21st IEEE Real Time Conference

iFDAQ for the COMPASS experiment

Bodlák Martin, Frolov Vladimir, Huber Stefan, Jarý Vladimír, Konorov Igor, Květoň Antonín, Nový Josef, Steffen Dominik, Šubrt Ondřej, Tomsa Jan, Virius Miroslav

Sponsored by:

Bundesministerium für Bildung und Forschung

MINISTRY OF EDUCATION, YOUTH AND SPORTS

TUM LMU Excellence Cluster Universe

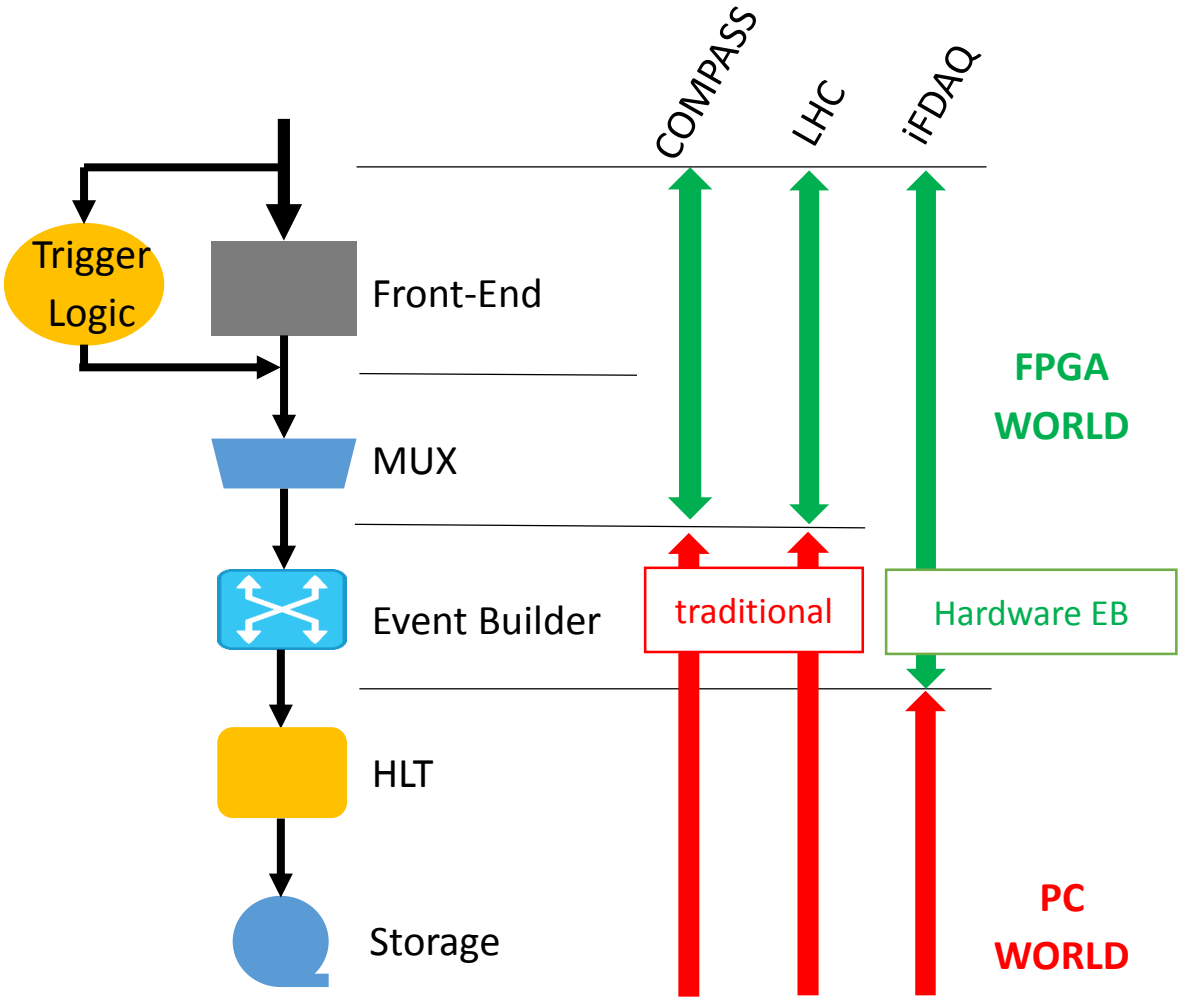
MLL

Contents

1. Motivation and Concept of Hardware Event Building
2. Design of the intelligent FPGA-based DAQ
   - System
   - Hardware
   - (Firmware)
   - (Software)
3. iFDAQ – integration and performance in COMPASS
   - Setup of COMPASS spectrometer
   - iFDAQ – Performance and Status in 2017/2018
4. Switching Network Topology
   - Required Hardware Developments
   - Software Developments
5. Outlook and Conclusion

1. **Motivation and Concept of Hardware Event Building**
2. Design of the intelligent FPGA-based DAQ
   - System
   - Hardware
   - (Firmware)
   - (Software)
3. iFDAQ – integration and performance in COMPASS
   - Setup of COMPASS spectrometer
   - iFDAQ – Performance and Status in 2017
4. Switching Network Topology
   - Required Hardware Developments
   - Software Developments
5. Outlook and Conclusion

Different DAQ Systems



Concept of iFDAQ:
Minimize amount of real-time processes and push Event Building into FPGA-world

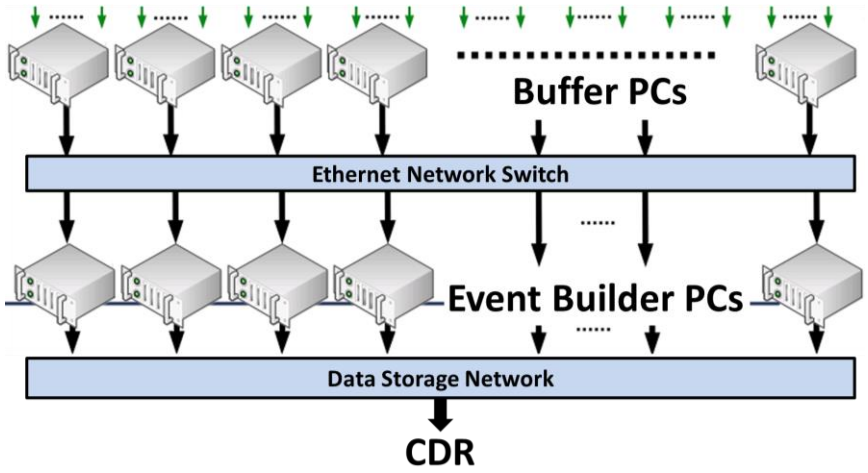
Feasibility due to recent FPGA developments:
- Increased I/O bandwidth
- Support for high-performance SDRAMs

Expected Advantages:
- Increased Compactness
- Increased Reliability
- Higher Scalability
- Reduced Costs

Event Builder – CPU vs. FPGA

Event Building: combination of logically connected, but physically split data fragments

CPU



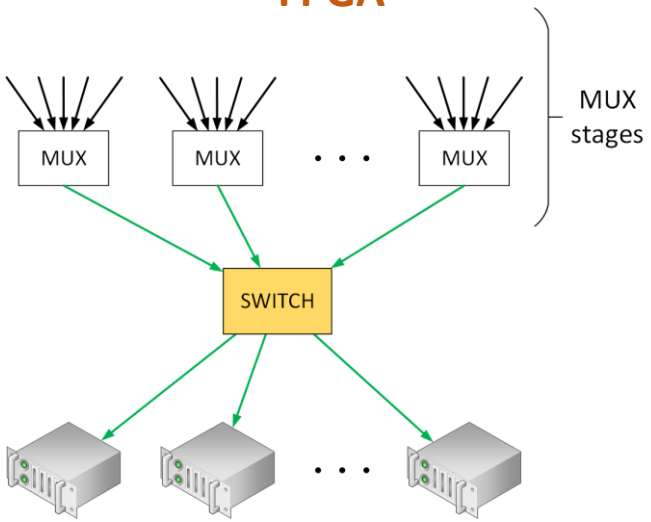
Buffer PCs:

- Buffers data until receiving control msg to forward to specific builder unit
- Replicated over number of computers to fit performance needs and application scenario

Event Builder PCs:

- Collects event fragments and combines them into complete event
- Replicated over number of computers to fit throughput requirements

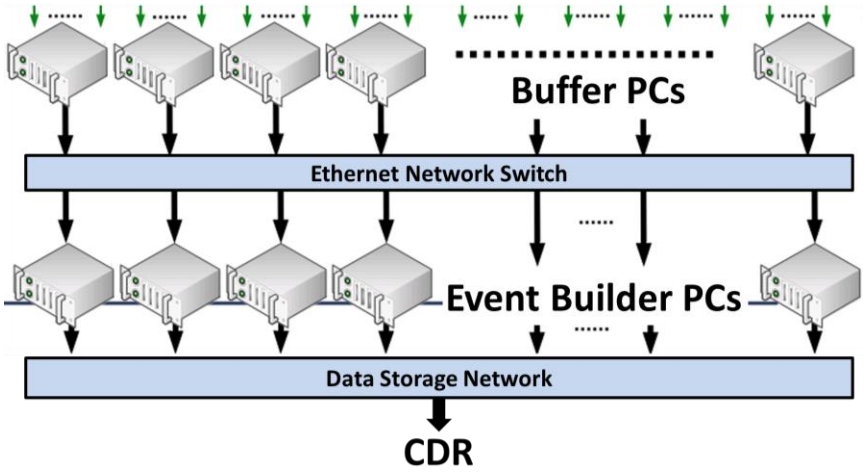
FPGA



- Usage of FPGAs and exploiting its properties:
  - Parallel processing
  - Pipeline architectures
- Collecting of all data in one FPGA-module
- Optional multiplexing stages to reduce number of incoming links
- Distribution of data flow to different computer nodes

Event Builder – CPU vs. FPGA

CPU (sequential)



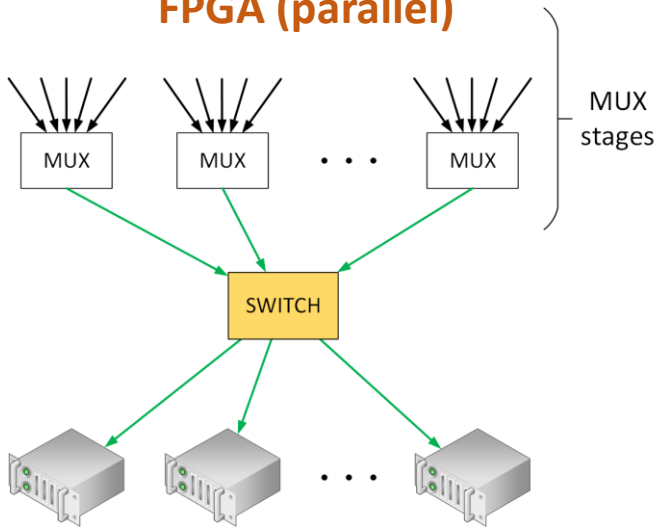
Disadvantages:

- Throughput limited by EB-network
- Performance of sequential execution strongly depends on algorithm complexity
- Recovery of hanging processes takes significant time

Advantages

- Easy integration of redundancy elements
- Uses mass-produced components
- Knowledge and templates available

FPGA (parallel)



Advantages:

- Continuation of the pipeline architecture in FEE
- Only FPGA allows to build real real-time systems
- High scalability
- High reliability
- Reduced costs

Disadvantages:

- Long development

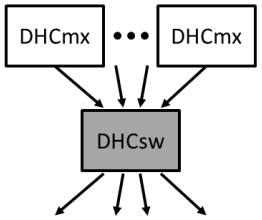
  But: Progress in higher-level tools (System Verilog, OSVVM)

⇒ **Motivation for**

- Minimizing real-time processes
- Development of highly automatized and reliable DAQ

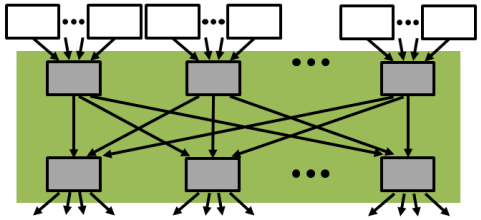
Scaling Possibility of Hardware EB

10 GB/s



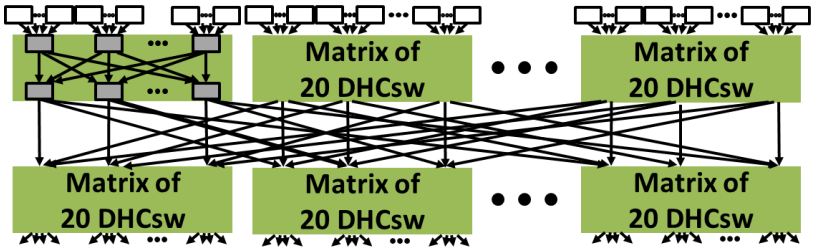
1 DHCsw

100 GB/s



2 layers of DHCsw

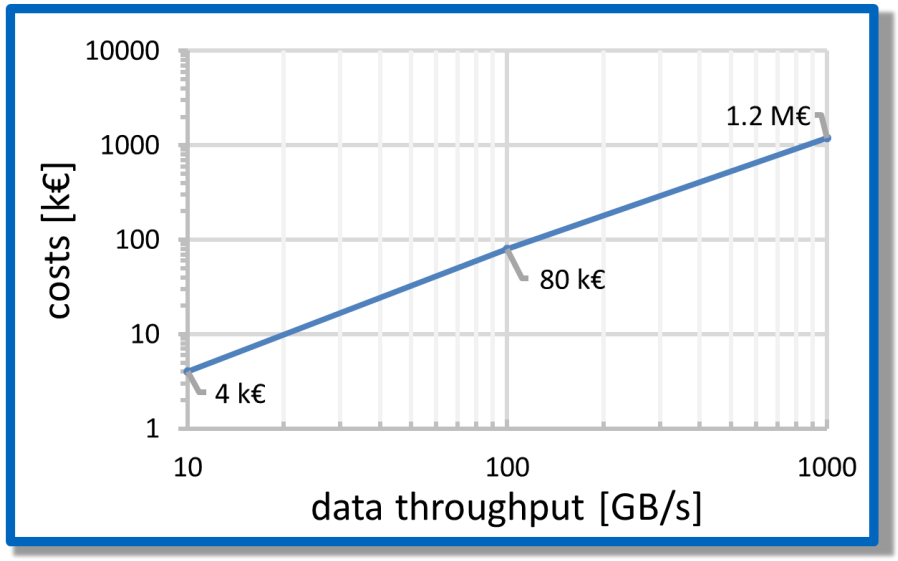
1 TB/s



4 layers of DHCsw

Scenario for:

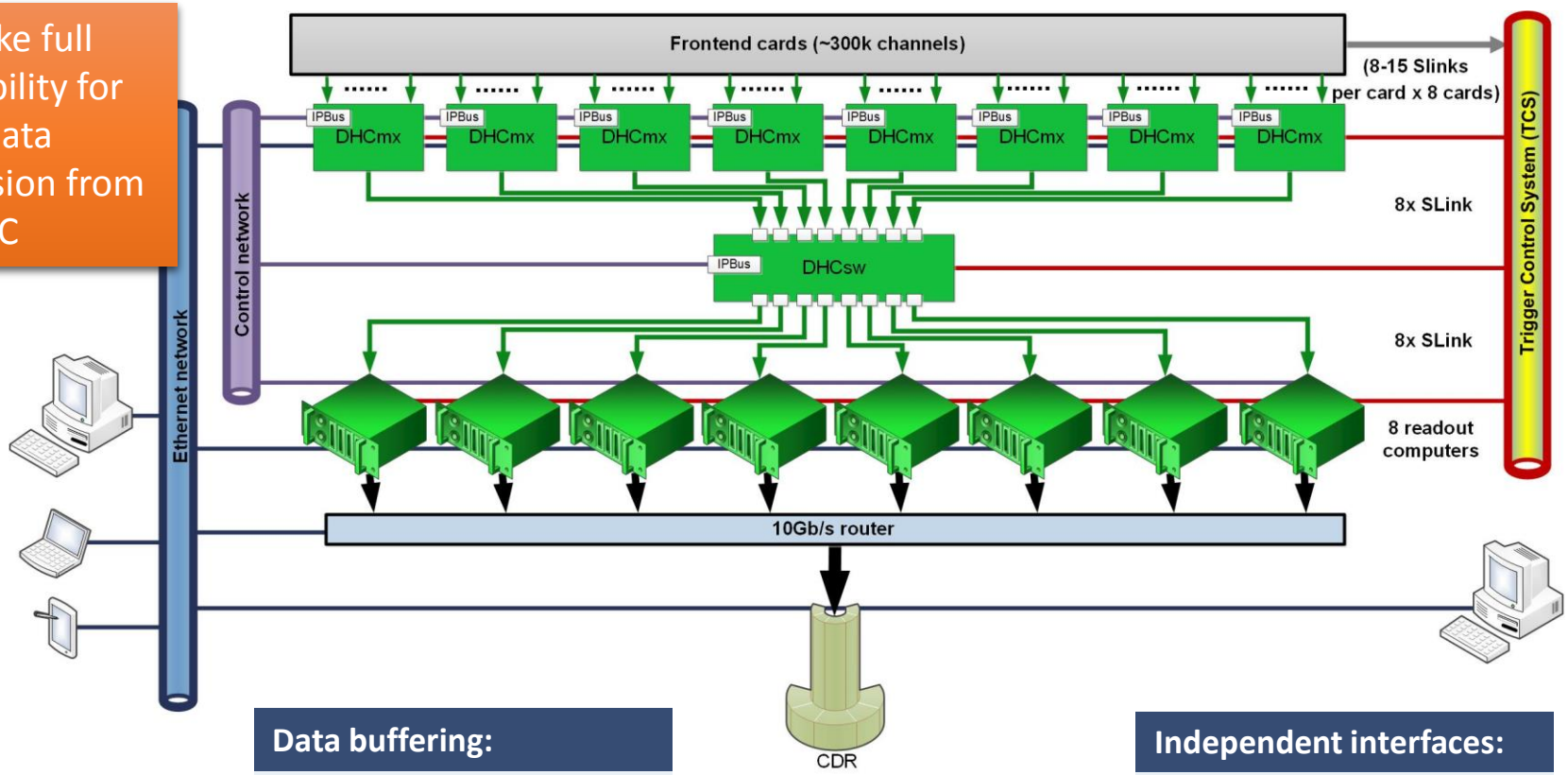
- Xilinx **7-series FPGA**
- SLINK interfaces replaced by **Aurora**



1. Motivation and Concept of Hardware Event Building
2. **Design of the intelligent FPGA-based DAQ**
   - **System**
   - **Hardware**
   - **(Firmware)**
   - **(Software)**
3. iFDAQ – integration and performance in COMPASS
   - Setup of COMPASS spectrometer
   - iFDAQ – Performance and Status in 2017
4. Switching Network Topology
   - Required Hardware Developments
   - Software Developments
5. Outlook and Conclusion

iFDAQ – System Design

FPGAs take full responsibility for reliable data transmission from FEEs to PC



Data buffering:

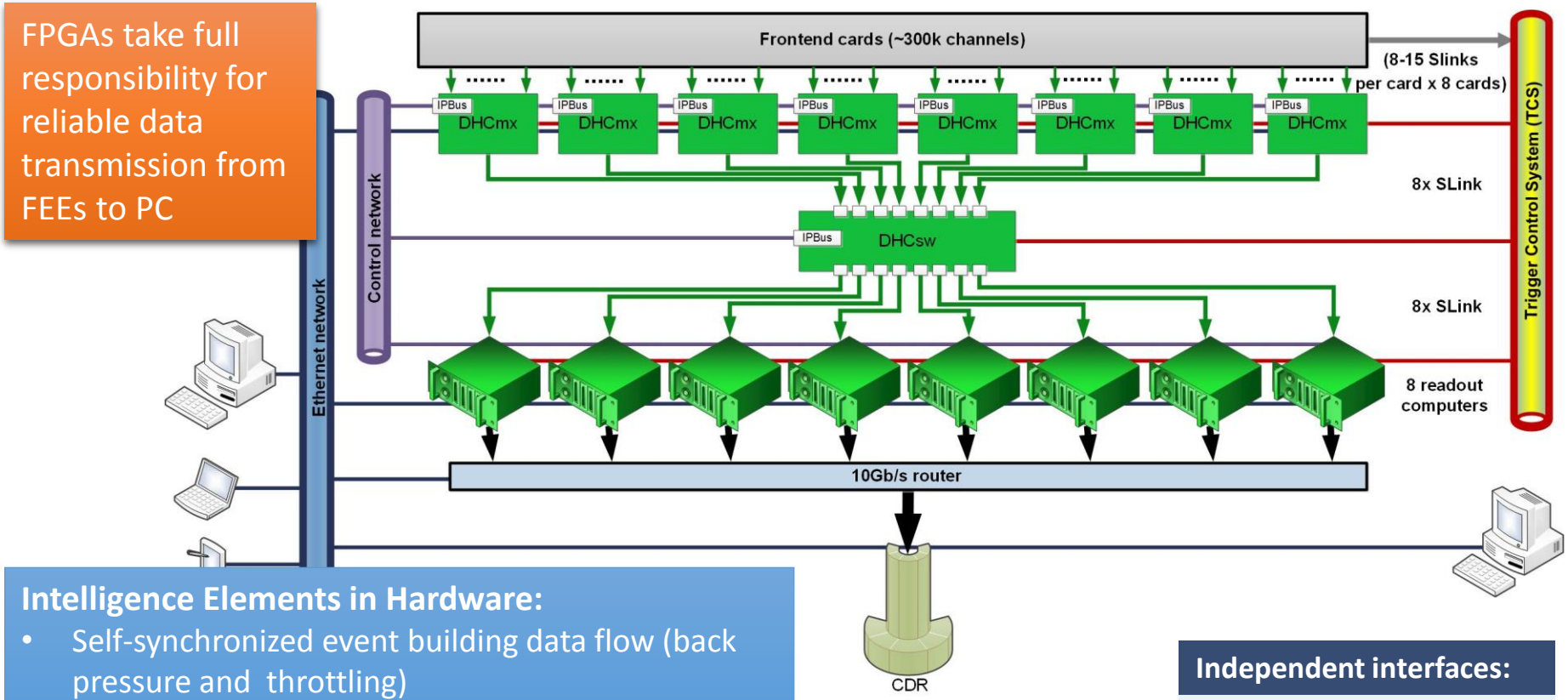
- 4GB RAM on each module => possibility to average data rate over spill cycle
- 1 GB/s sustained rate

Independent interfaces:

- synchronization → **TCS (Trigger Control System)**
- data flow (event building) → **SLINK**
- configuration and data flow control -> **IPbus**

iFDAQ – System Design

FPGAs take full responsibility for reliable data transmission from FEEs to PC



Intelligence Elements in Hardware:

- Self-synchronized event building data flow (back pressure and throttling)
- FEE error diagnostic and handling to prevent DAQ crash
  => monitoring and localization of failing FEEs
- Automatic resynchronization of FEEs
  => FEEs can be attached/detached at any time

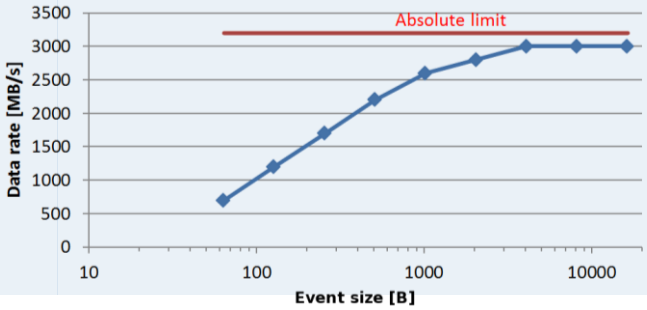
Independent interfaces:

- synchronization → **TCS (Trigger Control System)**
- data flow (event building) → **SLINK**
- configuration and data flow control -> **IPbus**

Data Handling Card (DHC)

AMC module

- **form factor:** AMC standard
- **FPGA:** Virtex6 XC6VLX130T
- **memory:** 4 GB DDR3 SDRAM
- **firmware:**
  - **DHCmx** 12:1 multiplexer [1]
  - **DHCsw** 8x8-switch
- **data rate:**



VME carrier card

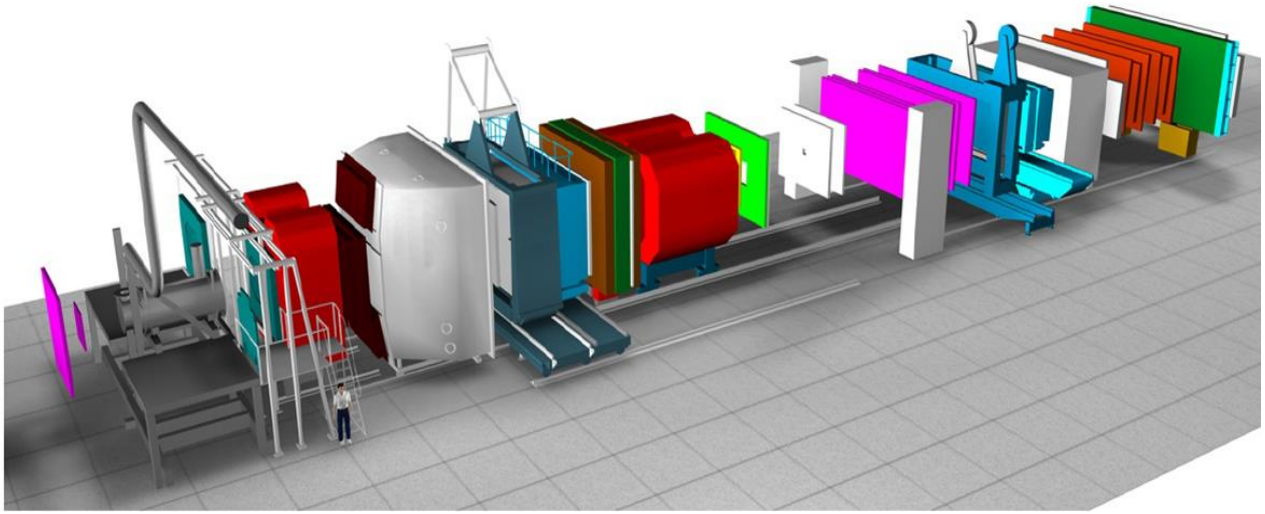
- **form factor:** 6 U VME
- **interfaces:**
  - TCS (Trigger Control System) receiver
  - 1 Gb Ethernet for control network (IPbus)
  - 16 serial data links (SLINK)
  - JTAG for backup programming of FLASH

1. Motivation and Concept of Hardware Event Building
2. Design of the intelligent FPGA-based DAQ
   - System
   - Hardware
   - (Firmware)
   - (Software)
3. **iFDAQ – integration and performance in COMPASS**
   - **Setup of COMPASS spectrometer**
   - **iFDAQ – Performance and Status in 2017 and 2018**
4. Switching Network Topology
   - Required Hardware Developments
   - Software Developments
5. Outlook and Conclusion

COMPASS – Overview

COmmon
Muon
Proton
Apparatus for
Structure and
Spectroscopy

- Fixed target experiment at SPS accelerator at CERN (M2 beamline)
- High intensity beams: $4 \cdot 10^7 \frac{\mu}{s}$; $2 \cdot 10^7 \frac{\text{hadrons}}{s}$
- Multi-purpose experiment
- Start of data-taking: 2001
- Since 2014: New DAQ with hardware event builder (iFDAQ)

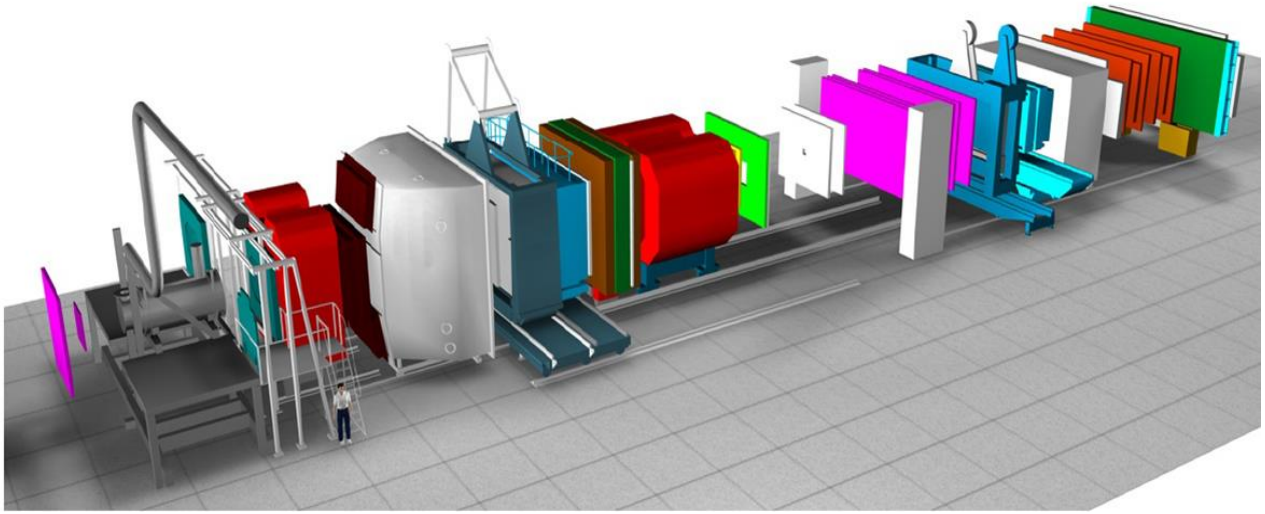


COMPASS – Spectrometer Setup

spectrometer facts	
• Length:	60m
• Amount of channels:	300.000
• Trigger rate:	30 kHz
• On-spill data rate:	1.5 GB/s
• Event size:	20-50 kB

- Fixed target experiment at SPS accelerator at CERN (M2 beamline)
- High intensity beams: $4 \cdot 10^7 \frac{\mu}{s}$; $2 \cdot 10^7 \frac{\text{hadrons}}{s}$
- Multi-purpose experiment
- Start of data-taking: 2001
- Since 2014: New DAQ with hardware event builder (iFDAQ)

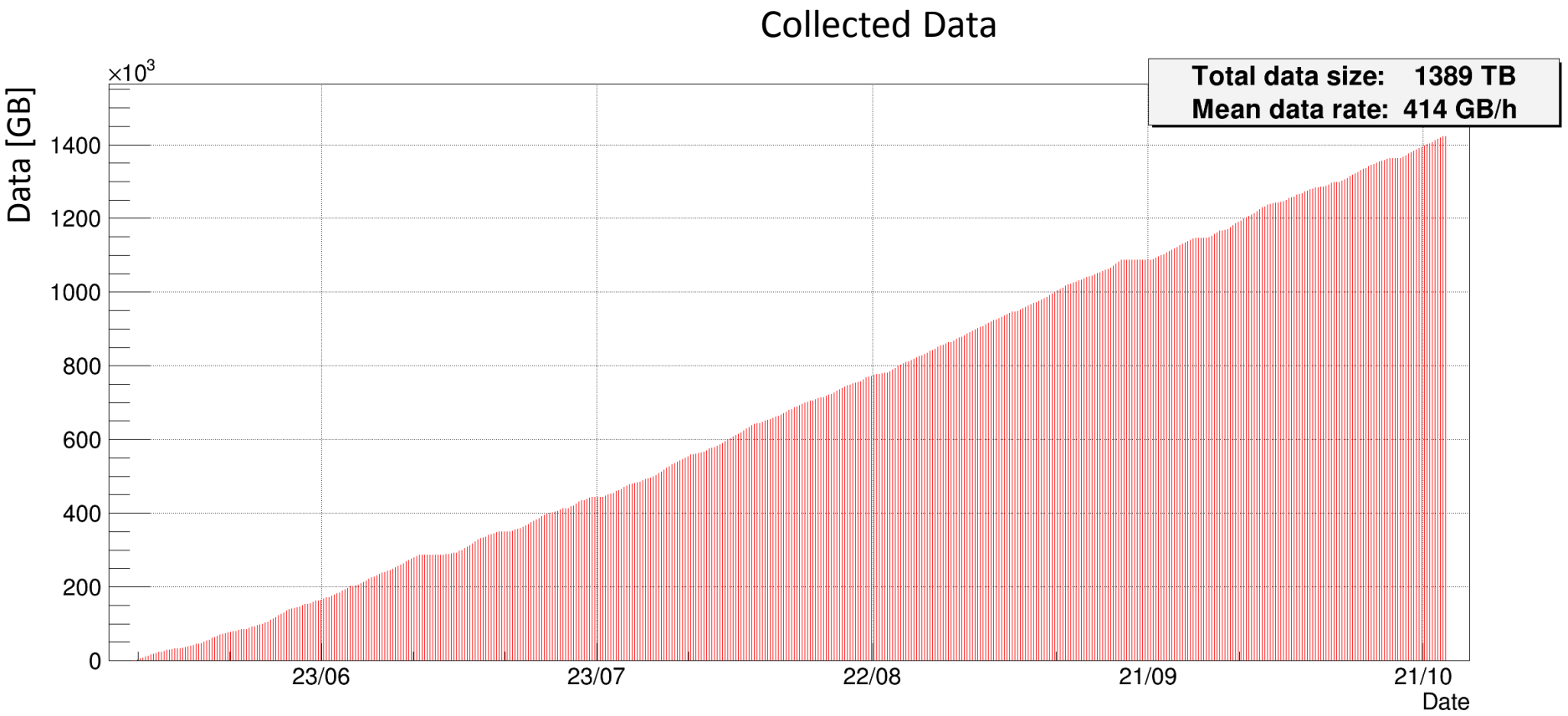
Source [2]



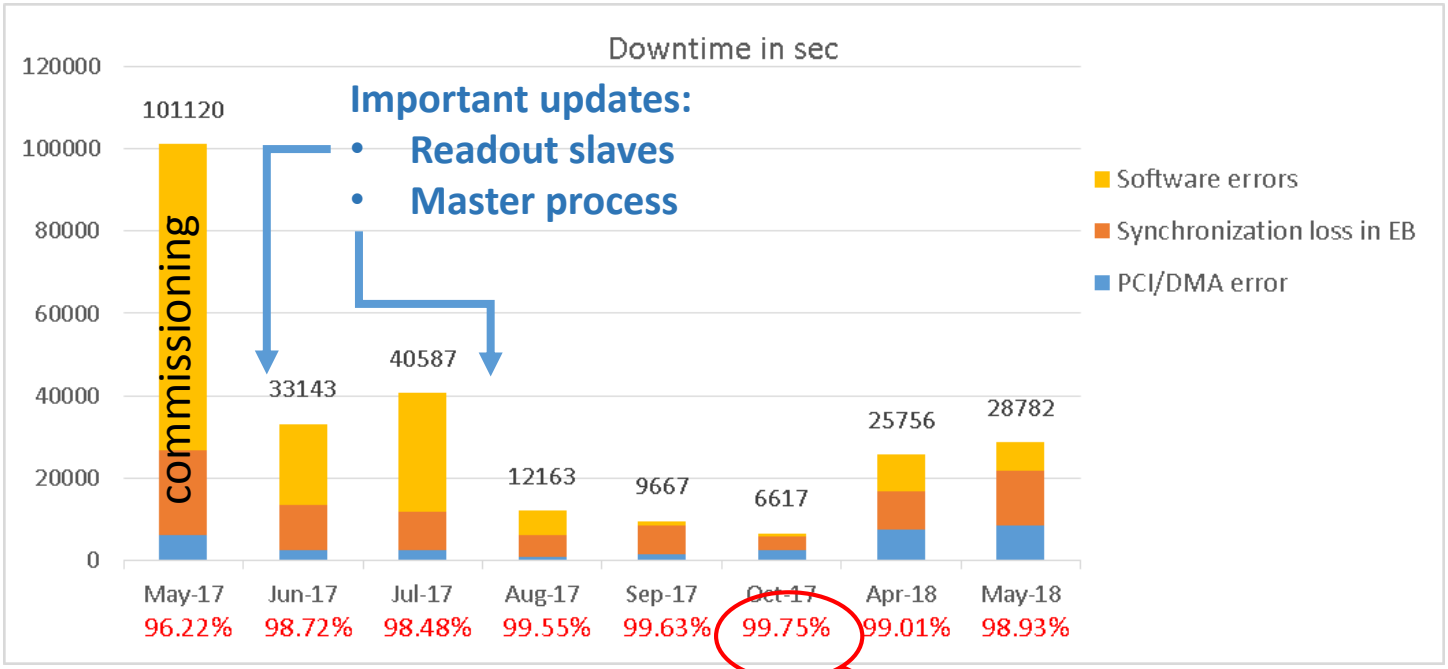
iFDAQ in COMPASS – Hardware Parts

- Very compact: 30 online PC in **former** DAQ
  Now: 1 VME crate (6-U) + one rack (8 computers)
- Highly flexible: Easily adaptable to different spectrometer setups (e.g. DVCS 2017 vs. Drell Yan 2018)

Performance – Accumulated Data 2017



Performance – Uptime in 2017



Effects:

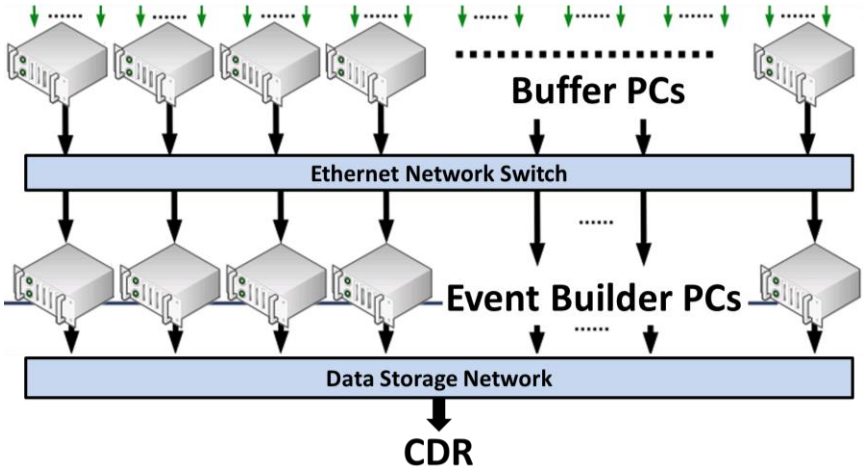
1. Detectors more stable during run than during commissioning phase
2. Upgrades of RCCAR software

Highly reliable

1. Motivation and Concept of Hardware Event Building
2. Design of the intelligent FPGA-based DAQ
   - System
   - Hardware
   - (Firmware)
   - (Software)
3. iFDAQ – integration and performance in COMPASS
   - Setup of COMPASS spectrometer
   - iFDAQ – Performance and Status in 2017
4. **Switching Network Topology**
   - **Required Hardware Developments**
   - **Software Developments**
5. Outlook and Conclusion

General Network vs. Point-to-Point

General Network (Ethernet)



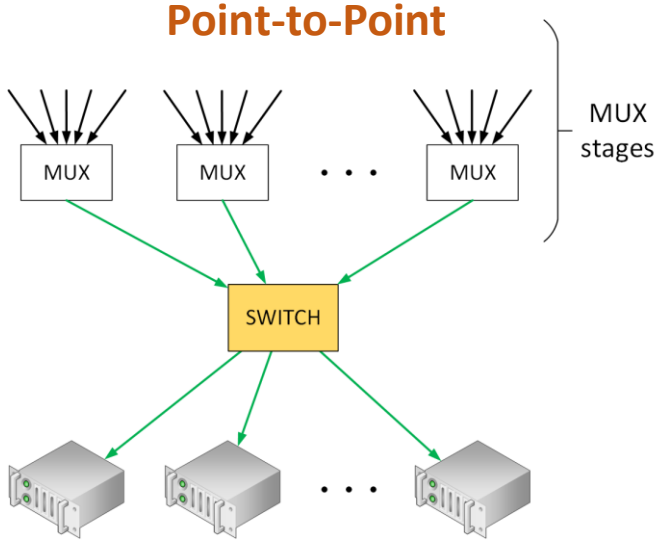
Advantages

- Easy integration of redundancy elements (traffic shaping according to load on nodes)
- Usage of mass-produced components and standards

Disadvantages:

- Throughput limited by EB-network switch
- Inefficient usage of max. bandwith due to:
  - Improper comm. pattern (N senders -> 1 receiver) => network congestion
  - Data overhead due to addressing etc.

Point-to-Point



Advantages:

- Independence of network switch
- Efficient usage of link bandwidth (no addressing etc.)
- High reliability

Disadvantages:

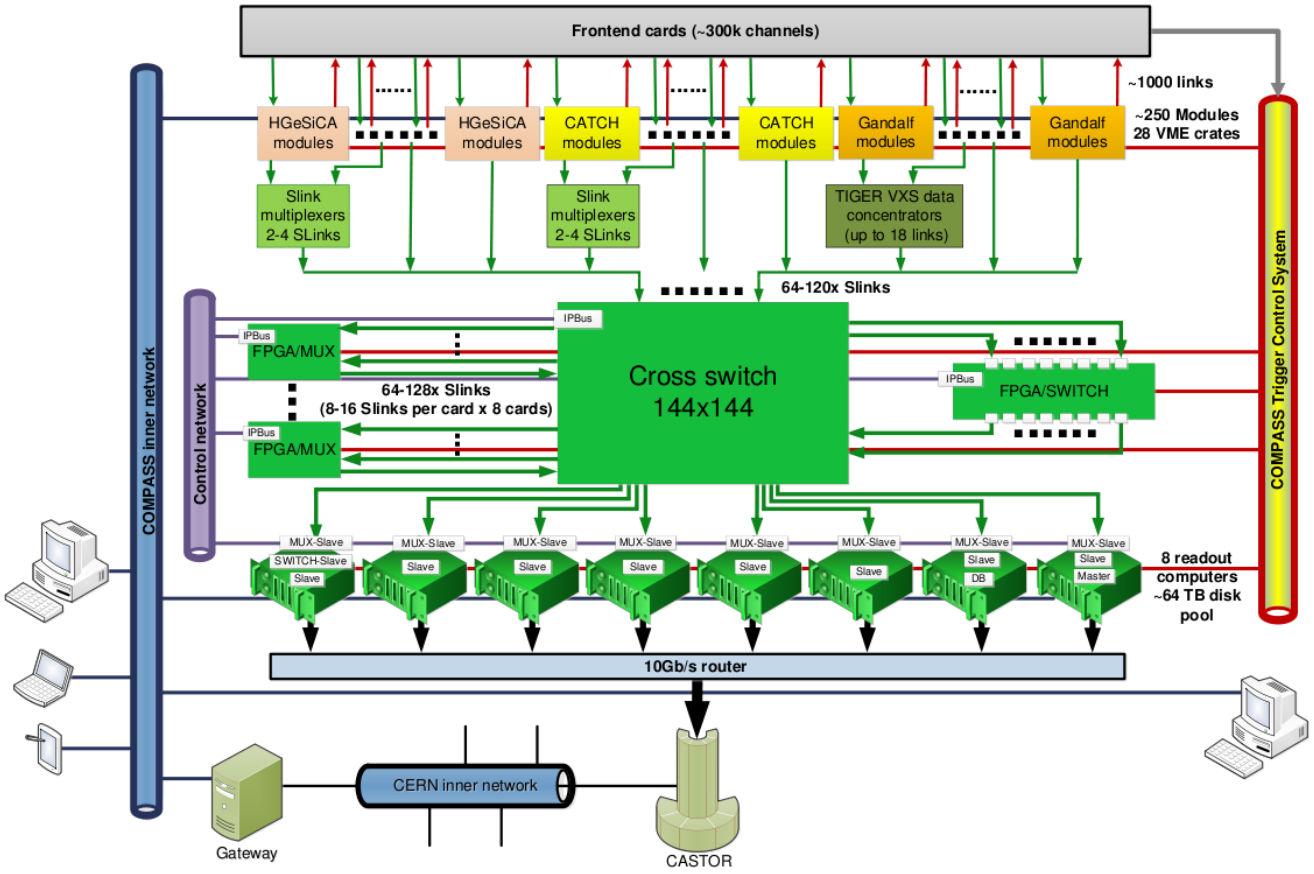
- Strong dependence on reliability of network nodes (no rerouting possibility in case of hardware failure)
- No possibility for dynamic network optimization (e.g. load balancing)

Crosspoint Switch - Integration

Crosspoint Switch

- connects:
  - FE electronics
  - DHCmx modules
  - DHCsw module
  - Spillbuffers
- purpose:
  - Ease of load balancing
  - System redundancy to compensate hardware failures

⇒ **provides fully customizable network topology**



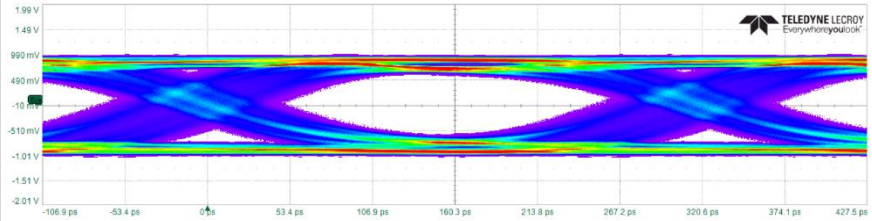
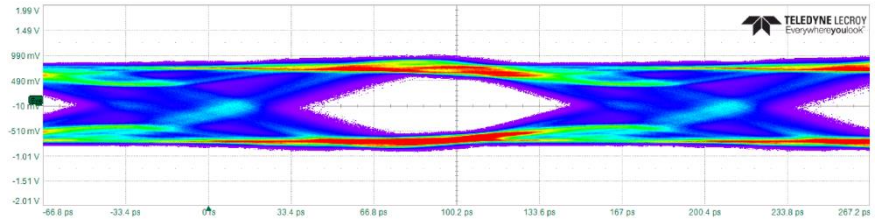
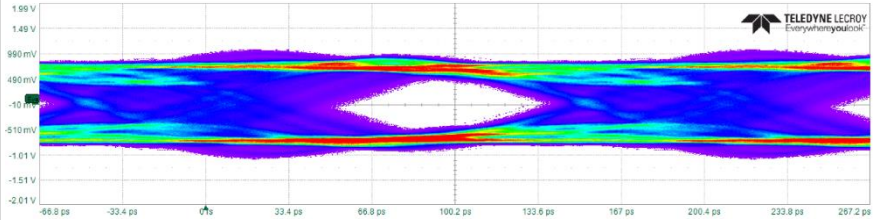
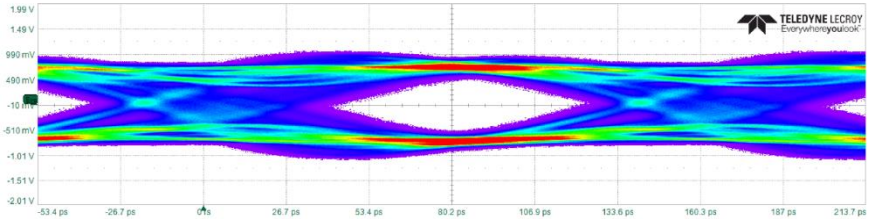
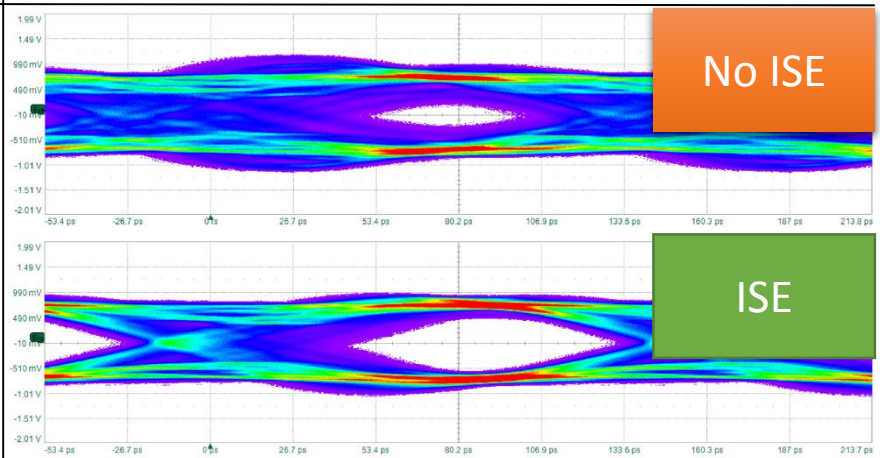
Crosspoint Switch – Hardware Design

Crosspoint Switch Components

- **interfaces:**
  - 12 x 12 channel CXP transceiver (MPO fiber connectors)
  - Ethernet for IPbus
  - JTAG
  - TCS (Trigger Control System) receiver
- **Switching and Control:**
  - **Vitesse VSC3144-02** – fully configurable 144x144, asynchronous, 6.5 Gbps crosspoint switch
  - **Xilinx Artix-7 FPGA** for switch control and monitoring

- **Interface FPGA – Crosswitch:**
  - 90 MHz, 11-bit parallel data bus
  - Multiple program assignments can be queued and issued simultaneously ⇒ fast programming (<< 1us)

Crosspoint Switch – Hardware Test

Line rate	Signal BEFORE switch	Signal AFTER switch
3.13		
5.00		
6.25		

Input Signal Equalization on Switch

1. Motivation and Concept of Hardware Event Building
2. Design of the intelligent FPGA-based DAQ
   - System
   - Hardware
   - (Firmware)
   - (Software)
3. iFDAQ – integration and performance in COMPASS
   - Setup of COMPASS spectrometer
   - iFDAQ – Performance and Status in 2017
4. Switching Network Topology
   - Required Hardware Developments
   - Software Developments
5. **Outlook and Conclusion**

Outlook

Ongoing development:

- Integration of crosspoint switch

  -> test of crosspoint switch successful and promising

  -> integration of the hardware for the crosspoint switch (results to be expected soon)

  -> upgrade of Software for automatic identification of malfunctioning hardware parts

Ideas for the Future:

- upgrade of TCS to bidirectional PON (**p**assive **o**ptical **n**etwork) with use of **U**niversal **C**ommunication **F**ramework (UCF) developed at TUM for on-the-fly reconfiguration of interconnections
- Minimizing of real-time processes

  -> direct writing of data onto SSD

Conclusion

- Improvements since commissioning of iFDAQ in 2014:
  - Increased reliability (Uptime around 99%)
  - Extended intelligence elements in software:

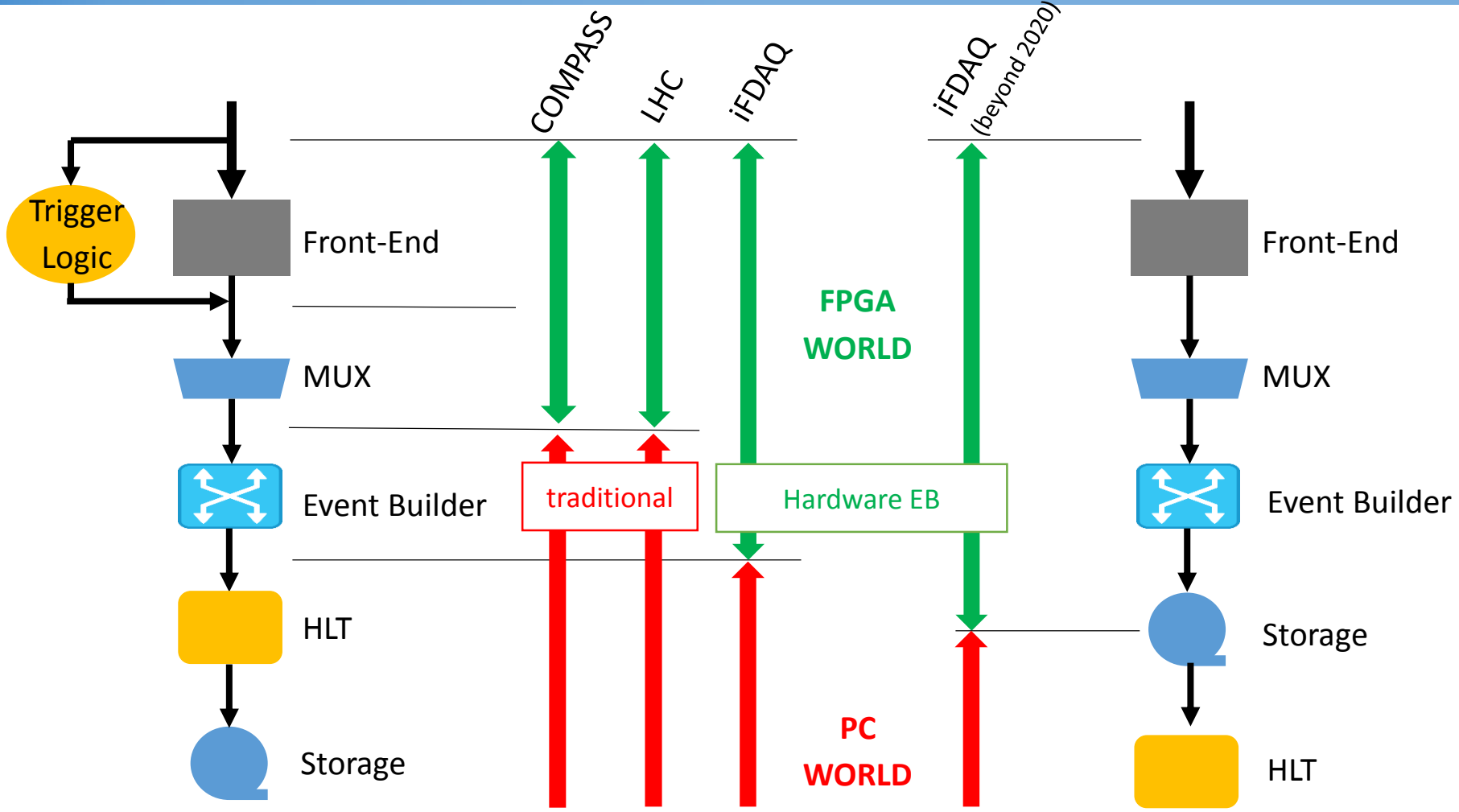
    -> Automatic safe stop of the run for self-recovery

    -> Continuously running
  - No event size limit due to upgrades in firmwares
- Performance in 2017:
  - Data rate: 91.7 MB/s (average)

    250 MB/s (in stable beam conditions)

    380 MB/s (peak sustained rate)
  - On-spill data rate: 1.5 GB/s
- iFDAQ transferred to other HEP experiment (NA64)

Possible DAQ Systems



THANK YOU for your Attention

References

[1] Y. Bai et al., *New data acquisition system for the COMPASS experiment*, in proceedings of *Topical Workshop on Electronics for Particle Physics (TWEPP)*, Lisbon Portugal September 2015

[2] COMPASS collaboration, P. Abbon et al., *The COMPASS experiment at CERN, Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip., 577(3):455-518, 2007.*

Backup slides

DHC – Data error handling in firmware

Data Consistency Check:

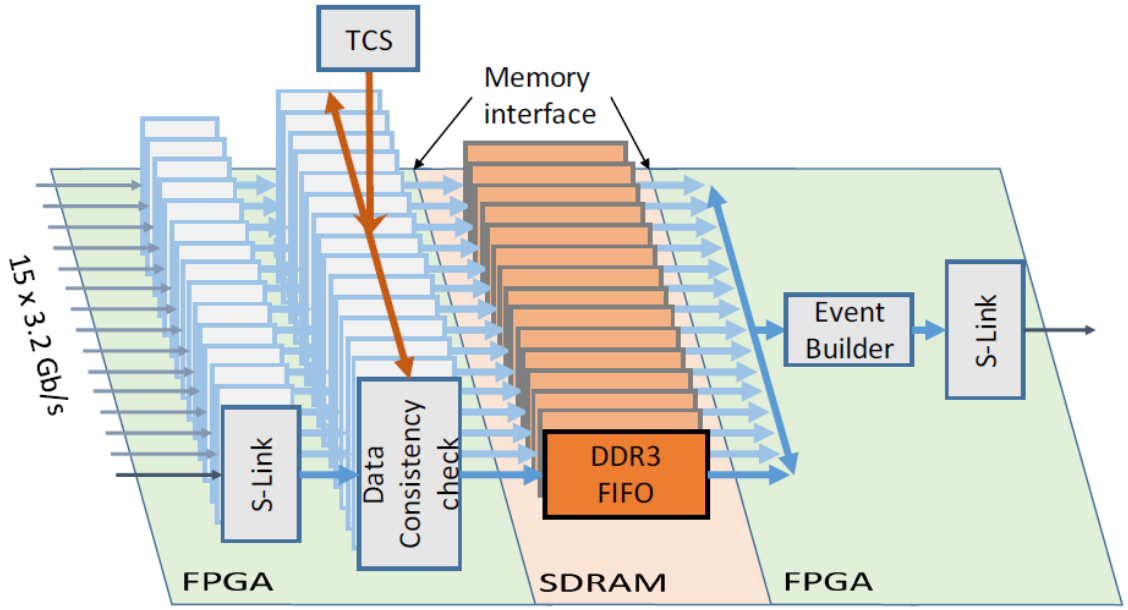
- Transmission errors detected by S-Link
- Truncation, i.e. mismatch between real and declared data block size
- Inconsistency of event label
- Missing data -> timeout

Error detected

Error handling

- Discarding/throttling of wrong data
- Adding of specific header for empty frame
- Setting error flag in local register (diagnostics of FEE errors)

Firmware – DHCmx



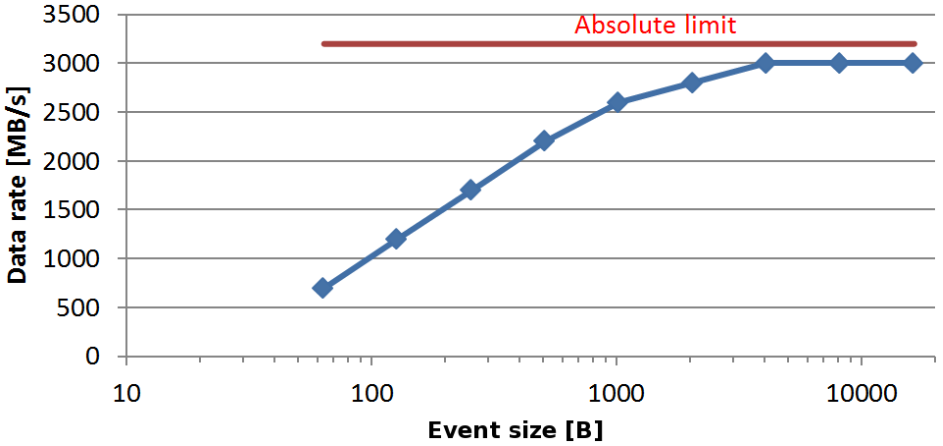
Data Consistency Check:

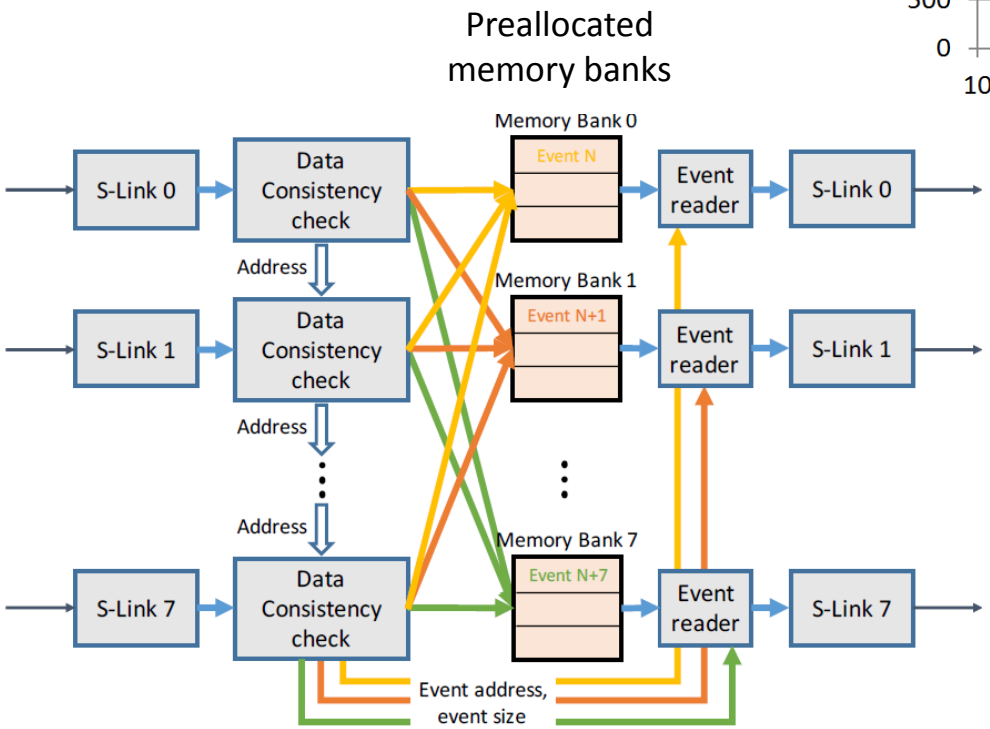
- Transmission errors detected by S-Link
- Truncation, i.e. mismatch between real and declared data block size
- Inconsistency of event label
- Missing data -> timeout

Error detected

- Discarding/throttling of wrong data
- Adding of specific header
- Setting error flag in local register

Firmware – DHCsw

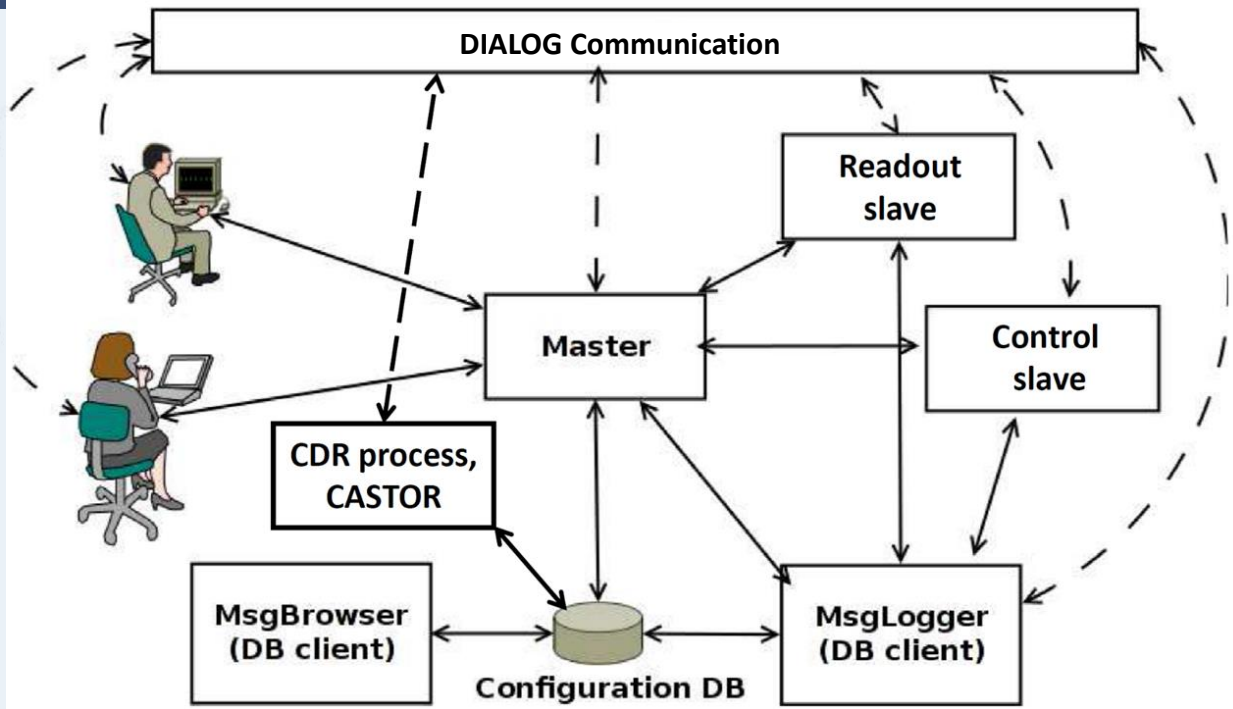




RCCAR software

Run Control, Configuration, And Readout Software (RCCARS):

- **Multilayer system** around master process
  - **master:**
    Middleman between GUI, DB, all other processes
  - **GUIs**
    display overall status of iFDAQ, control behavior of iFDAQ (when in Control mode)
  - **readout slaves (only real-time processes)**
    readout of data, analysis of events, error check, transformation into DATE-format, distribution to monitoring tools, writing to HDD
  - **control slaves**
    monitoring and control of hardware nodes
  - **message logger**
    collection of msgs from processes, storing msgs in DB
  - **message browser**
    display of msgs and support for advanced filtering
  - **Central Data Recording (CDR)**
    transfer of raw data to CASTOR, disk cleaning for new data
- Inter-process communication via **DIALOG library** (Custom-developed server/client communication)

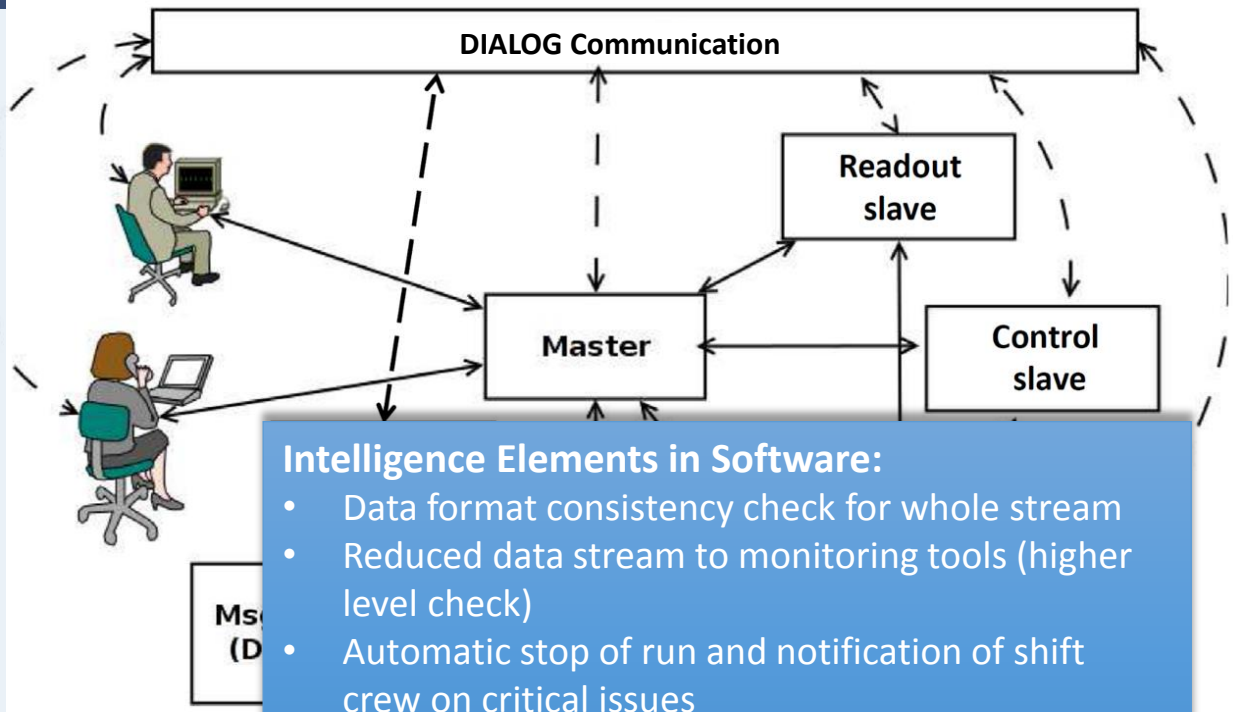


ICHEP 2017 Paris

RCCAR software

Run Control, Configuration, And Readout Software (RCCARS):

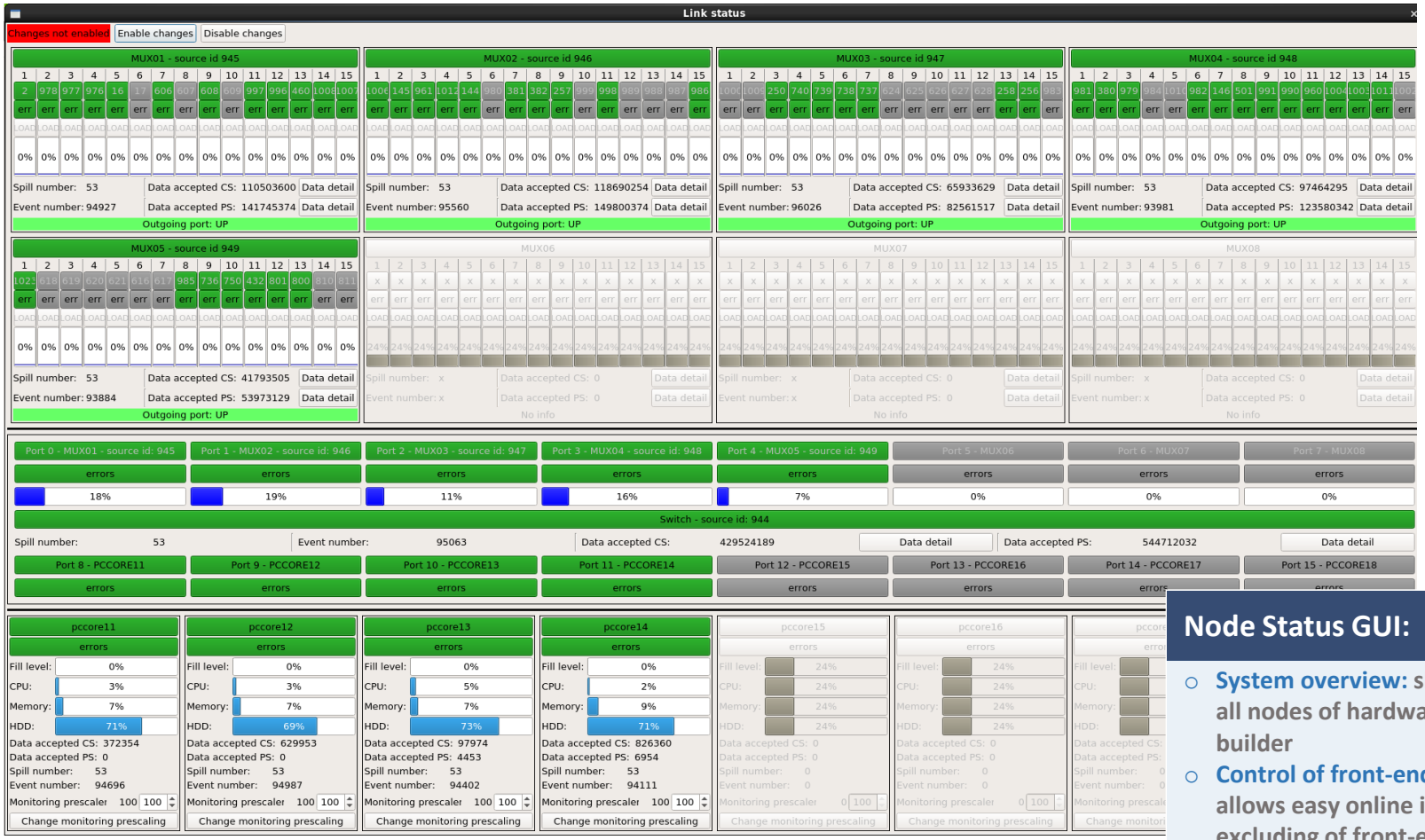
- **Multilayer system** around master process
  - **master:**
    Middleman between GUI, DB, all other processes
  - **GUIs**
    display overall status of iFDAQ, control behavior of iFDAQ (when in Control mode)
  - **readout slaves (only real-time processes)**
    readout of data, analysis of events, error check, transformation into DATE-format, distribution to monitoring tools, writing to HDD
  - **control slaves**
    monitoring and control of hardware nodes
  - **message logger**
    collection of msgs from processes, storing msgs in DB
  - **message browser**
    display of msgs and support for advanced filtering
  - **Central Data Recording (CDR)**
    transfer of raw data to CASTOR, disk cleaning for new data
- Inter-process communication via **DIALOG library** (Custom-developed server/client communication)



Intelligence Elements in Software:

- Data format consistency check for whole stream
- Reduced data stream to monitoring tools (higher level check)
- Automatic stop of run and notification of shift crew on critical issues
- Continuous data processing through EB and to monitoring tools (Start of run just enables data writing to disk)

Run Control & Node Status GUI



Node Status GUI:

- **System overview:** shows status of all nodes of hardware event builder
- **Control of front-end modules:** allows easy online including, excluding of front-end modules into DAQ and reloading of front-end modules

Hardware: Vitesse VSC3144

- 144 x 144 strictly non-blocking cross-point switch
- Up to 6.5 Gbps bandwidth per port
- No registers used in data path i.e. asynchronous data path => no restrictions on the phase, frequency, or signal pattern of any input (protocol independent)
- 45mm x 45mm 1072-pin BGA package
- Core programming on port-by-port basis OR simultaneous issuing of multiple queued assignments