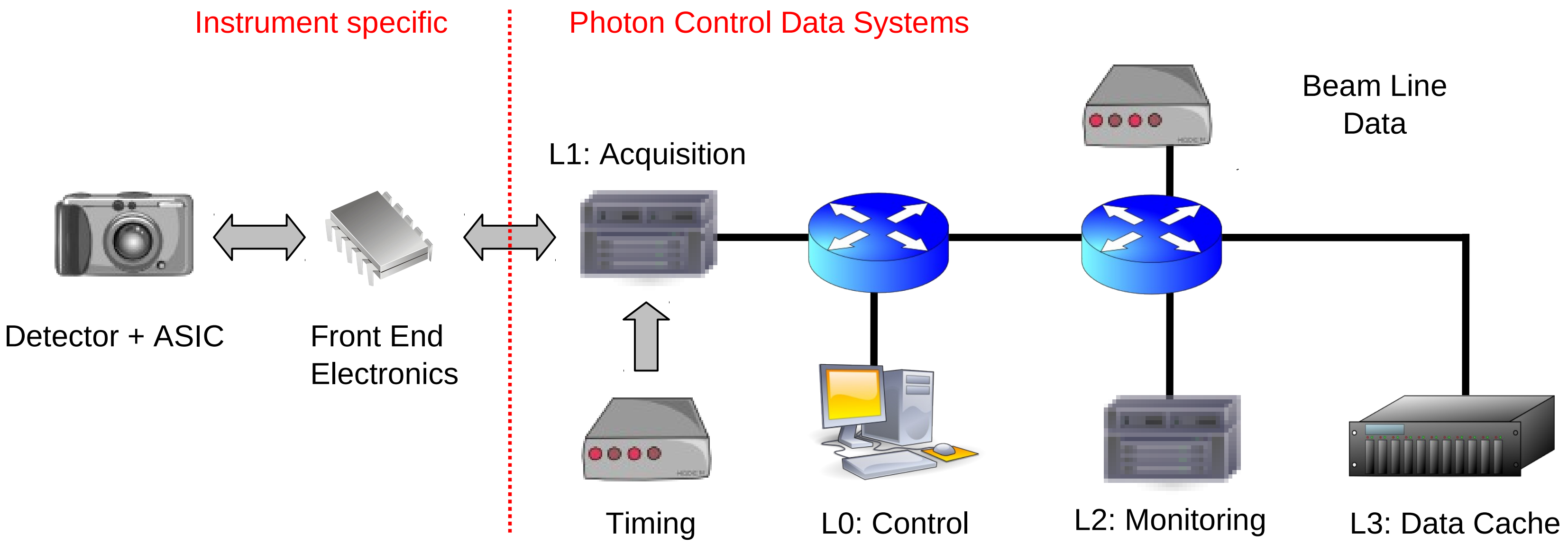
Channel Access Gateway & Archiver

- **Controls traffic among the different networks, and with the accelerator, managed through a channel access gateway**
  - goal of the gateway is twofold:
    - limit the load on each IOC and
    - allow different access rules from different nodes
  - latter required to guarantee safe operations while allowing read access to all operators
- **Selected EPICS process variables from all control areas are saved to dedicated server**
  - one archiver engine for each instrument

Data Acquisition

- **Each instrument has dedicated DAQ network**
- **Each network built around:**
  - one 1Gbps high performance edge switch (hutch), one dedicated 10Gbps switch (server room)
  - consoles for DAQ operators (control room)
  - variable number of readout nodes (typically between 10 and 20 per hutch)
  - monitoring nodes, data cache nodes, fast feedback analysis node (server room)
- **DAQ can currently acquire up to 2GB/s without introducing dead-time in the system**

DAQ Architecture



Monitoring

- **Online monitor framework allows users to analyze, on the fly, the quality of the data**
- **Implemented by snooping on the DAQ traffic between the readout nodes and the data cache nodes**
  - Guarantees that monitoring does not impact data acquisition
- **Users can augment the existing monitoring features by dynamically plugging in their code to the core monitoring framework**

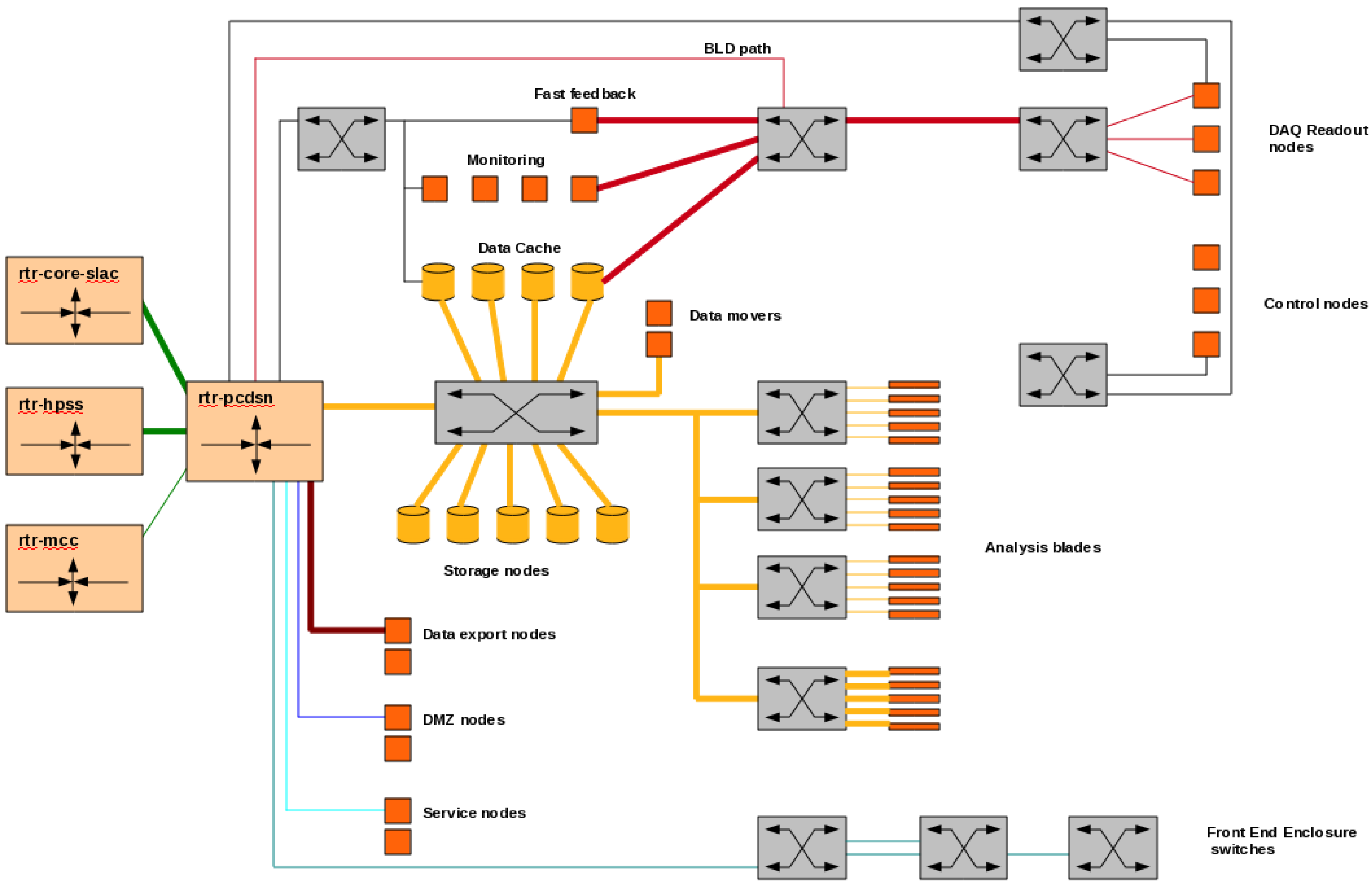
Online Data Cache

- **Data cache nodes:**
  - assemble the components from the different readout nodes which correspond to same pulse (event building)
  - store full event to the local RAID array
- **Data cache currently 200TB per instrument**
  - isolates DAQ system from users operations
  - allows experiments to take data even during outages of the offline system
- **Data files are copied over 10Gbps links from online cache to medium-term storage where they are made available to the users for offline analysis and for off-site transfer**

DAQ Interfaces

- **Controls: DAQ interfaces to controls in order to:**
  - store some user selected EPICS process variables together with the science data
  - control any device that can be used to perform a scan or a calibration run
- **Beam Line Data: DAQ receives small pieces of information which contain key beam measurements**
  - currently three packets per pulse:
    - e-beam parameters from accelerator, timing information from RF cavity, gas detector measurements from front-end enclosure
  - timestamped with the pulse ID and stored with the science data

Data Networks



Offline Analysis

- **Analysis system shared among the different instruments**
- **Main physical components of analysis system are:**
  - medium-term storage
  - long-term storage
  - processing farm
- **Analysis system also provides software frameworks to:**
  - copy the science data to medium and long term storage
  - translate the data into user formats (HDF5)
  - parse and analyze the data

Storage

- **Medium-term storage is disk based**
  - Current size 4 petabytes
  - Each PB has maximum aggregated throughput of 12GB/sec
  - Each client has throughput from 50 to 800 MB/s
- **Long-term storage uses tape staging system in the SLAC central computing facilities**
  - Can scale up to several petabytes
- **Science data files policies:**
  - Kept on disk for 1 year
  - Kept on tape for 10 years
  - Access to the data for each experiment granted only to members of that experiment

Data Retention Policies

Area	Size	Lifetime	Backup	Comment
xtc	Unlimited	1 year	Tape	Raw data
hdf5	Unlimited	1 year	Tape	Translated data (on demand)
scratch	Unlimited	1 year	None	Temporary area
User home	Unlimited	Indefinite	Tape and disk	User code, results
Tape	Unlimited	10 years	Dual copy	Raw data (can be restored to disk on demand)

Data Movers

- **Experimenters allowed to transfer their data files to their home institution if they decide to do so**
  - two data mover nodes allocated for that purpose
- **Disk storage communicates with**
  - tape staging system
    - dedicated dual 10Gbps links
  - SLAC main router for off-site data transfer
    - additional dual 10Gbps links

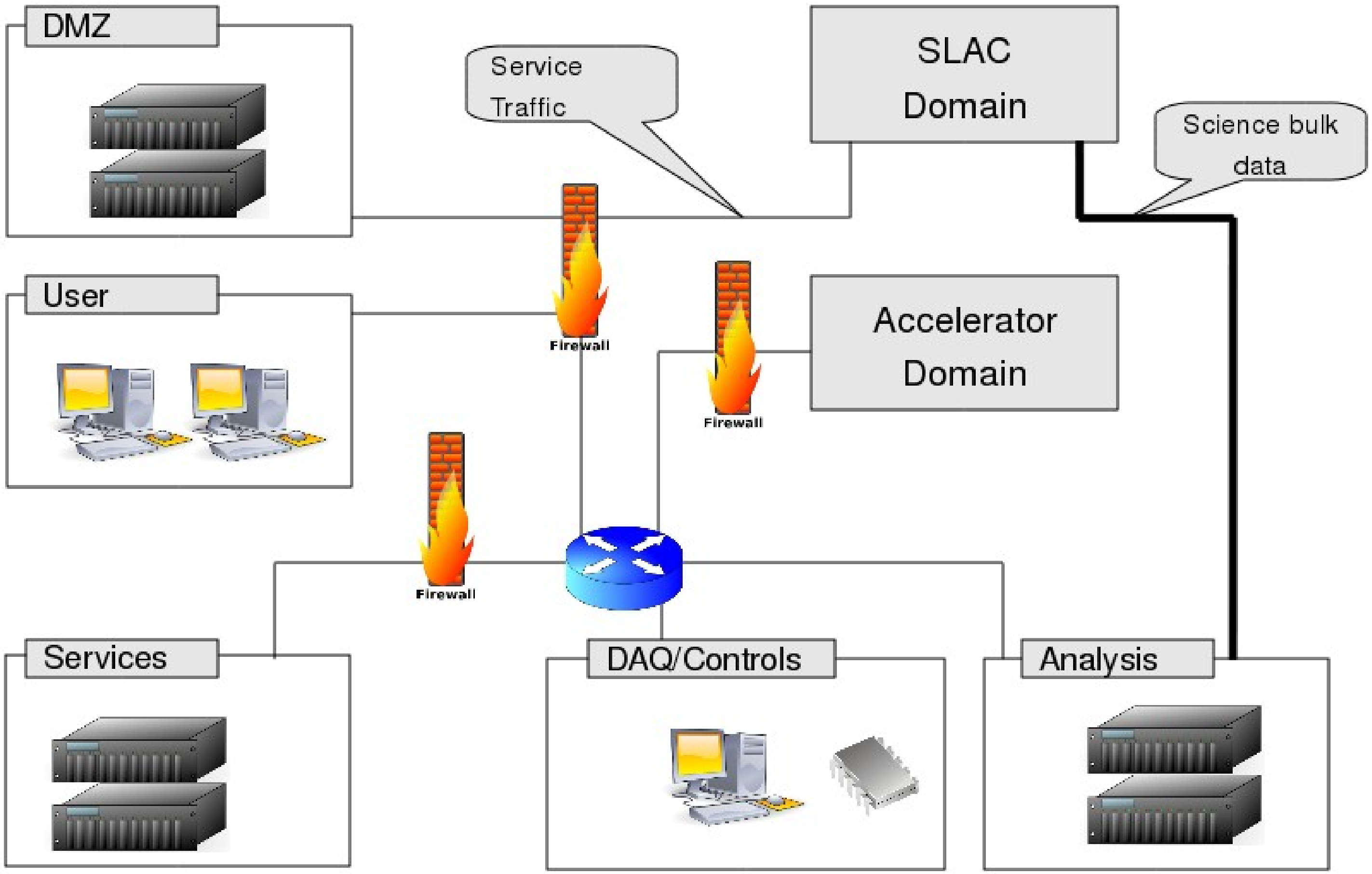
Processing

- **Processing farm based on:**
  - Batch pool: 1000 cores
  - Interactive pool: 192 cores
- **Farms live in the experimental areas with fast access to the science data files in medium-term storage**
  - Batch nodes: Infiniband QDR
  - Interactive nodes: 10Gb/s Ethernet

Control Room Services

- **In addition to console nodes, each instrument control room provides**
  - User workstations with access to online/offline systems and Internet
  - Printers
  - Wireless access to the visitor network
  - Taps for fast access to science data from users' computers
  - Various patch panels for hutch communication
    - BNC patch panel for coaxial cables
    - Fiber patch panel for SMFs
    - Patch panel for CAT6 cables

High Level Networking



Lessons learned (1)

- **Very hard to implement effective trigger/veto system**
  - Not a technical/computing issue: the ability to veto events is already implemented in the system
  - Vetoing based on beam parameters not effective (most pulses are good)
  - Hard to get help from users in setting veto parameters which define event quality
    - Users themselves often don't know what these parameters or their thresholds should be
    - Users are usually very suspicious of anything which can filter data on-the-fly
- **Benefit of vetoing events based on the event data is potentially very large**
  - factor 10-100

Lessons learned (2)

- **HEP style online/offline separation doesn't work**
  - The core online monitoring is not enough for many experiments
  - The skill level required to write on-the-fly analysis code is too high for most users
  - As a consequence some experiments feel they fly blind
- **Critical to provide users the ability to run offline style code for fast feedback**
  - Currently an issue for:
    - High data volume combined with low hit rate experiments: offline designed to keep up with DAQ only in average, not instantaneously; fast feedback nodes which look at subset of the data don't provide enough statistics
    - HDF5 based experiments: must wait for additional translation step

Lessons learned (3)

- **Plan to modify data retention policy with dual-fold goal: encourage users to filter their data and provide fast access to the data for longer period**

Area	Size	Lifetime	Backup	Comment
xtc	Unlimited	3 months	Tape	Raw data
ftc	20TB	2 years	None	Filtered, translated, compressed
tmp	Unlimited	3 months	None	Temporary area
res	1TB	2 years	Tape and disk	Analysis results
User home	20GB		Tape and disk	User code
Tape	Unlimited	10 years	Dual copy	Raw data (can be restored to disk on demand)

Lessons learned (4)

- **High fragmentation analysis tools adopted by users for data analysis**
  - psana (LCLS C++ framework), pyana (LCLS Python framework), Matlab, IDL, Igor, etc
- **Strong need of high performance, open source framework**
  - HEP community attempted something similar with ROOT, but was not fully successful
- **Should provide**
  - Algorithms needed by the photon science community
  - High quality and powerful plotting tools
  - Both scripting and compiled languages