

CMS DAQ

Today and the Future

Remigius K Mommsen
Fermilab

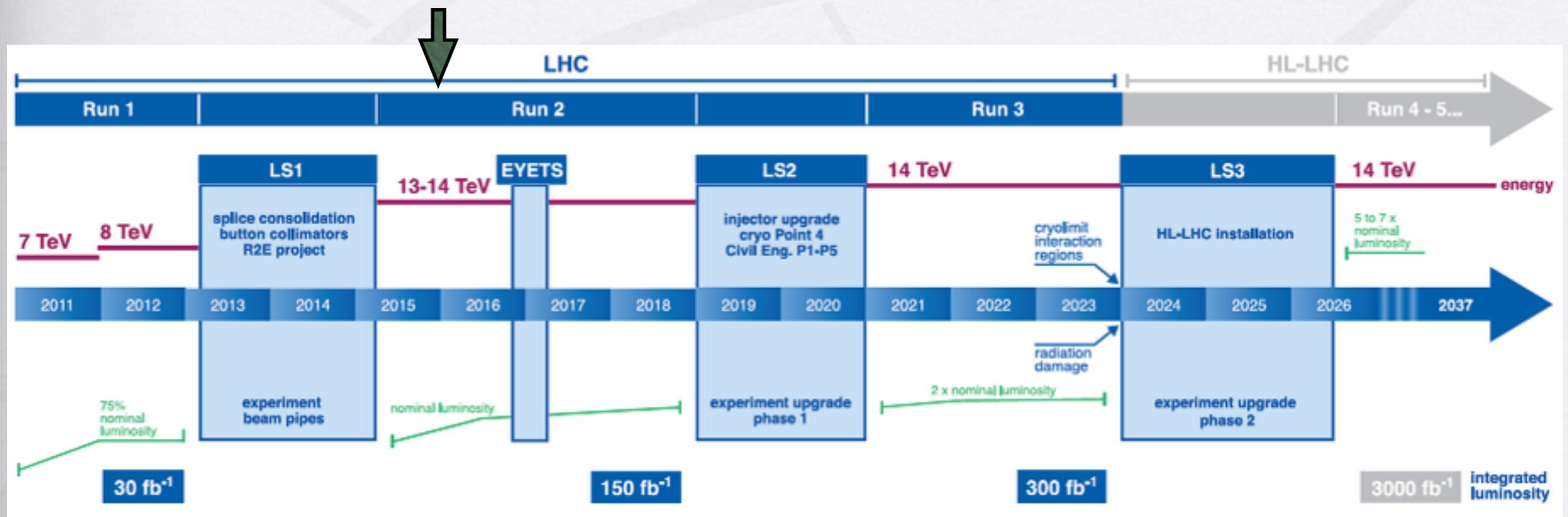
Overview

A new DAQ for run II

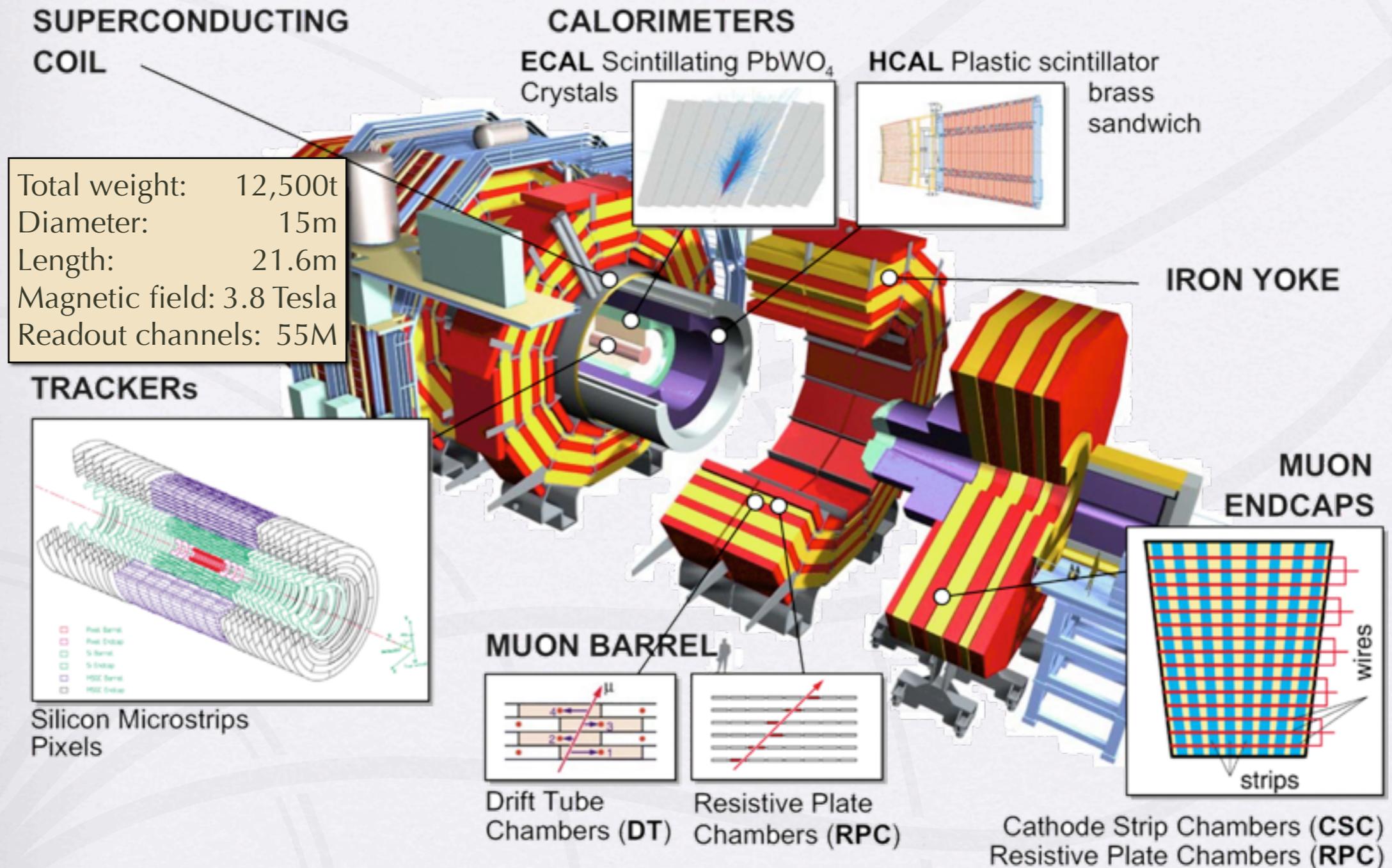
Plans for the near future

Ideas about DAQ3 (2019)

Data-acquisition on the time-scale of CMS phase 2 (2025)



Compact Muon Solenoid (CMS)



A New DAQ for LHC Run 2

Requirements

- 100 kHz level 1 trigger rate (unchanged)
- Event size might double to 1.5 MB
 - Increase in pileup
 - New detectors
- Accommodate legacy and new μ TCA-based detector readouts

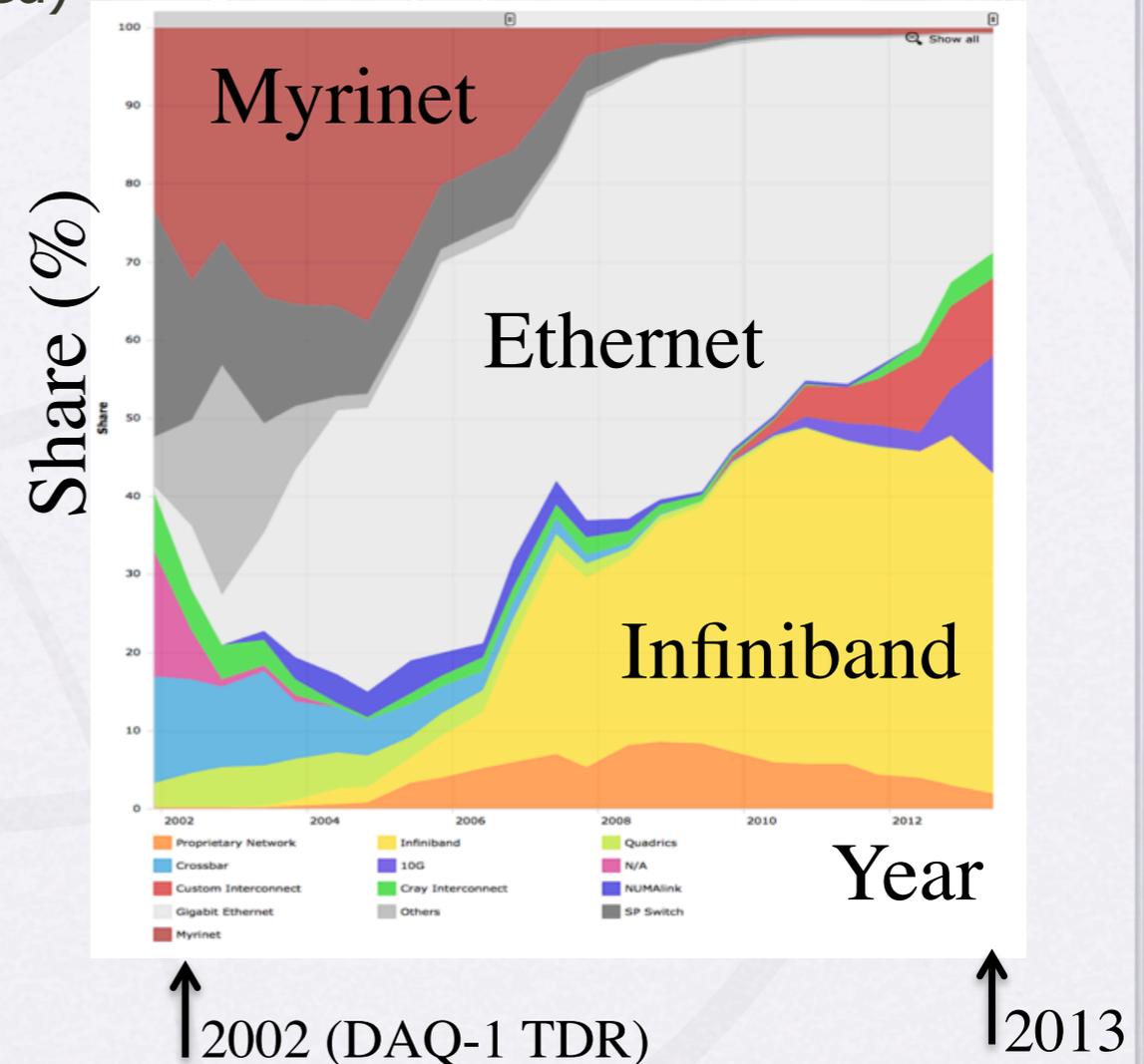
Aging hardware

- Most components of DAQ 1 reached end-of-life cycle

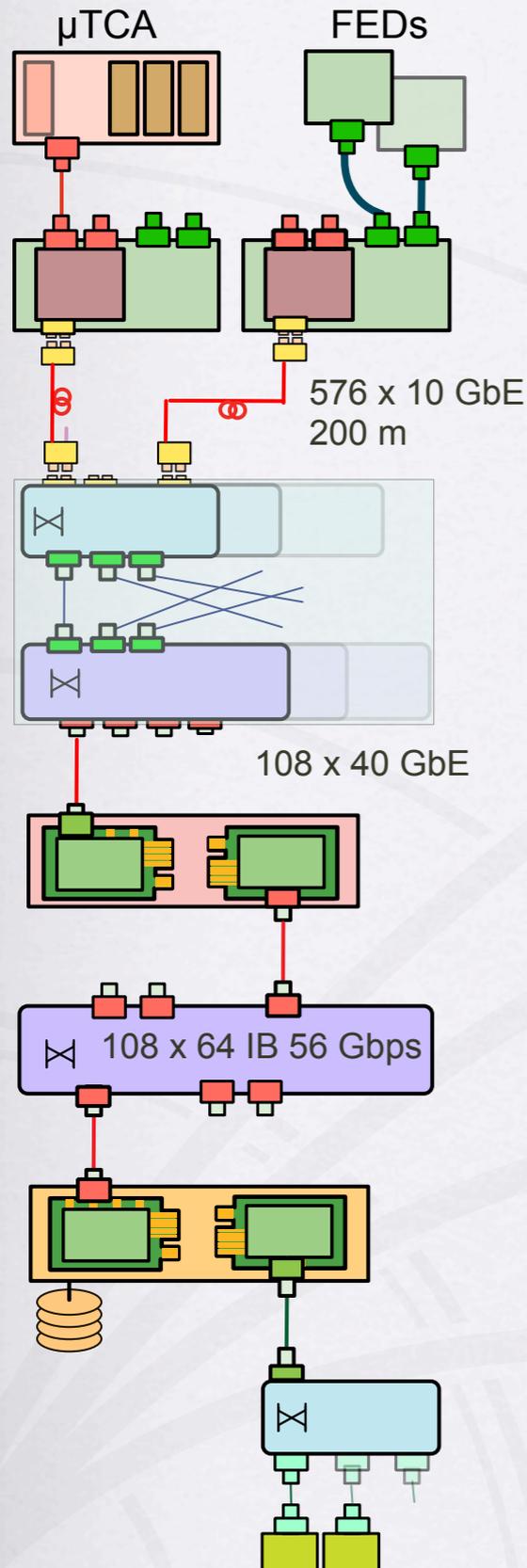
New technologies

- Myrinet widely used when DAQ 1 was designed
- Ethernet and Infiniband dominate the top-500 supercomputers today

Top500.org by Interconnect Family



CMS Event Builder



Detector front-end (custom electronics)

- ~700 front-end drivers (FEDs) with 1–8kB/fragment at 100 kHz

Front-End Readout Optical Link (FEROL)

- Optical 10 GbE TCP/IP

Data Concentrator switches

- Data to Surface
- Aggregate into 40 GbE links

Up to 108 Readout Unit (RU) PCs

- Combine 4-18 FEROL fragments into one super-fragment

Event Builder switch

- Infiniband FDR 56 Gbps CLOS network

Up to 64 Builder Unit (BU) PCs

- Event building
- Temporary recording to RAM disk

~900 Filter Unit (FU) PCs with ~16k cores

- Run HLT selection using files from RAM disk
- Select $O(1\%)$ of the events for permanent storage

10 GbE Replacing Myrinet

Front-End Readout Optical Link (FEROL)

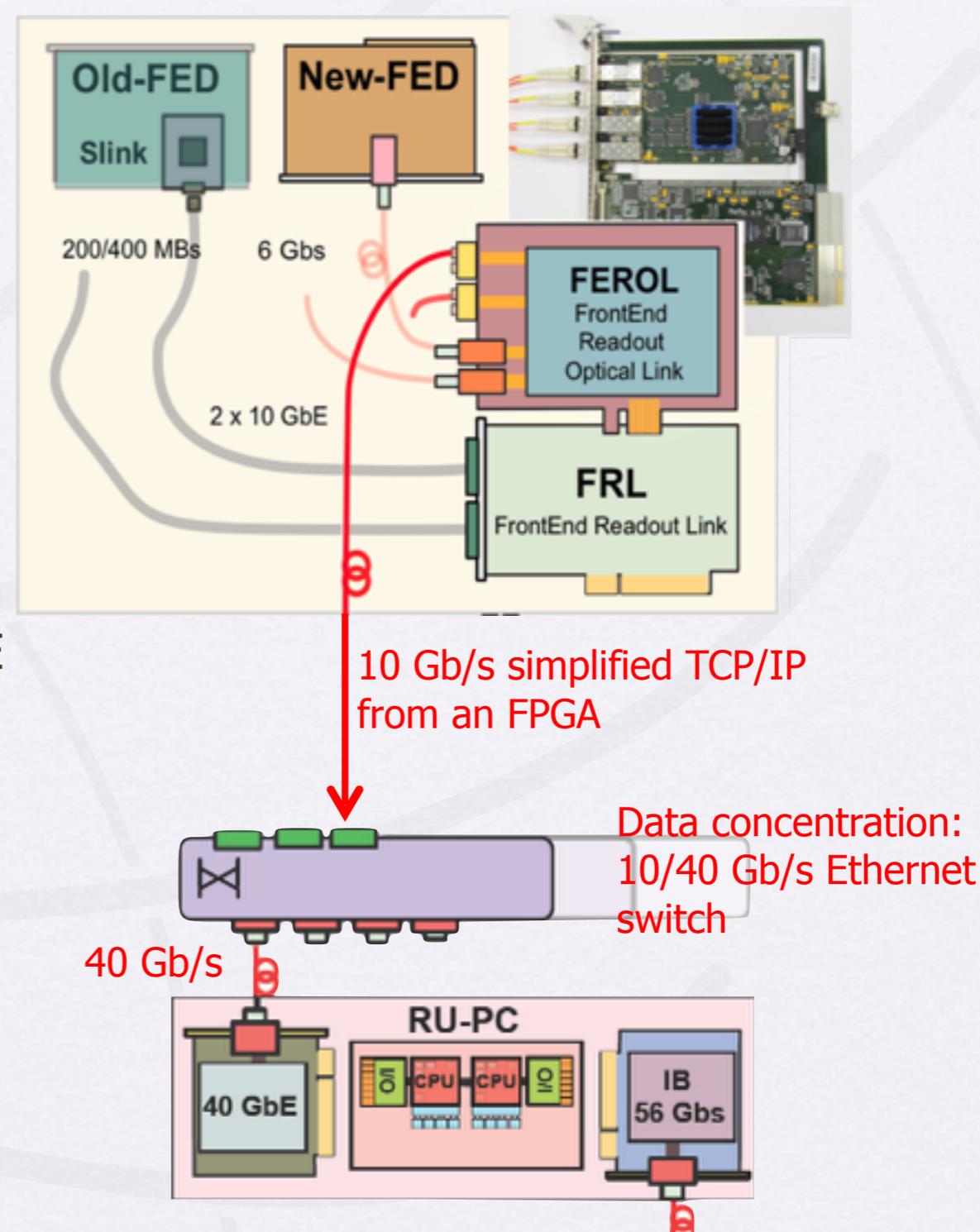
- ❑ Legacy input via Slink / FRL
- ❑ Optical up to 10 Gb/s from new μ TCA crate via AMC13

Data to surface

- ❑ Simplified TCP protocol over 10 GbE
- ❑ New fibers from USC to SCX
- ❑ 4-18 FEROL merged into 40 Gbit Ethernet at switch level

Each FED has one TCP stream

- ❑ Readout Unit (RU) builds super-fragments



TCP/IP from FPGA

Benefits of using TCP/IP

- TCP/IP guarantees a reliable and in-order data delivery
- Standard and well known protocol suite
- Implemented in all mainstream operating systems
- Debugging and monitoring tools widely available (tcpdump, wireshark, ...)
- Network composed from of-the-shelf hardware

In principle a very difficult task for an FPGA

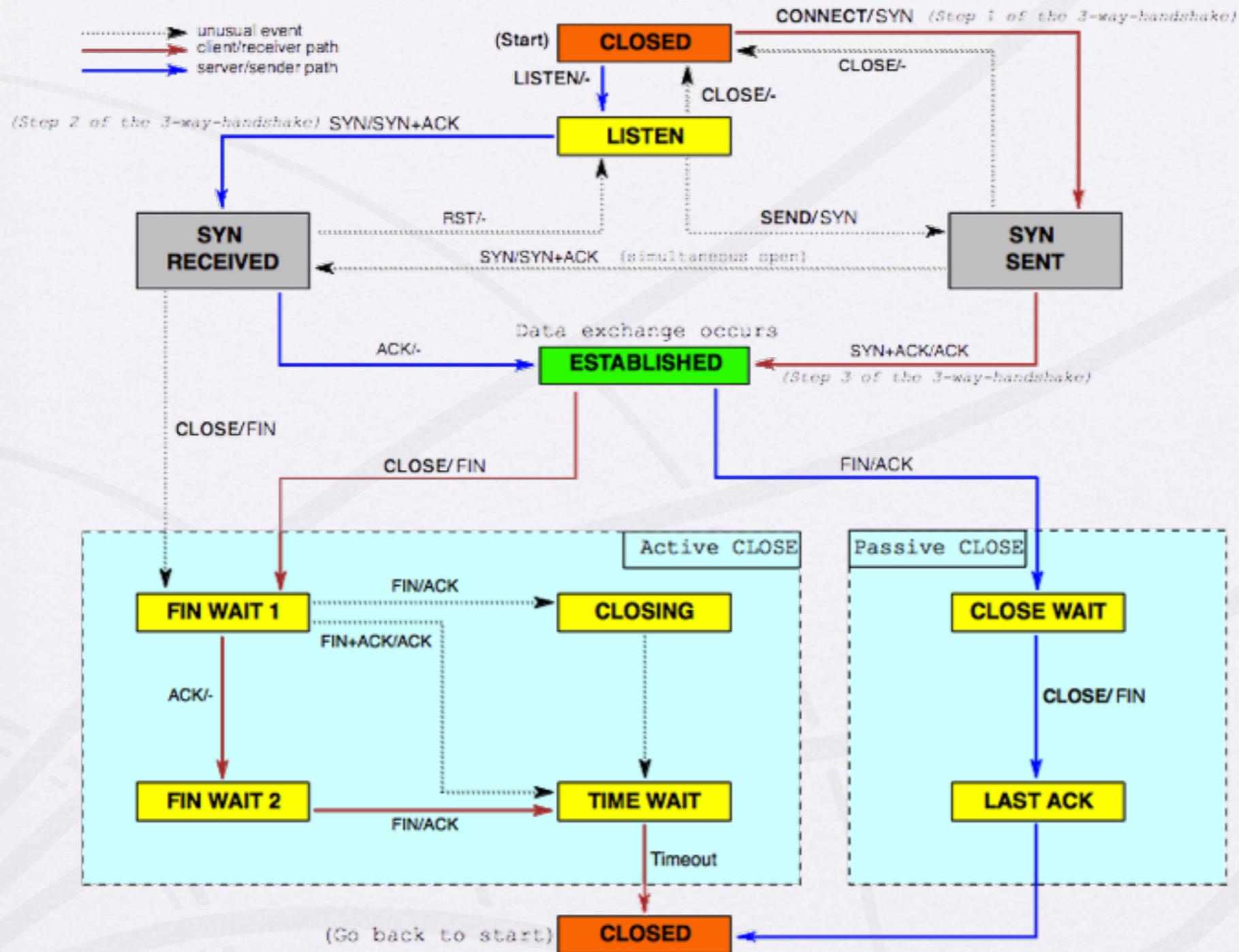
- Even for a PC the TCP/IP is a very resource-hungry protocol
- ~15,000 lines of C code in the Linux Kernel just for TCP

Consideration

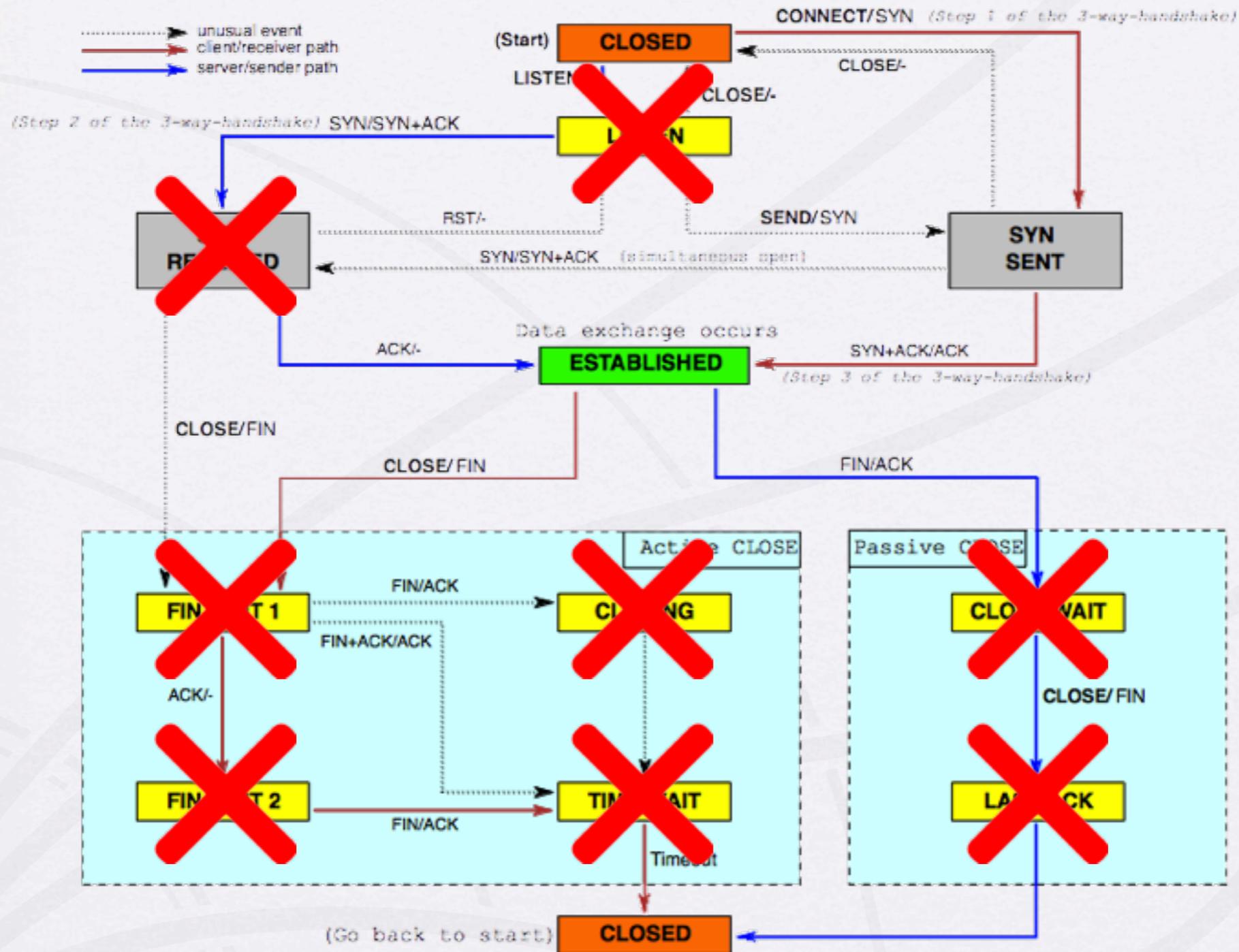
- CMS DAQ network has a fixed topology
- The data traffic goes only in one direction from FEROL to Readout Unit (PC)
- The aggregated readout network throughput is sufficient (by design) to avoid the packet congestion and packet loss

Can we simplify the TCP?

TCP State Machine

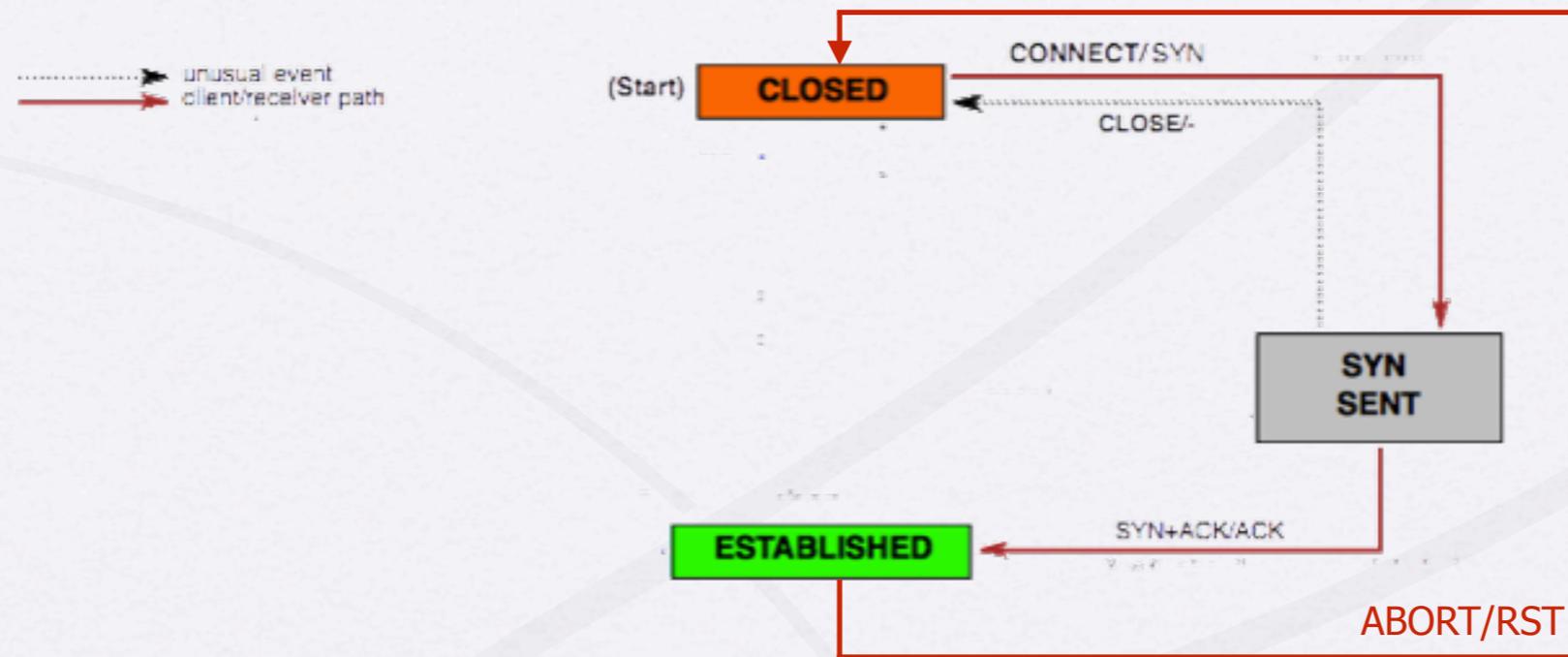


Simplifying TCP Implementation



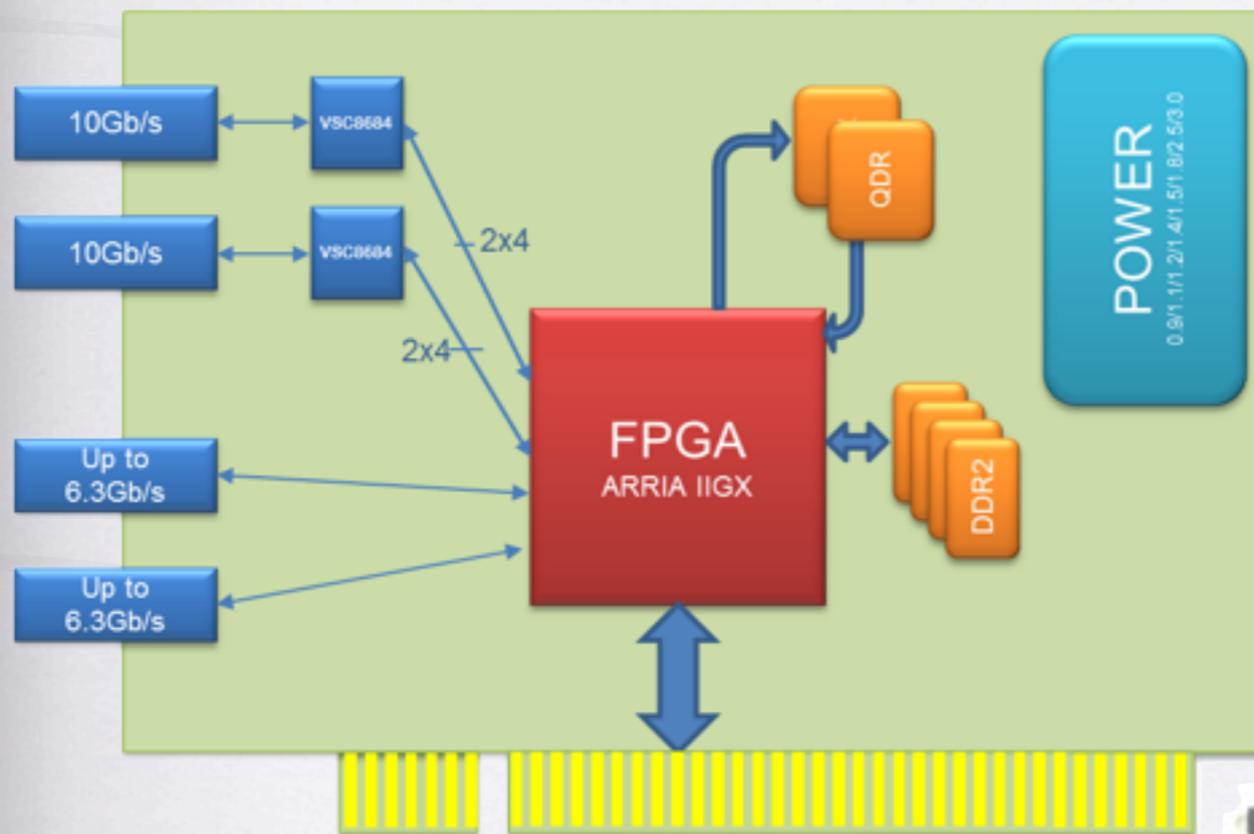
We do a connection abort instead of connection close

Final TCP implementation



Final State Diagram

FEROL Hardware Architecture

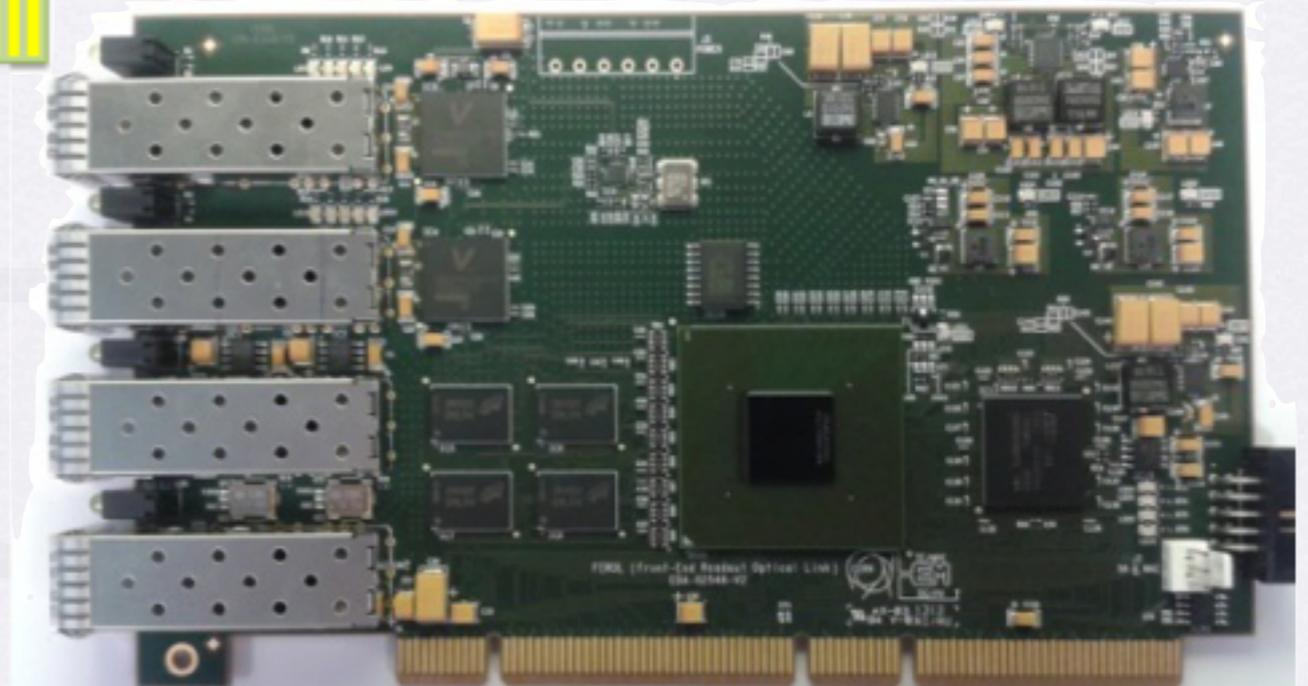


Hardware

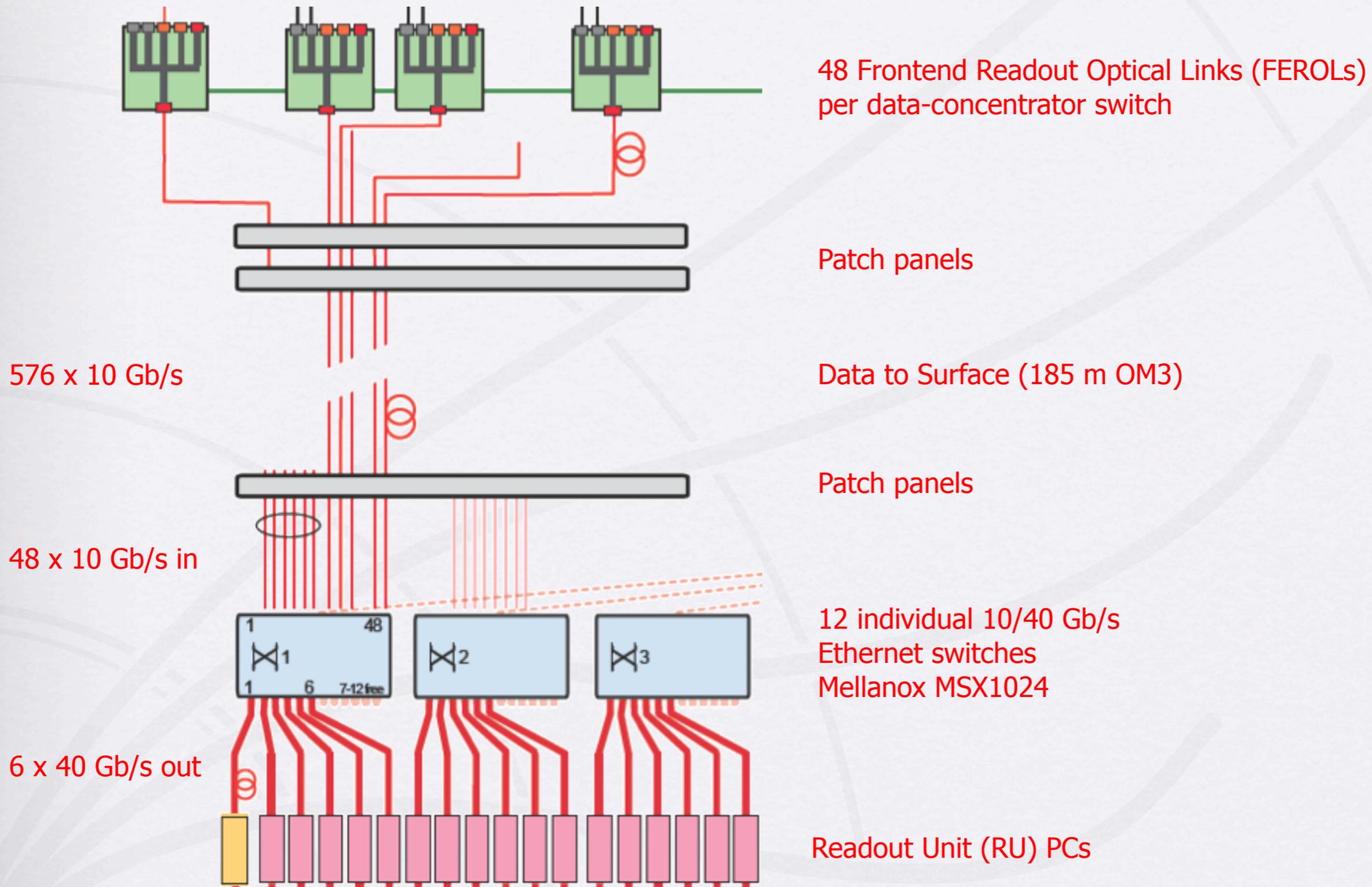
- Altera Aria II GX FPGA
- Vitesse transceiver 10GbE / XAUI
- QDR Memory (16 MBytes)
- DDR2 Memory (512 MBytes)

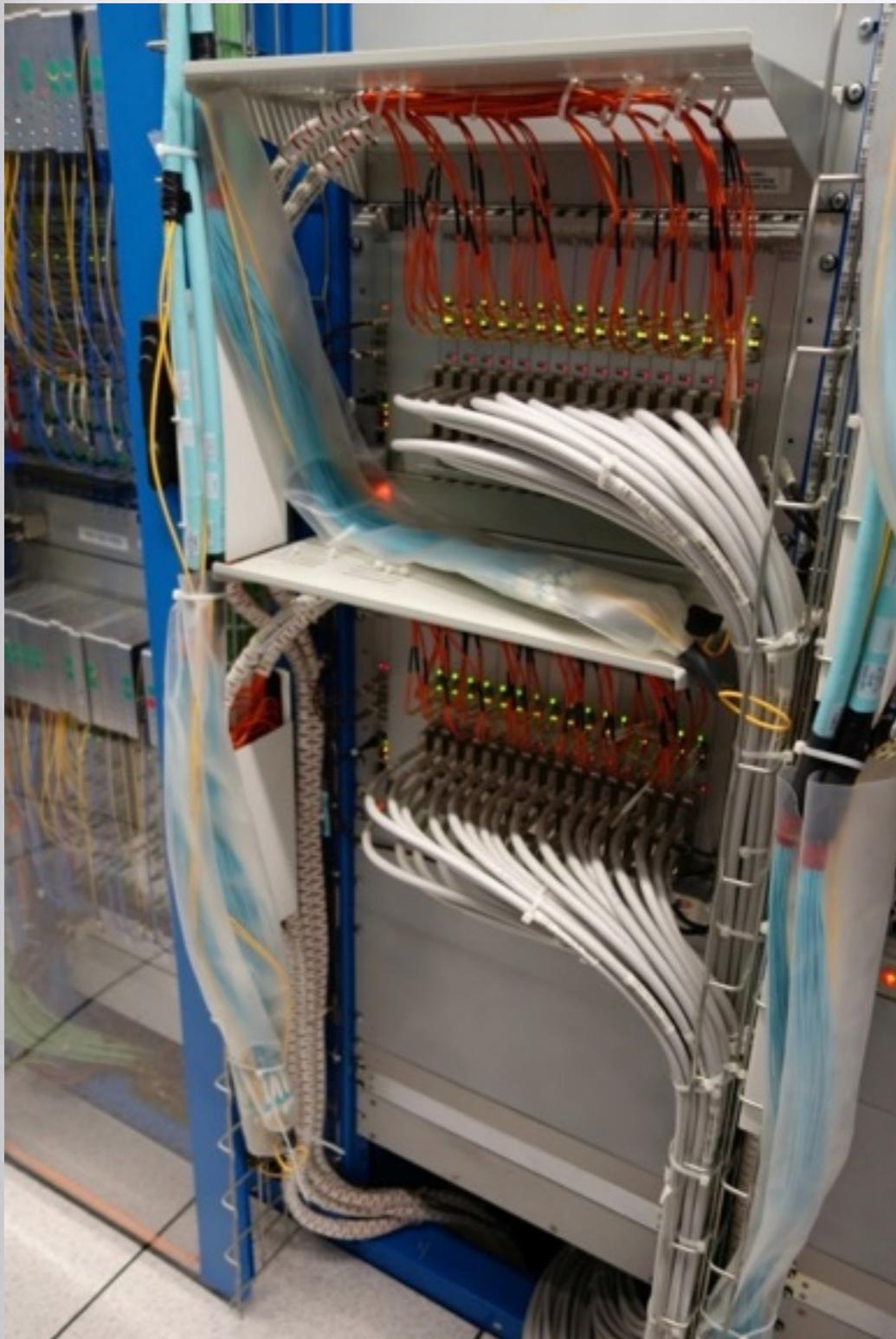
Interfaces

- FED/SlinkXpress interface
 - 2x optical 6 Gbit/s
 - 1x optical 10 Gbit/s
- DAQ interface
 - 1x optical 10 Gbit/s Ethernet

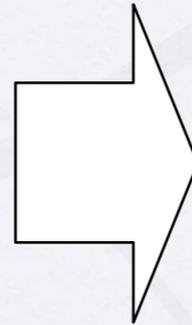


Data Concentrator



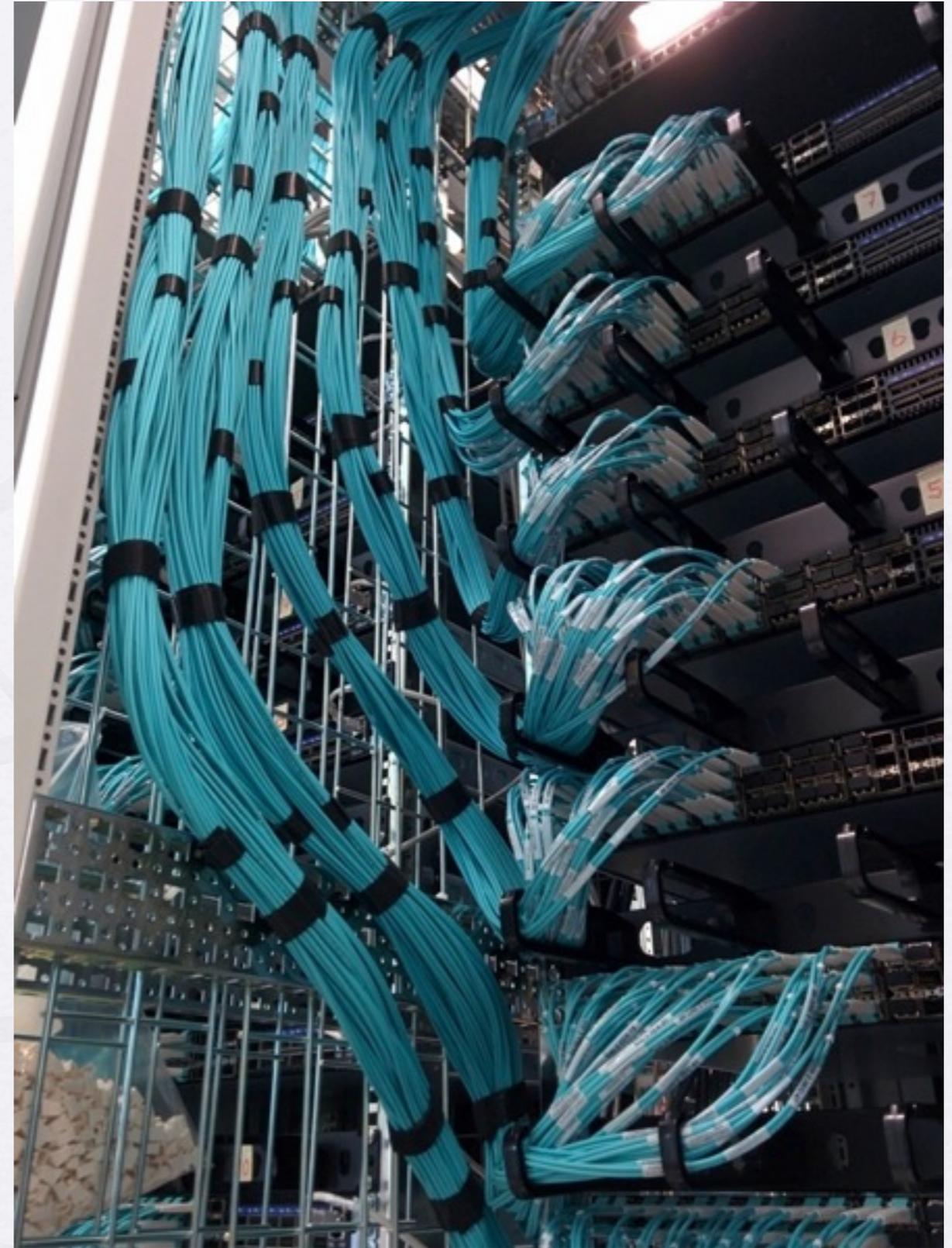
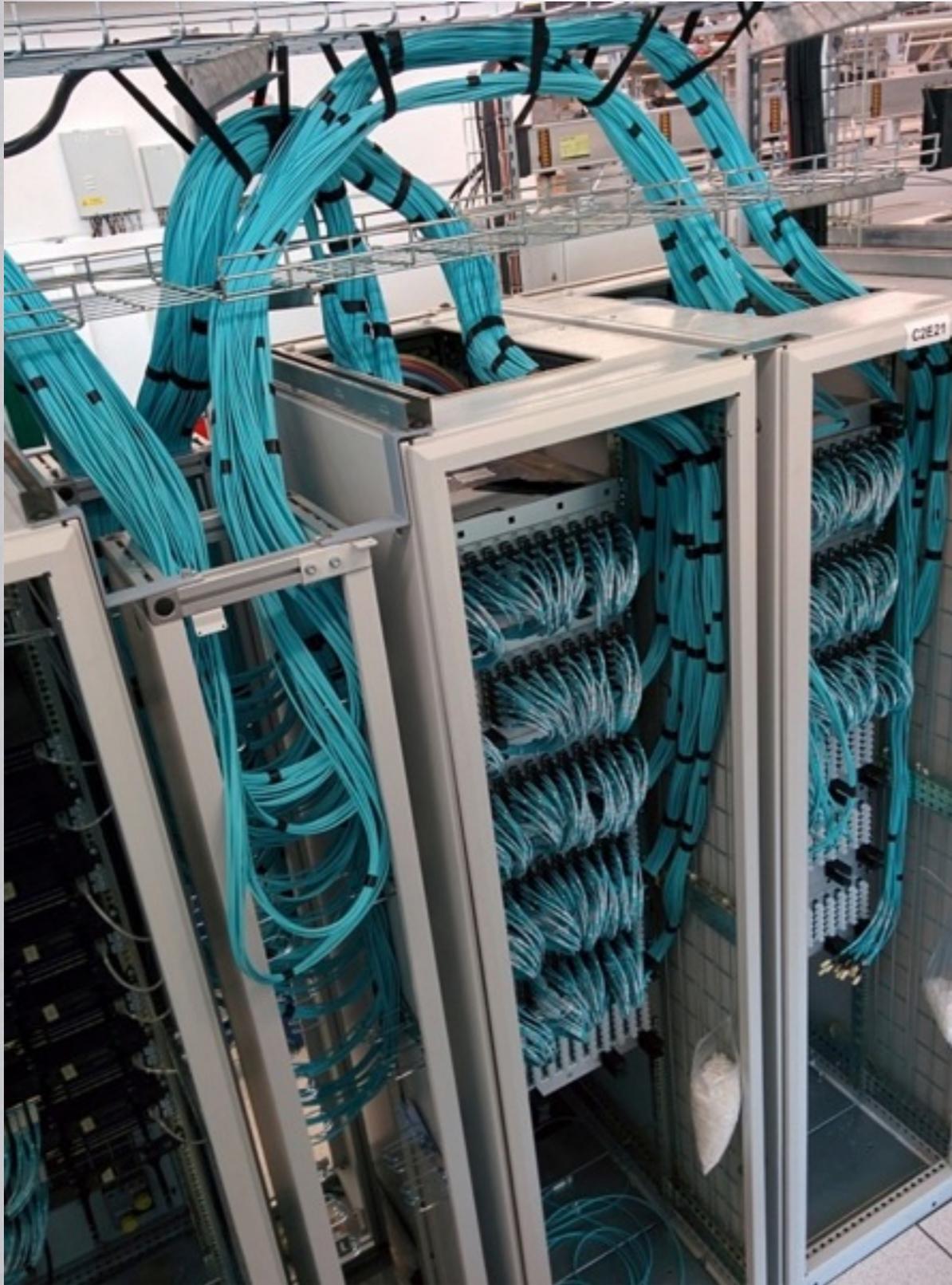


FRL/Myrinet



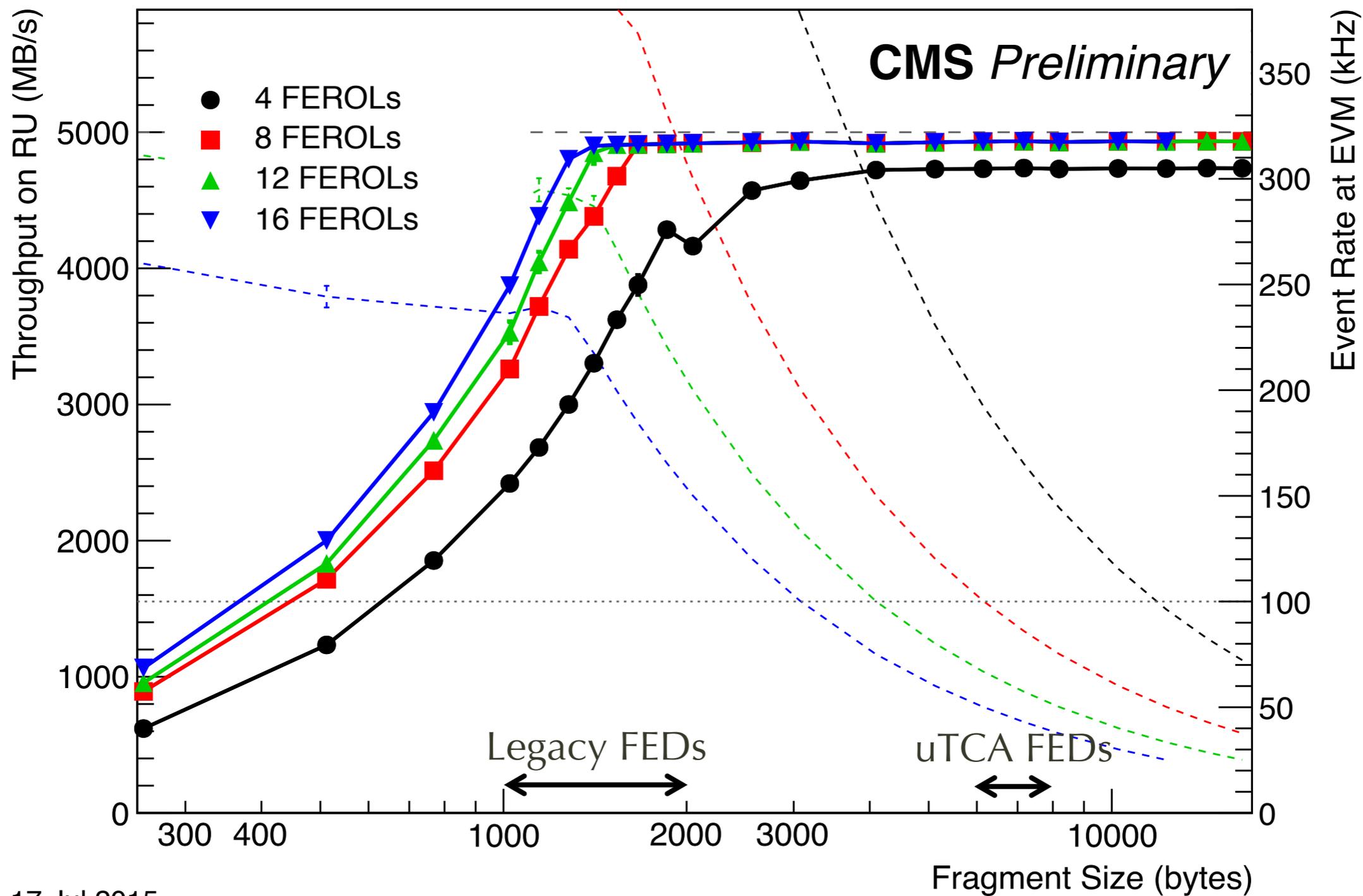
FRL/FEROL 10 Gb/s Ethernet

Switchover done in May 2014



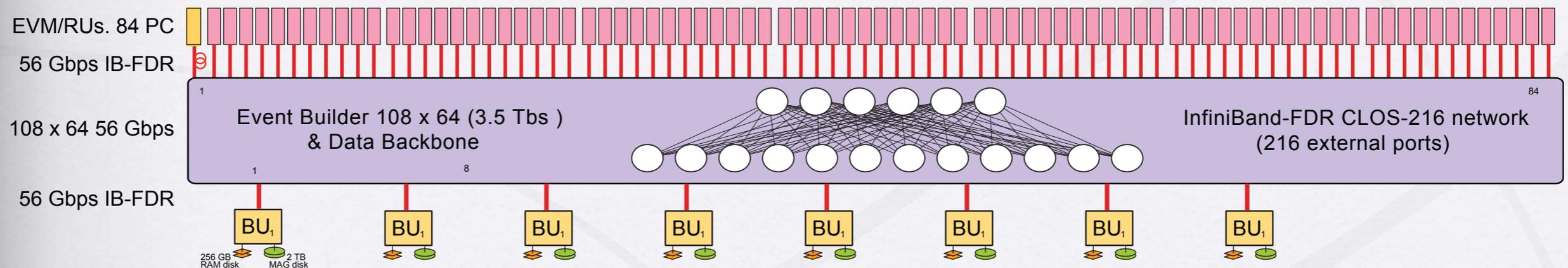
Data concentrator patch panels ... and switches

Data Concentrator



17 Jul 2015

Event Builder



Single-sliced event builder (DAQ1: 8 separate slices)

- 63 RUs and 62 BUs (DAQ1: 480 RUs and ~1000 BUs)
- Required re-engineered event-builder protocol and software
- Heavily multi-threaded & templated code to meet performance goal
- Optimized for NUMA (non-uniform memory architecture)

InfiniBand – most cost-effective solution

- Reliability in hardware at link level (no heavy software stack)
- Credit-based flow control (switches do not need to buffer)
- Can construct a large network from smaller switches

HLT farm organized in sub-farms connected to one BU

- DAQ1: each HLT node is also doing event building

Data Network

40/56Gb NICs
(Infiniband or Ethernet)

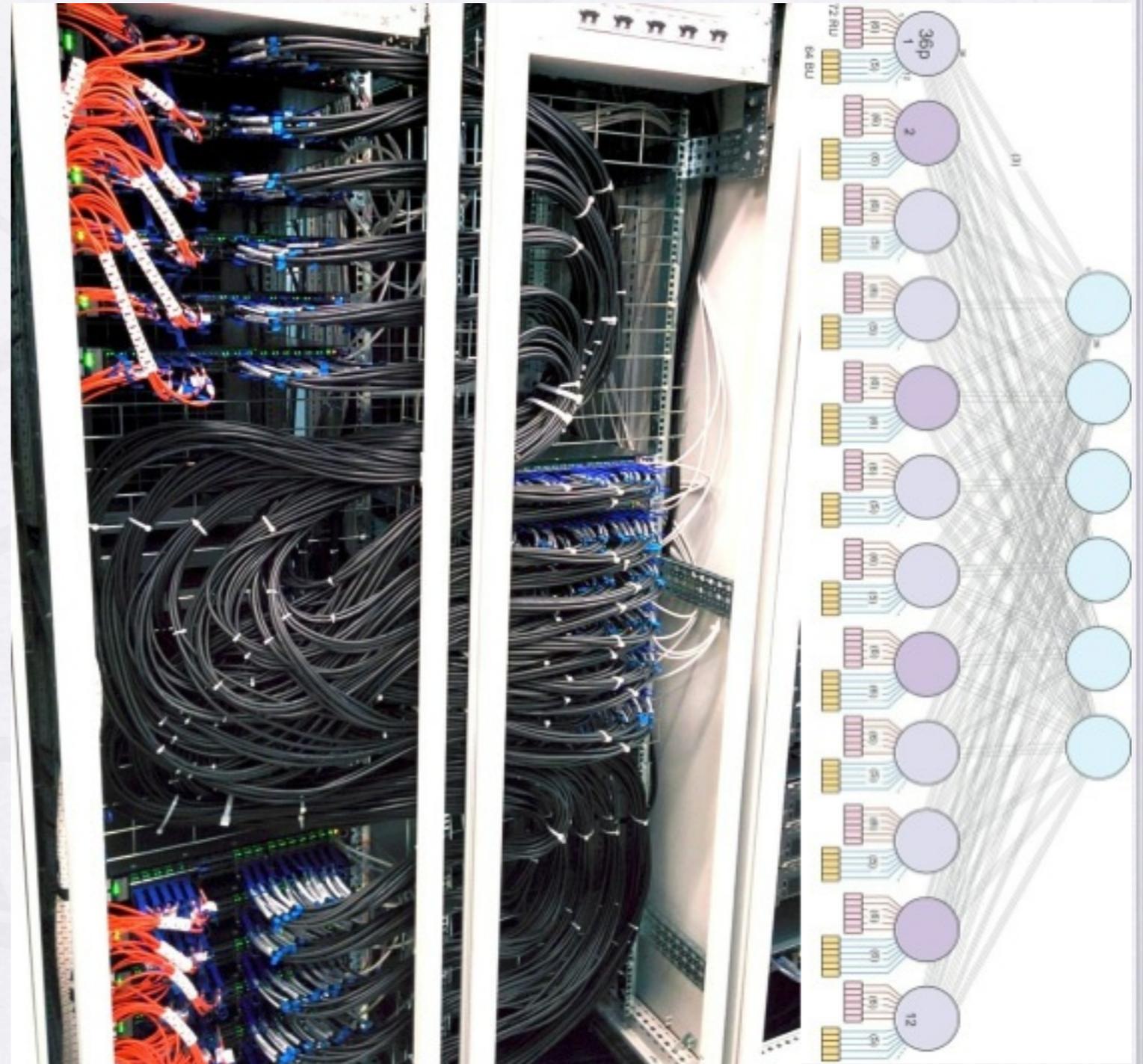
- Mellanox Technologies MT27500 Family [ConnectX-3]

10/40 GbE switches

- Mellanox SX1024 & SX1036

Infiniband switches

- Mellanox SX6036



Infiniband CLOS network

Computers

Readout Unit (RU)

- Dell PowerEdge R620
- Dual 8 core Xeon CPU E5-2670 0 @ 2.60GHz
- 32 GB of memory



Builder Unit (BU)

- Dell PowerEdge R720
- Dual 8 core Xeon CPU E5-2670 0 @ 2.60GHz
- 32+256GB of memory (240 GB for Ramdisk on CPU 1)
- 3.7 TB output disk (raid 1)



How to Achieve Performance

Avoid high rate of small messages

- Request multiple events at the same time
- Pack data of multiple events into one message

Avoid copying data

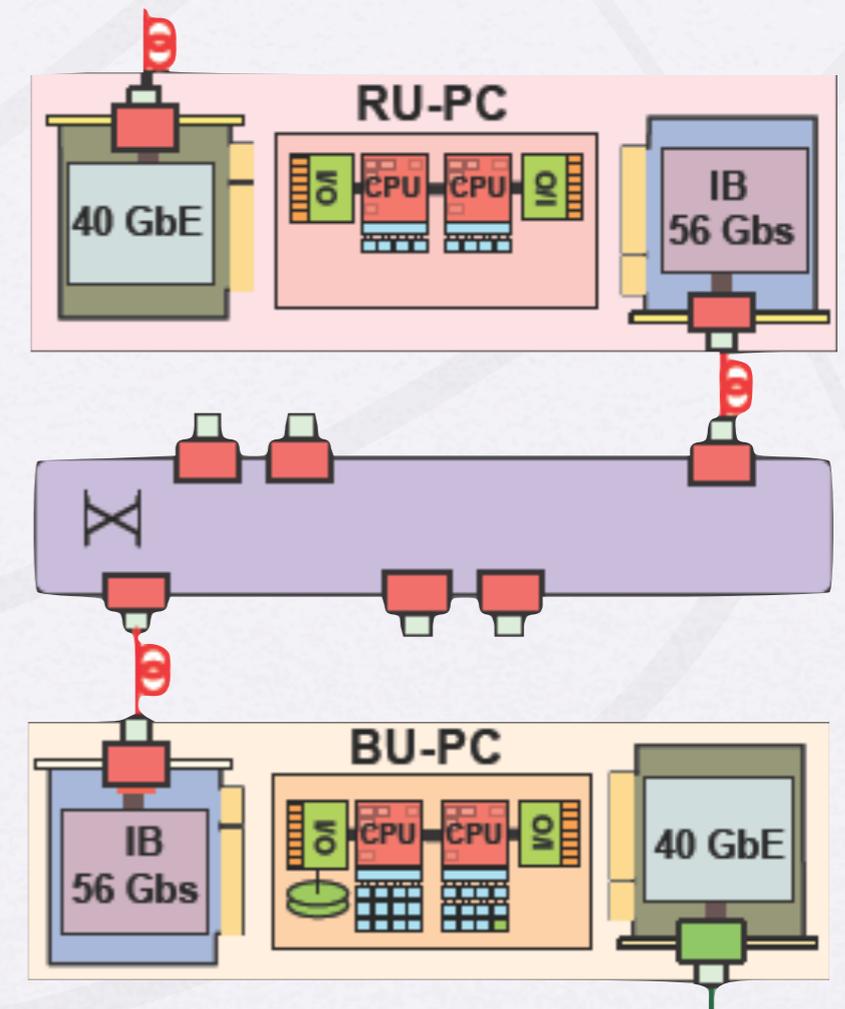
- Operate on pointers to data in receiving buffers
- Copy data directly into RDMA buffers of Infiniband NICs
- Stay in kernel space when writing data

Parallelize the work

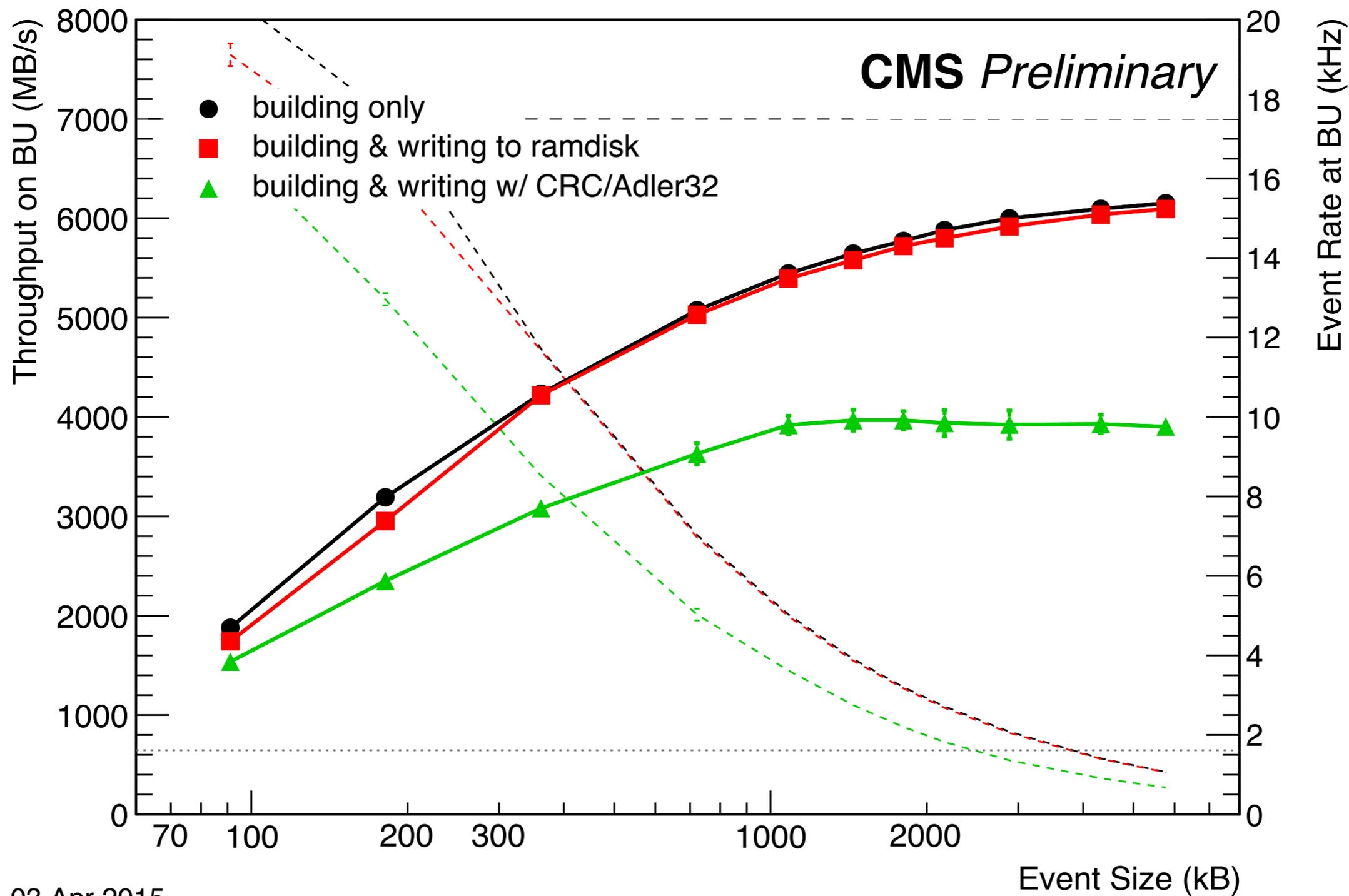
- Use multiple threads for data transmission and event handling
- Write events concurrently into multiple files

Bind everything to CPU cores and memory (NUMA)

- Bind each thread to a specific core
- Allocate memory structures on pre-defined CPU
- Restrict interrupts from NICs to certain cores
- Tune Linux TCP stack for maximum performance



Builder Unit



03 Apr 2015

File-based Filter Farm

CMSSW input/output is file based

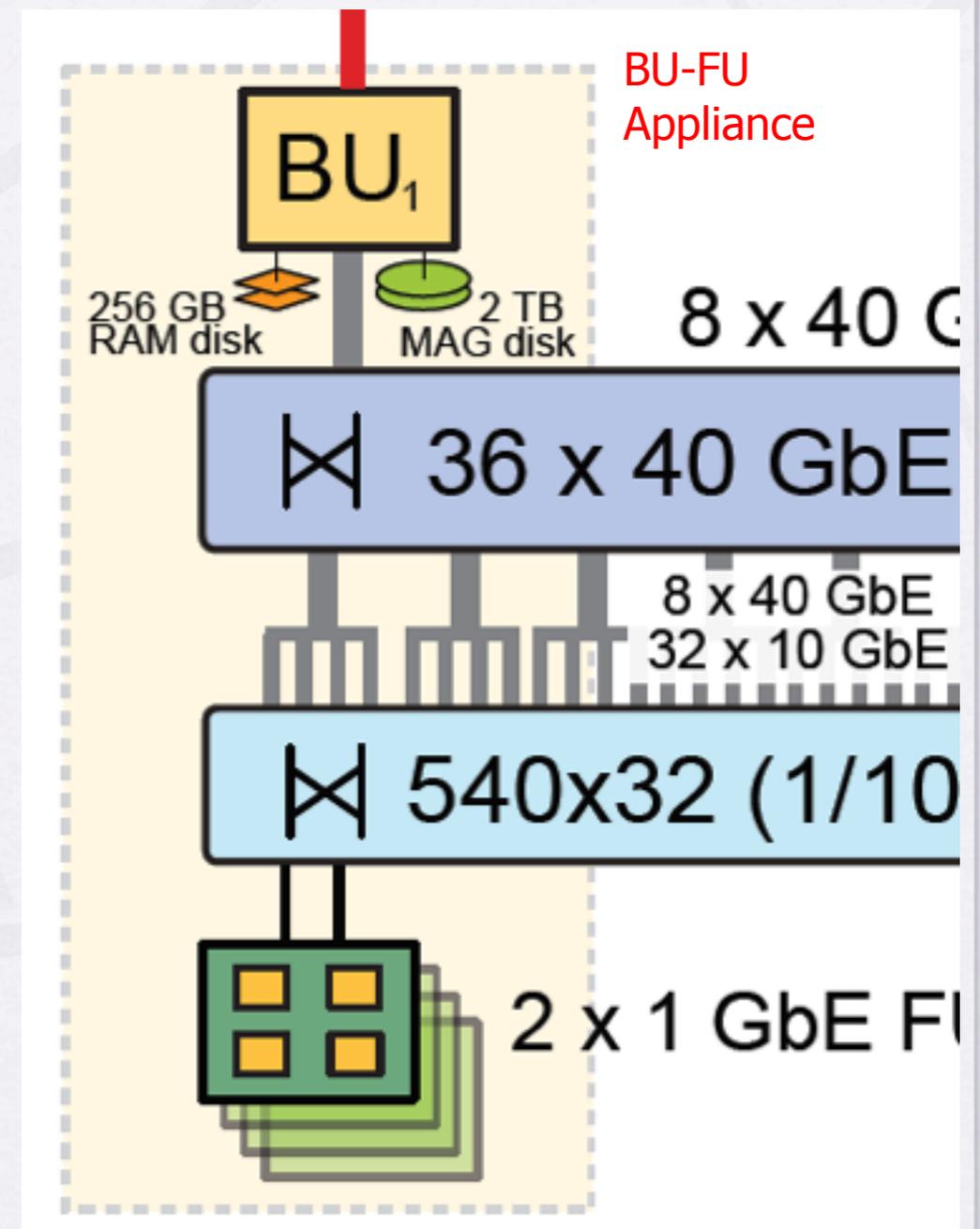
- Write the raw data to RAM disk on BU (256 GB/BU)
- Run standard CMSSW jobs (with cmsRun) with output to disk

Decouple CMSSW from XDAQ world

- Different frameworks with separate state machines, services, etc
- Different release cycles, compilers, etc
- Convoluted debugging

8-16 FUs mount RAM disk via NFS4

- Each FU runs multiple CMSSW instances
- Discover new files on BU RAM disk
- Output merged into a single file per FU
- Copied it back to output disk on the BU



High-Level Trigger Farm

May 2011
72x



May 2012
64x



2015
90x



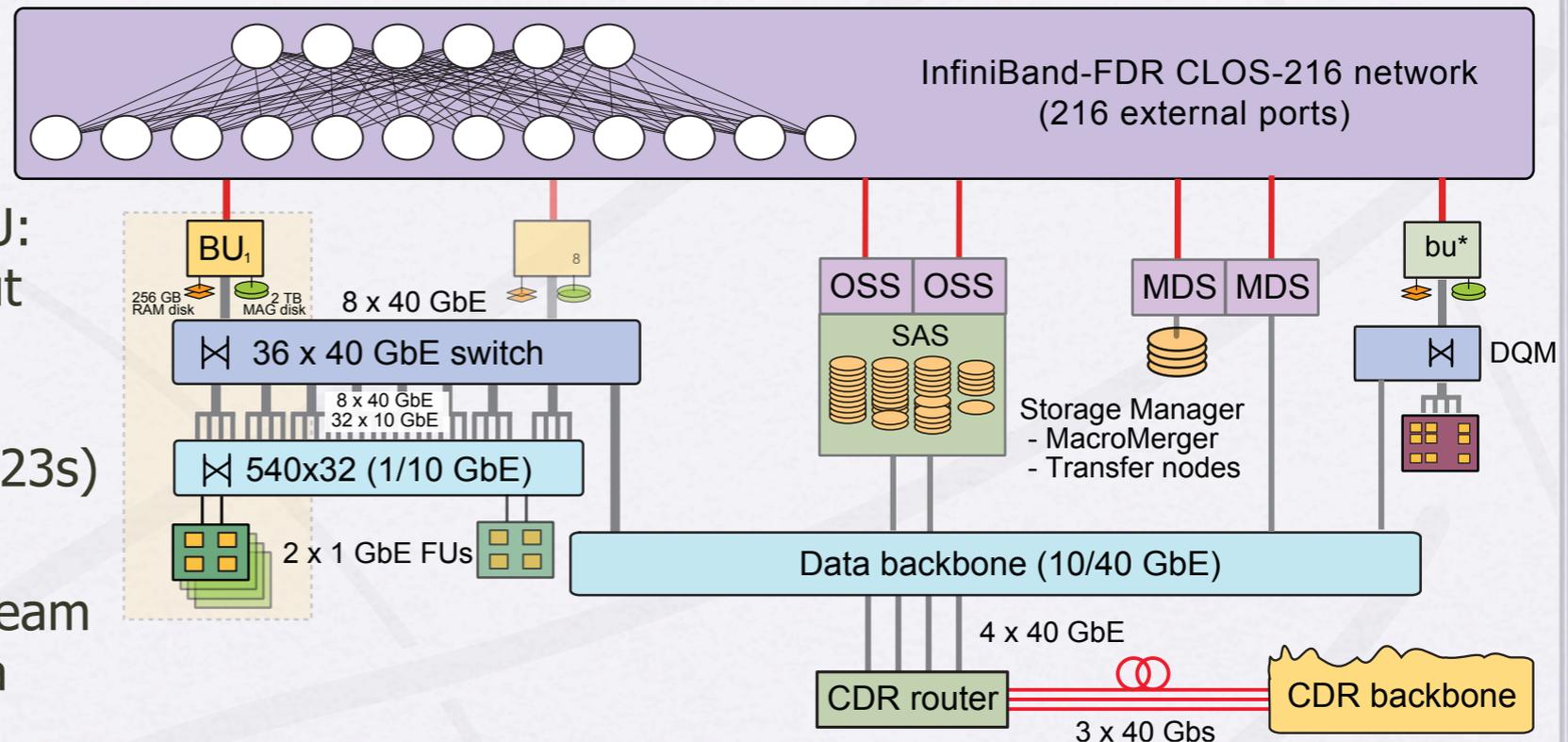
	2011 extension of DAQ 1 Dell Power Edge c6100	2012 extension of DAQ 1 Dell Power Edge c6220	HLT PC 2015 Megware S2600KP
Form factor	4 motherboards in 2U box	4 motherboards in 2U box	4 motherboards in 2U box
CPUs per mother-board	2x 6-core Intel Xeon 5650 Westmere , 2.66 GHz, hyper-threading, 24 GB RAM	2x 8-core Intel Xeon E5-2670 Sandy Bridge , 2.6 GHz, hyper threading, 32 GB RAM	2x 12-core Intel Xeon E5-2680v3 Haswell , 2.6 GHz, hyper threading, 64 GB RAM
#boxes	72 (=288 motherboards)	64 (=256 motherboards)	90 (=360 motherboards)
#cores	3456	4096	8640
Data link	2x 1Gb/s	2x 1Gb/s	1x 10 Gb/s, 1x 1Gb/s
Rel. perf. per node	0.6	1.0	1.66

Total: ~ 16 k cores on 900 motherboards

Merging and Storage

File-Based Filter Farm produces output files

- After merging on FU: 900 files x 10 output streams scattered over 62 BUs for every lumi section (23s)
- To be merged into 1 file per output stream and lumi section in a central place



Merging can be done by a file system

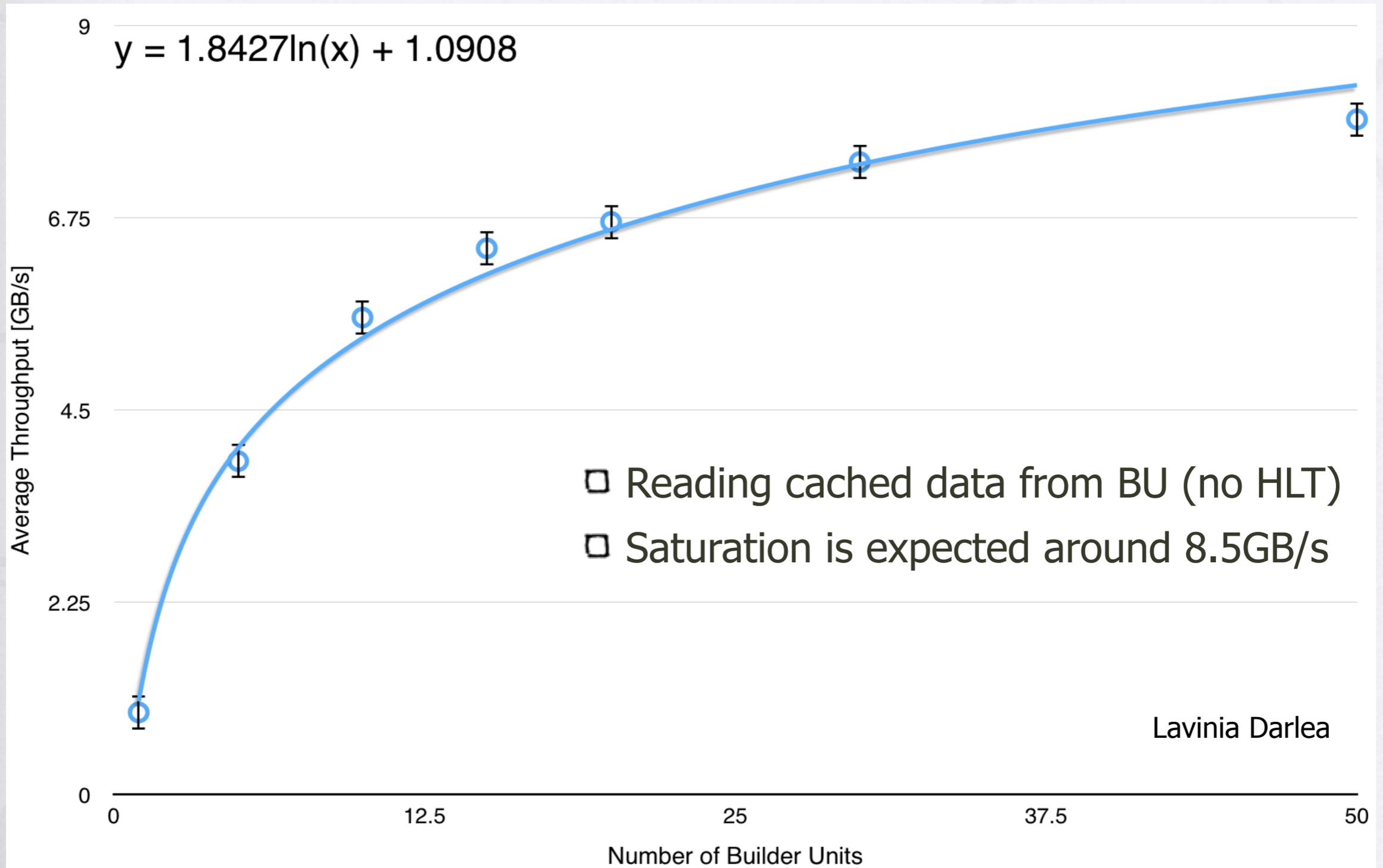
- Just need to find a file system that can handle it

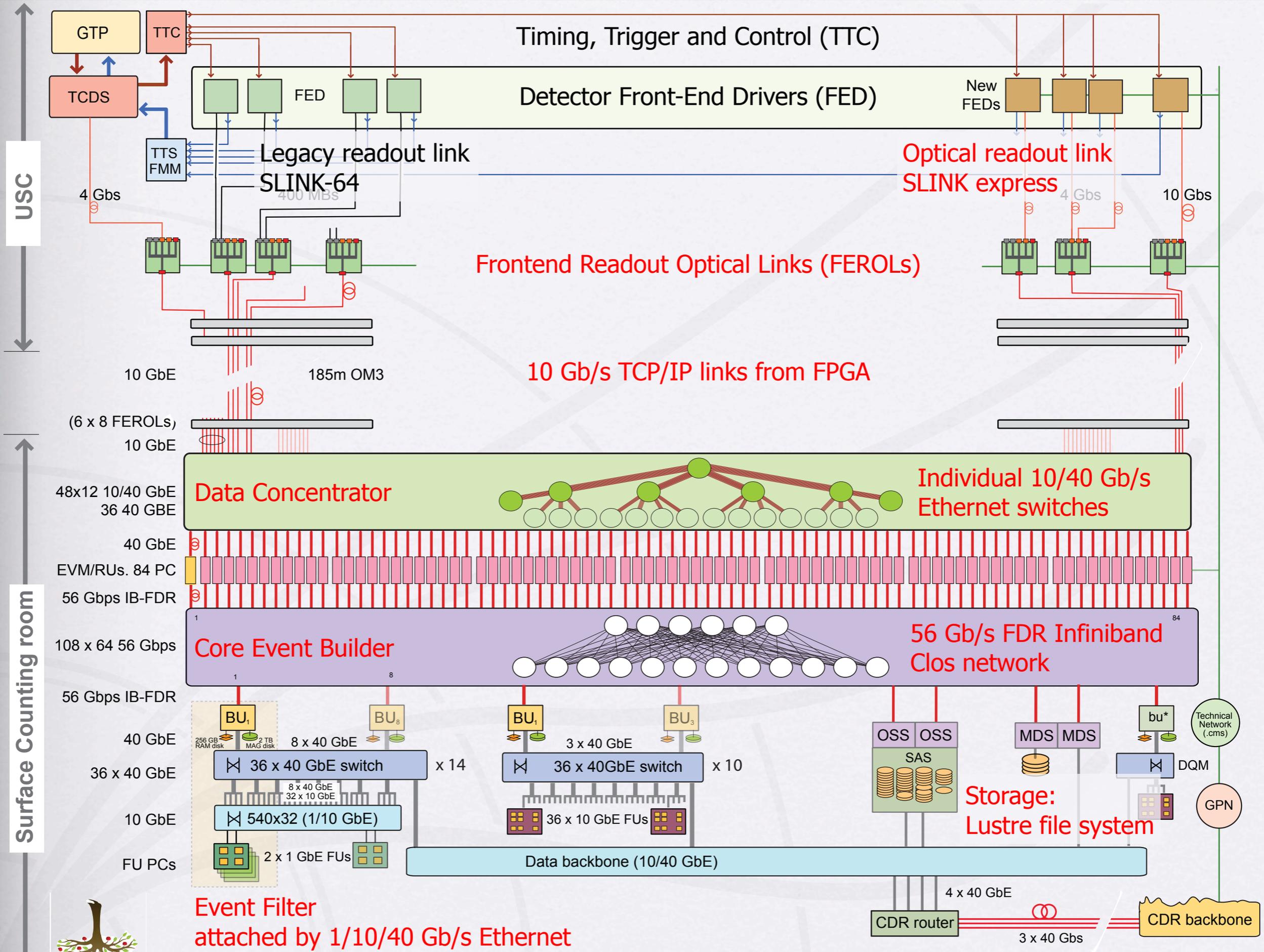
Solution: Global File System (Lustre) on a Storage System with 350 TB

- Merger process on BU reads data of all FUs in appliance from local disk
- Data are written concurrently from the BUs to a single output file in the global file system

Merged data is then transferred to tier 0 or to consumers at pt.5

Lustre Performance

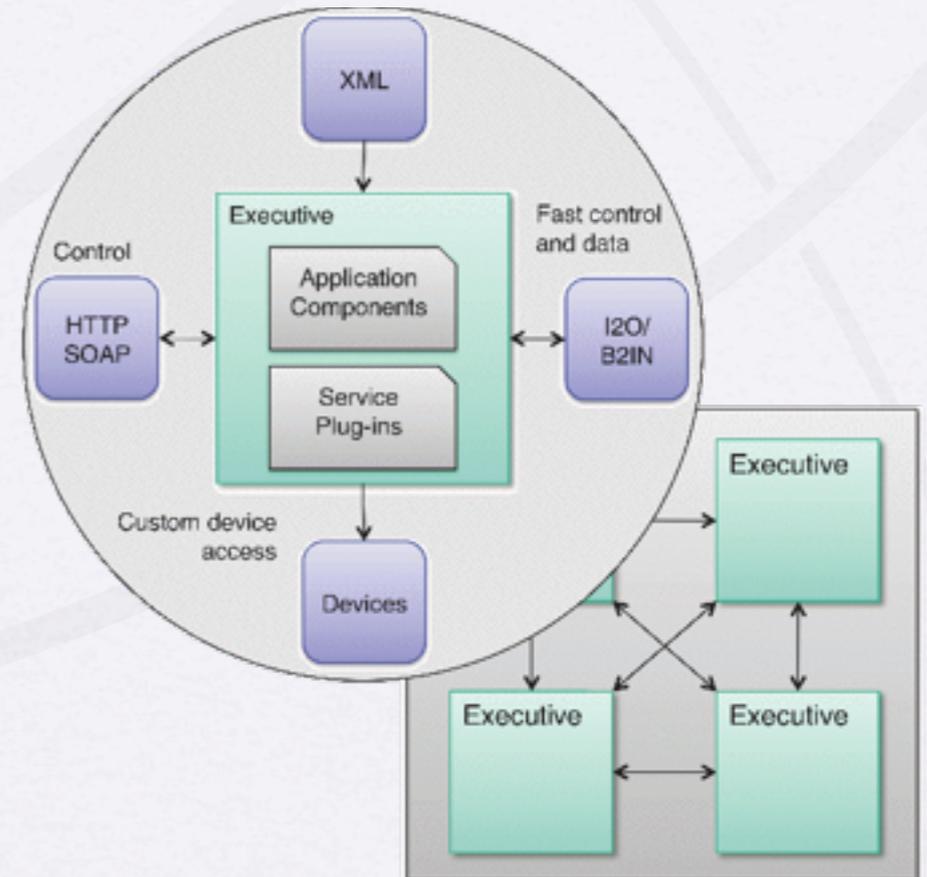




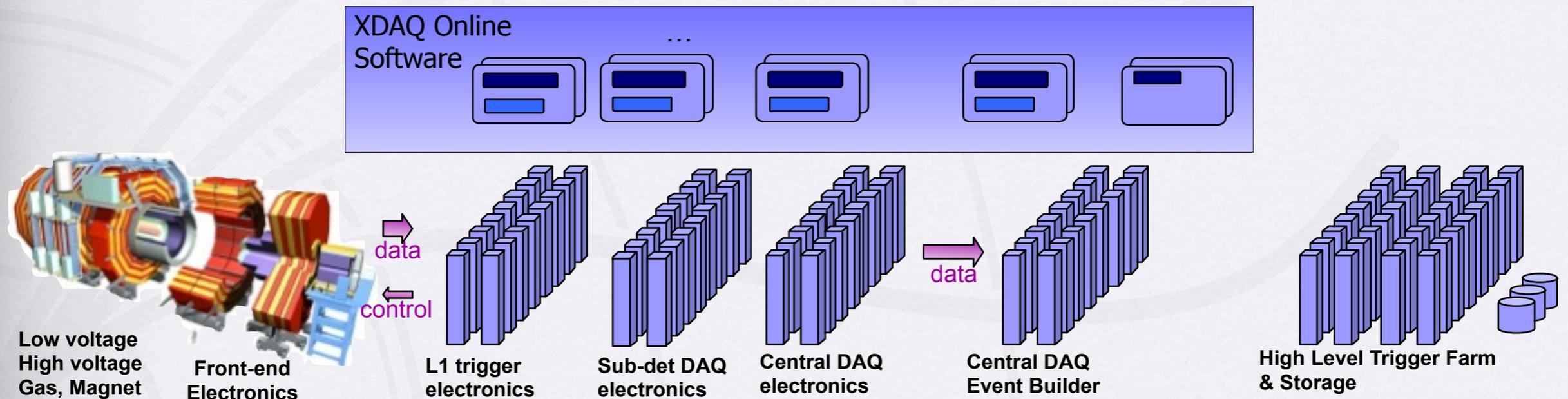
Online Software

C++ applications based on XDAQ framework

- Developed at CERN for distributed DAQ applications
- Reusable building blocks for
 - Hardware access
 - Transport protocols
 - Services
- Dynamic configuration based on XML
- Controlled and browsable with HTTP/SOAP



FFF daemons and services written in python 2.6



Run Control

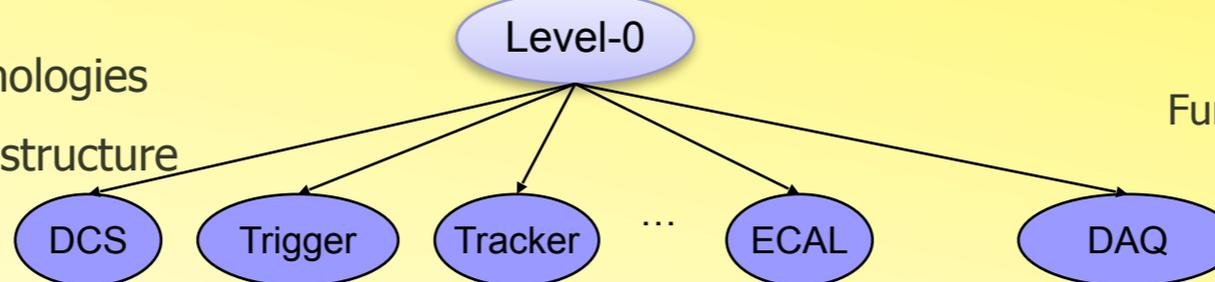


GUI in a web browser

- HTML, CSS, JavaScript, AJAX

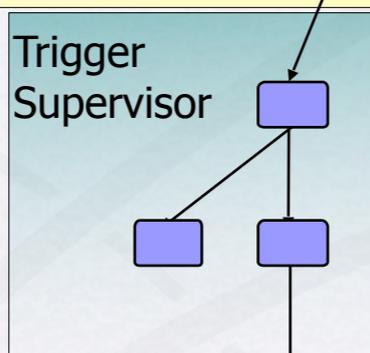
Run Control System

- Java, web technologies
- Defines control structure

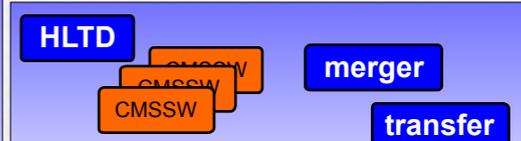
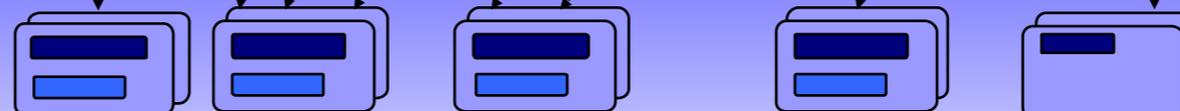


Function Manager

- Node in run-control tree
- Defines state machine & parameters
- Dynamically loaded into web application



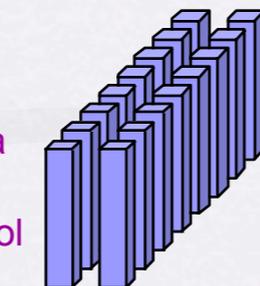
XDAQ Online Software



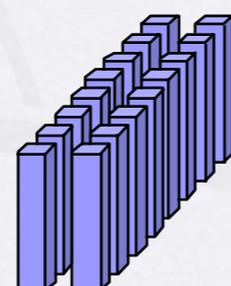
Low voltage
High voltage
Gas, Magnet

Front-end
Electronics

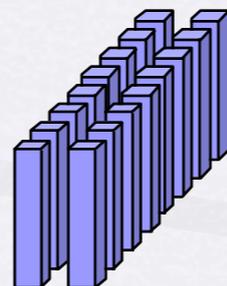
data
control



L1 trigger
electronics

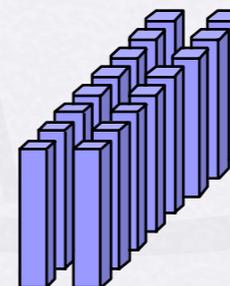


Sub-det DAQ
electronics

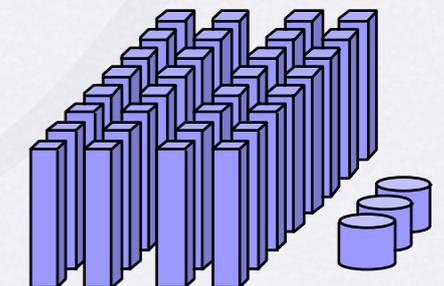


Central DAQ
electronics

data

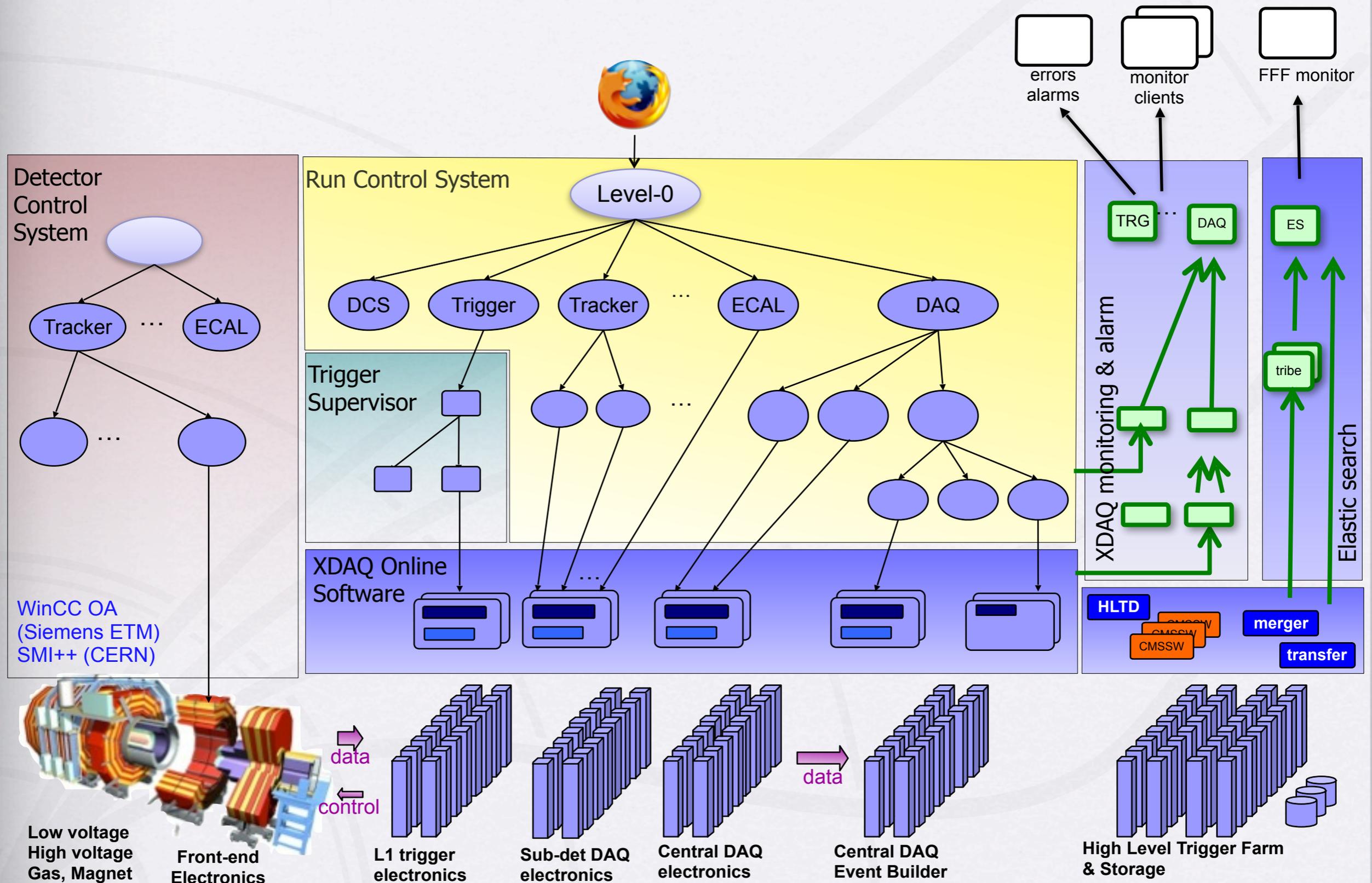


Central DAQ
Event Builder



High Level Trigger Farm
& Storage

Control & Monitoring



Level 0 Web-GUI

The screenshot displays the Level 0 Web-GUI interface, which is used for monitoring and controlling the LHC system. The interface is divided into several sections:

- Top Bar:** Contains navigation buttons like 'Status Table', 'RCMonitor', 'FED & TTS', 'Lock', 'save', 'Refresh', and 'Detach/Destroy'.
- Running Status:** A green bar at the top indicates the system is 'Running' with a time of '00:17.2'.
- Control Panel:** Includes buttons for 'Connect', 'Configure', 'Start', 'Pause', 'Resume', 'Stop', 'Halt', 'ColdReset', 'ForceStop', 'ForceStart', 'Recover', 'Interrupt', 'FixSoftError', 'TTChasync', 'TTChasreset', 'TTStestMode', and 'TestTTS'.
- DCS/LHC flag table:** A table showing the status of various flags like ES_HV_ON, PIX_HV_ON, TK_HV_ON, PHYSICS_DECLARED, and LHC_RAMPING.
- SETTING and CURRENTLY APPLIED VALUE:** A table showing settings like CMS Run Mode, L1HLT Trigger Mode, L1HLT Key, HLT Key, HLT SW ARCH, L1 Trigger Key, TSC Key, GT_RS Key, Clock source, and TCDS System.
- Configuration and Run Information:** Shows configuration details like 'Configuration: /app/prod/Global/levelZeroPM' and 'Run Number: 257487'. It also lists parameters like SID, Seq Name, Global Key, INWCFG Key, Level-0 Action, and Level-0 Error.
- Subsystem Status Table:** A table showing the status of various subsystems like PIXEL, TRACKER, ES, ECAL, HCAL, HF, DT, CSC, RPC, TCDS, TRG, SCAL, DAQ, DQM, and DCS. Each subsystem has a status indicator (Running, Connected, etc.) and a time value.
- Commander:** A row of buttons for each subsystem, each with a 'save' button and a status icon.

- Global state machine, aware of LHC states, e.g. stable beams
- Trigger configuration and clock source (LHC/local)
- Sub-system configurations, e.g. level of zero suppression
- Cross-checks and warnings to help the shifter
- Status and error messages of subsystem function-managers

Performance of Full System

Proton-proton running

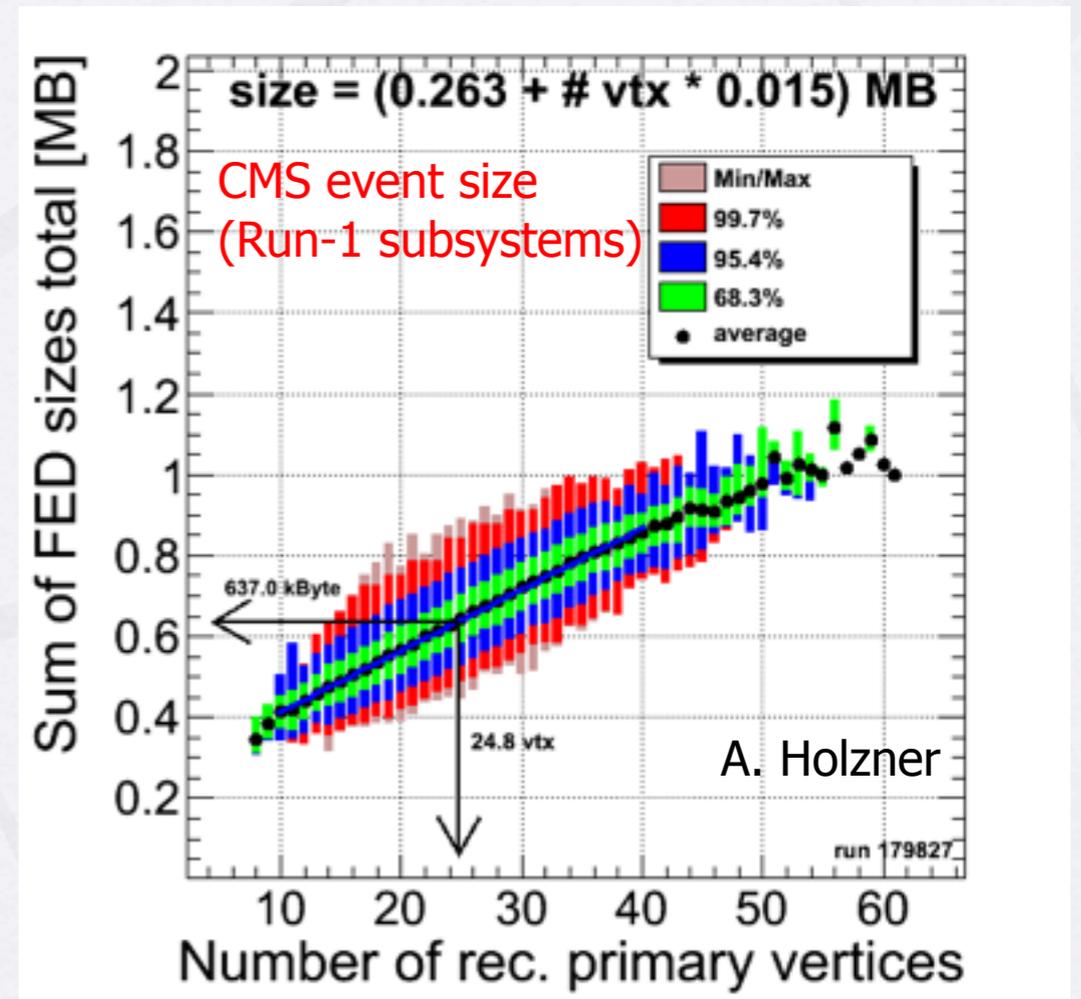
- nominal sizes for all legacy FEDs with a total event size of 750 kB (run 1 conditions with PU=40)
- 100 kHz with a margin in fragment sizes of a factor 1.5

Heavy ion conditions

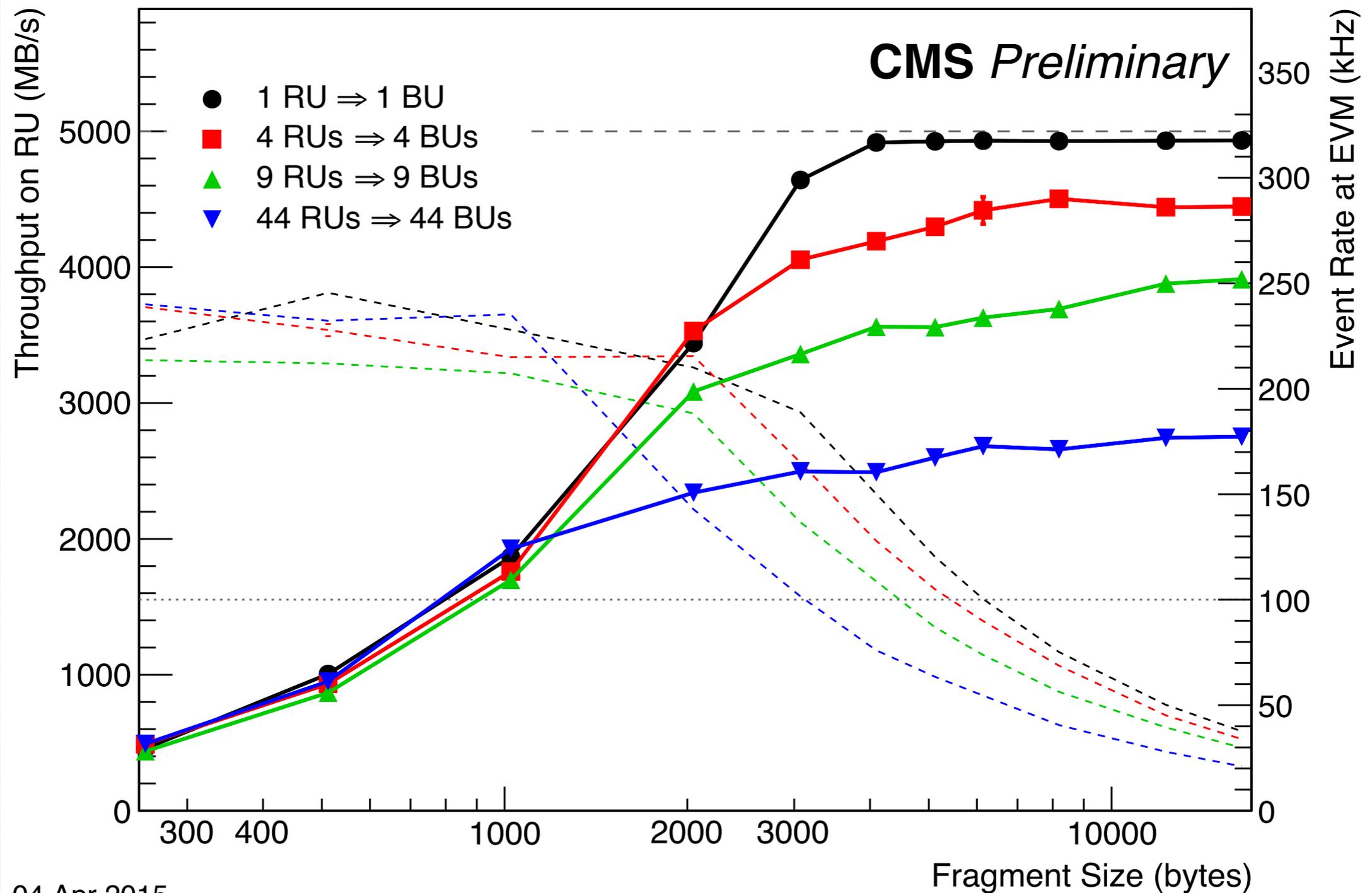
- 17 MB event size @ 10 kHz L1
- Event builder runs at 9.5 kHz
- Filter farm currently limited to ~8 kHz
 - Plan to add 10 more builder-units

Scaling behavior of event-builder is not fully understood

- No immediate concern for data taking
- Requires extended use of production system



Scaling of Event Builder



04 Apr 2015



Near Future

Integration of μ TCA Readouts

Integrated into DAQ

- TCDS with 1 μ TCA FED - in the readout since Sep 2014
- HCAL HF with 3 μ TCA FEDs - working fine since May 2015

Infrastructure installed and commissioning on the way

- HCAL HB/HE with 9 μ TCA FEDs
- L1 upgrade adds another 28 μ TCA FEDs
- Pixel test blade with 1 μ TCA

YETS 2016-17

- New pixel with 112 μ TCA FEDs @ 10Gb/s
- Developing next generation FEROL based on μ TCA
 - 4x 10Gb/s slink express in, 1x 40 Gb/s TCP/IP out
 - Not depending on FRL anymore

Work for the Next Years

Focus during LS 1 on commissioning the core DAQ system

- Many not-so-urgent issues pushed aside
- Will be worked on during run 2

Software improvements

- Improve fault tolerance and error reporting
- Performance improvements to reach ultimate goal of 1.5MB @ 100kHz
- Develop automatic testing facilities in DAQ function manager & LVL0

Monitoring

- Basic tools adapted from run 1
- Plan to use elastic search for the whole DAQ (beyond FFF)
 - Some work already started
 - Build on expertise gained to improve legacy monitoring
 - Allows much more post-mortem analysis
- New expert system for shifters & experts
- Revamp the error reporting for DAQ and CMS

Test Environments

DAQ testbed (daq2val)

- Based on legacy FEDs only
 - No TCDS
 - No μ TCA systems
- Not all aspects can be tested before deployment in production
- Plan to expand it during the next year

Subsystem testbed (bldg 904)

- Uses DAQ1 system
- Needs to be ported to DAQ2

Improve testing procedure

- No systematic or automatic testing of DAQ software
- Need to develop tools to facilitate functional and performance tests
- No continuous integration builds and tests

Recuperate HLT CPU Cycles

HLT farm sized to handle maximum luminosity

- CPUs are not fully used most of the time
- Waste of money and energy

Employ unused CPUs for offline processing

- Opportunistic use whenever resources are available
 - Must not interfere with HLT usage
 - Minimal changes to underlying hardware
 - Quick start and especially stop of offline processes
- Cloud overlay based on OpenStack

Currently used during long pauses in data taking

Aim for usage between LHC fills or during data taking

- Interface to dynamically assign CPUs to HLT or cloud
- Careful monitoring of available resources
- Requires offline workflows which are efficient when run for short times

Plans for CMS DAQ Beyond Run 2

No Radical Change for DAQ3

DAQ2 h/w will be at end-of-life in 2019

- Need to replace computers and network infrastructure

FEROLs will stay (unless there's a major disaster)

- More systems will switch to μ TCA readouts
- Next generation FEROLs will still use TCP/IP
- Data-concentrator network will stay on Ethernet

Need to re-evaluate event-builder network

- Will Infiniband still be the most cost effective solution?
- Unlikely that there's a technology which allows to shrink the DAQ system substantially (as for DAQ2)

Take into account lessons to be learned during run 2

Networking for Event-Builder

Ethernet

- Not a reliable network in switched environment
- Speed
 - 40 GbE exists on switch and NIC since ~2012
 - 100 GbE exists but still very expensive
 - 400 GbE defined

High-Performance Computing (HPC) Fabric interconnect

- Low-latency, reliable
- Infiniband 4xFDR 56 Gbps and 4xEDR 100 Gbps available
- New fabric interconnect forthcoming ..
 - 128 Gbps (2017-18), 200 Gbps (after 2020)
 - Integration of fabric port onto the CPU socket

Both technologies have switches with ~50 Tbps

Phase II (2025)

	LHC Run-I 7-8 TeV	LHC Phase-I upgr. 13 TeV	HL-LHC Phase-II upgr. 13 TeV	
Energy				
Peak Pile Up (Av./crossing)	35	50	140	200
Level-1 accept rate (maximum)	100 kHz	100 kHz	500 kHz	750 kHz
Event size (design value)	1 MB	1.5 MB	4.5 MB	5.0 MB
HLT accept rate	1 kHz	1 kHz	5 kHz	7.5 kHz
HLT computing power	0.21 MHS06	0.42 MHS06	5.0 MHS06	11 MHS06
Storage throughput (design value)	2 GB/s	3 GB/s	27 GB/s	42 GB/s

Requirements on DAQ increase by factor ~ 25

- Feasible from technical point of view
- Likely obtainable within reasonable budget

Same two-level architecture as current system

- L1 hardware trigger: 40 MHz clock driven, custom electronics
- High Level Trigger (HLT): event driven, COTS computing nodes

Main cost driver will be the HLT farm

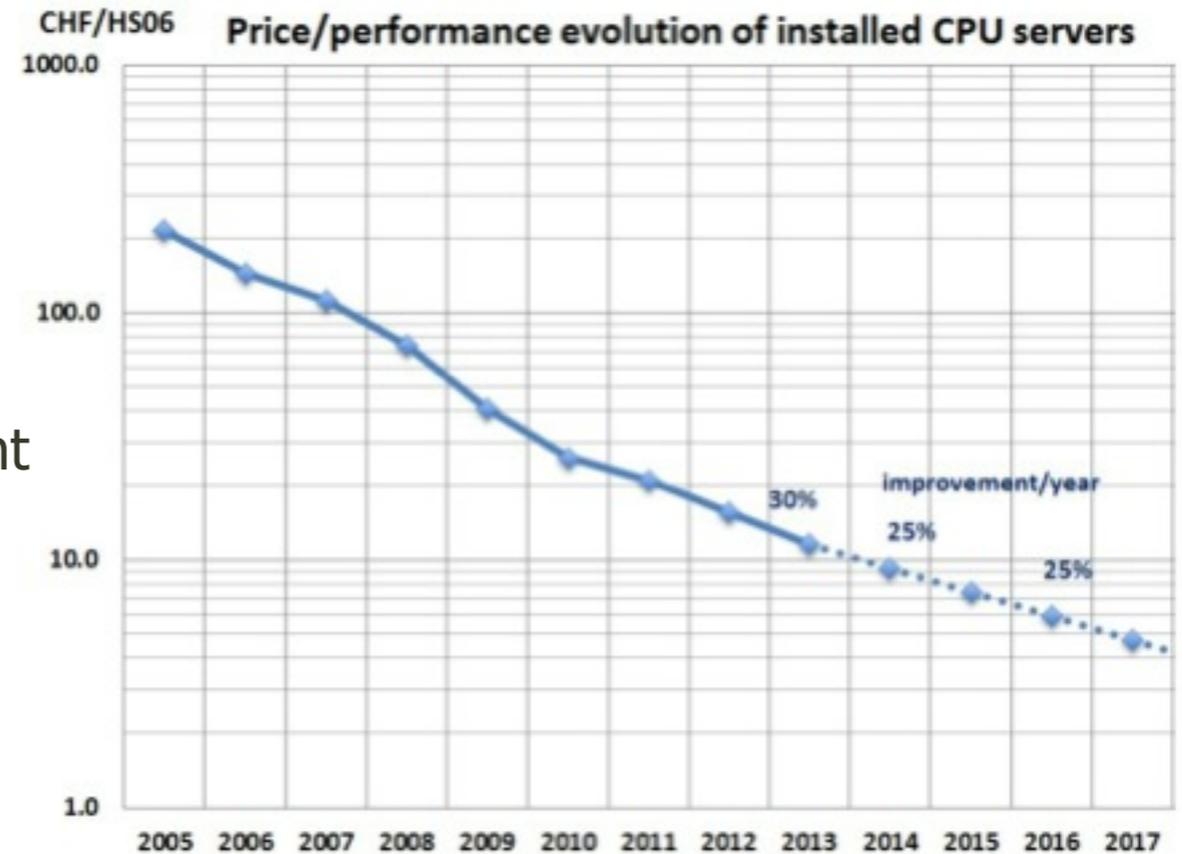
- Large uncertainty from technological progress and physics requirements

Costs Estimates

Extrapolation from existing system assuming technology evolution

DAQ readout, network, and storage

- Built with COTS computing, networking and storage equipment
- Assume factor of 10-20 performance improvement over 10 years at fixed costs
 - Observed in last decade from DAQ1 (~2007) to DAQ2 (2014)



High Level Trigger with similar reduction factor as present (1/100)

- CPU time per event largely unknown
 - Phase-II detector and L1/HLT menu
 - Influence of level-1 track trigger
 - Less time spend on track finding, but harder to reject events
 - Improvements due to code and selection optimization
- Possibly more cost effective CPU/GPU architectures in a decade

Personal Ideas

Rethinking the DAQ?

Predicting affordable technologies over >10 years is hard

- There will be CPUs with many, many cores
- Different specialized cores/processors will be available
- Access to main memory will still be relatively slow
- Accessing memory on remote machines will become faster
 - Intel® Omni-Path Architecture will move interconnects into the CPU
 - Optical I/O incorporated into FPGA

A big challenge for high-throughput computing

- Minimize time needed to get data into and out of the boxes
- Optimize data structures for easy access from CPU/GPU

Distributed computing/storage/search will be the standard

- DAQ has different usage pattern: write once, read once, delete
- Industry focuses on latency which is less important for DAQ

A chance to rethink how a data-acquisition system works?

Getting Data of the Detector

Ideally, data transferred at bunch-crossing rate from the detector

- Otherwise large fraction of b/w is used for trigger data

Trigger-less DAQ challenging in LHC environment

- High radiation environment affects on-detector electronics
- Power dissipation and size limits complexity of readout chips
- Would need high-speed, low-power, very radiation-hard readout links
 - Optical or (highly directional) wireless links integrated in the readout-ASICs
 - Maybe some Silicon Photonics or Internet-of-Things technology might help (if it's accidentally rad-hard)

Need careful definition of "raw" data

- Zero-suppression
- Pre-processing (e.g. tracklet)
- Can the same data be used for trigger and offline reconstruction?
 - Avoids that data needs to be transferred twice
 - Requires "offline" quality pre-processing on the detector

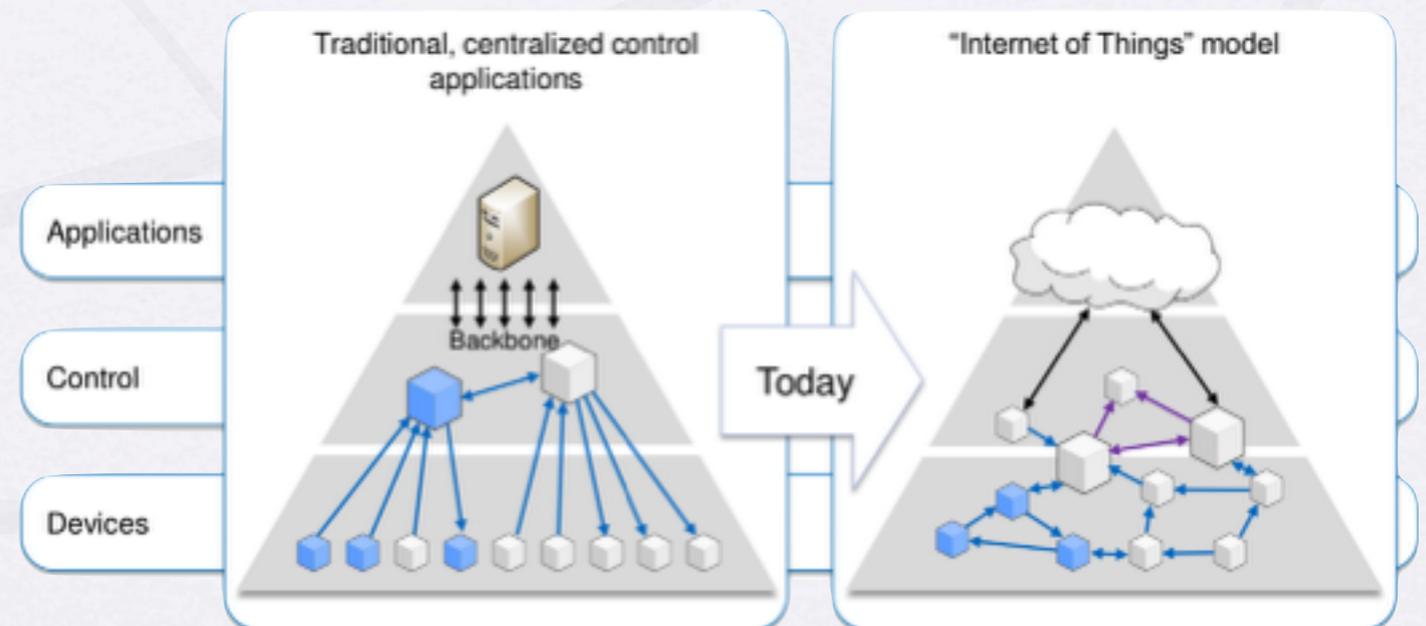
A New World?

Traditional event building uses hardware inefficiently

- Needs a lot of resources to transport data which is mostly unused
 - Used only for L1 trigger
 - Not processed by HLT and then discarded
- Network b/w is used only in one direction

Think more of a mesh (or Internet of Things)

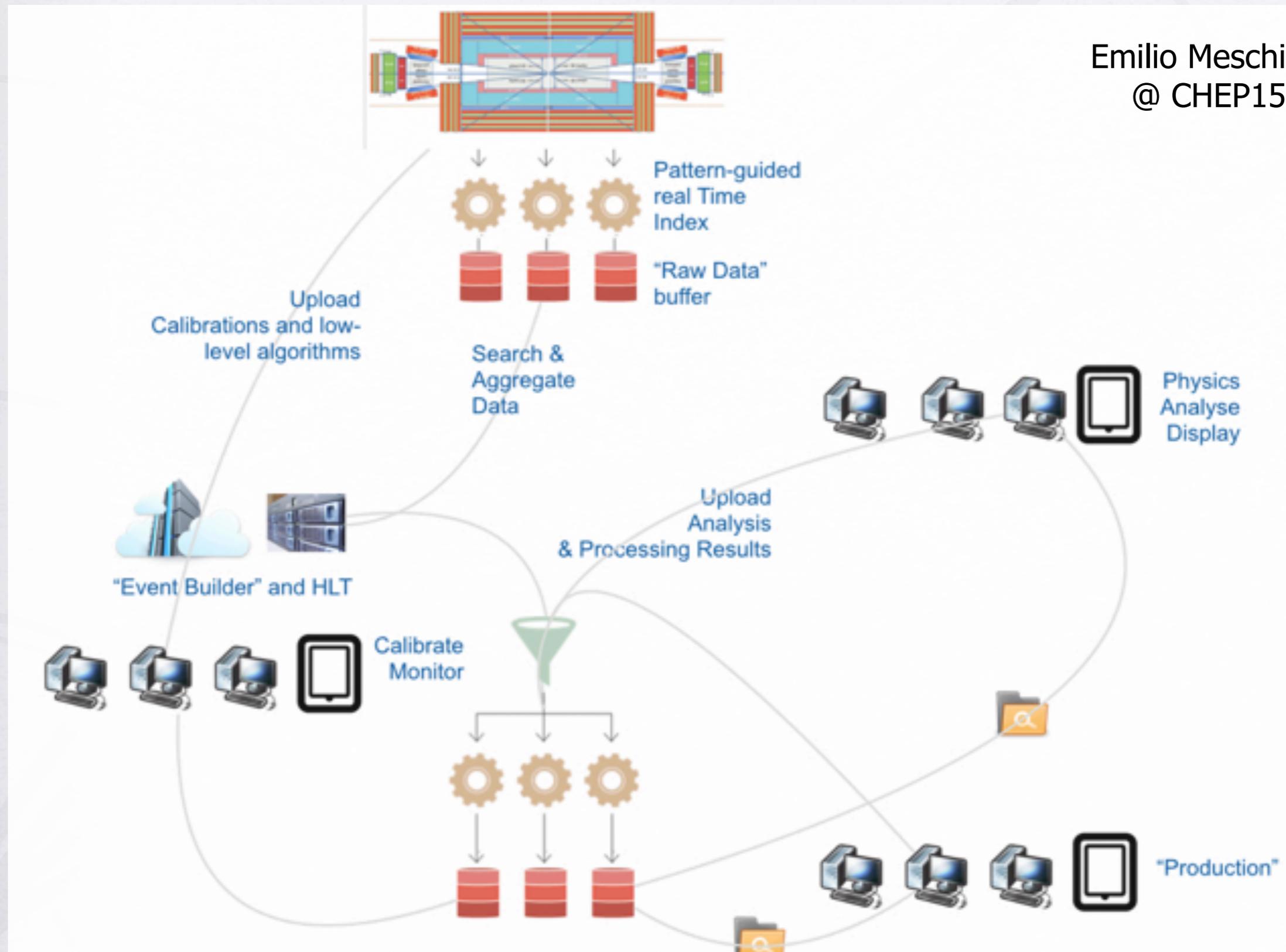
- Leave data as close as possible to the detector
- Pre-process it locally
 - Specialized processors (custom or commercial)
 - Generic CPUs
- Access it remotely
 - Event-building on demand
- Continuous calibrations with feedback to processors
 - Allows near offline-quality selection to reduce the event rate
 - Blurs boundary between online and offline reconstruction



Source: <http://db.in.tum.de/teaching/ws1314/industrialIoT>

Search for Your Data?

Emilio Meschi
@ CHEP15



Summary

CMS has a complete new DAQ system for LHC run 2

- State-of-the-art technology
- Order of magnitude smaller than old DAQ system
- Achieves run 1 performance
- More work needed to use full potential
- New sub-system readouts are being integrated

The changes for run 3 (2019) will be less radical

- But still a lot of planning and work will be needed

Open field for phase 2 DAQ (2025)

- Evolution of current DAQ as baseline
- Toying with radical new ideas
- Opportunities for R&D on modest budget
 - High-speed, low-mass, rad-hard readout links (mostly unidirectional)
 - Data-reduction schemes on- or near-detector w/o affecting physics
 - Tie event building and selection into a mesh

Endnotes

Acknowledgments

- The CMS DAQ group
- S. Cittolin, S. Erhan, F. Meijers, E. Meschi, N. Neufeld, H. Sakulin, P. Zejdl for their input and slides

References

- Technical Proposal for the Phase-II Upgrade of the CMS Detector ([CERN-LHCC-2015-010](#))
- CMS Phase II Upgrade Scope Document ([CERN-LHCC-2015-019](#))
- The New CMS DAQ System for Run 2 of the LHC ([CMS-CR-2014-082](#))
- 10 Gbps TCP/IP streams from the FPGA for the CMS DAQ Eventbuilder Network ([CMS-CR-2013-416](#))
- Using the CMS High Level Trigger as a Cloud Resource ([J.Phys.Conf.Ser. 513 \(2014\) 032019](#))

If you are interested to work on CMS DAQ, please talk to me