

The Streaming readout consortium

Jan C. Bernauer

EIC Software Meeting, May 2019



RBRC
RIKEN BNL Research Center



Stony Brook
University



Massachusetts Institute of Technology

Glossary

Streaming readout means that detector data is digitized **continuously**, and there is **no hardware trigger** which starts the digitization / selects data.

Glossary

Streaming readout means that detector data is digitized **continuously**, and there is **no hardware trigger** which starts the digitization / selects data.

Alternatively: The information only flows downstream.

EIC eRD23: Streaming readout consortium

- **Catholic University of America:** S. Ali, V. Berdnikov, T. Horn, M. Muhoza, I. Pegg, R. Trotta
- **INFN Genova:** M. Battaglieri, A. Celentano
- **Stony Brook University / RBRC:** J. C. Bernauer
- **Massachusetts Institute of Technology:** D. K. Hasell, R. Milner
- **Thomas Jefferson National Accelerator Facility:** C. Cuevas, M. Diefenthaler, R. Ent, G. Heyes, B. Raydo, R. Yoshida

Additionally many regulars like Martin Purschke (BNL), Marco Locatelli (CAEN), Jin Huang (BNL), Esko Mikkola (Alphacore),

→ We welcome new members! ←

A thought experiment

Imagine: You are a shop keeper. Every night, somebody eats all the candy. What do you do?

A thought experiment

Imagine: You are a shop keeper. Every night, somebody eats all the candy. What do you do?

Put up a photo camera with a photocell trigger, laser + mirrors?

A thought experiment

Imagine: You are a shop keeper. Every night, somebody eats all the candy. What do you do?

Put up a photo camera with a photocell trigger, laser + mirrors?



A thought experiment

Imagine: You are a shop keeper. Every night, somebody eats all the candy. What do you do?

Put up a photo camera with a photocell trigger, laser + mirrors?



A thought experiment

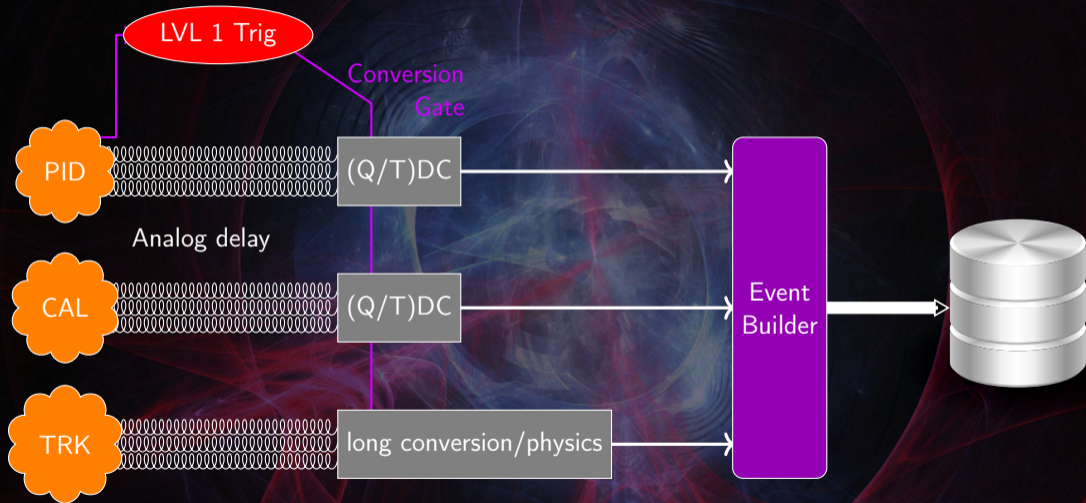
Imagine: You are a shop keeper. Every night, somebody eats all the candy. What do you do?

Put up a photo camera with a photocell trigger, laser + mirrors?

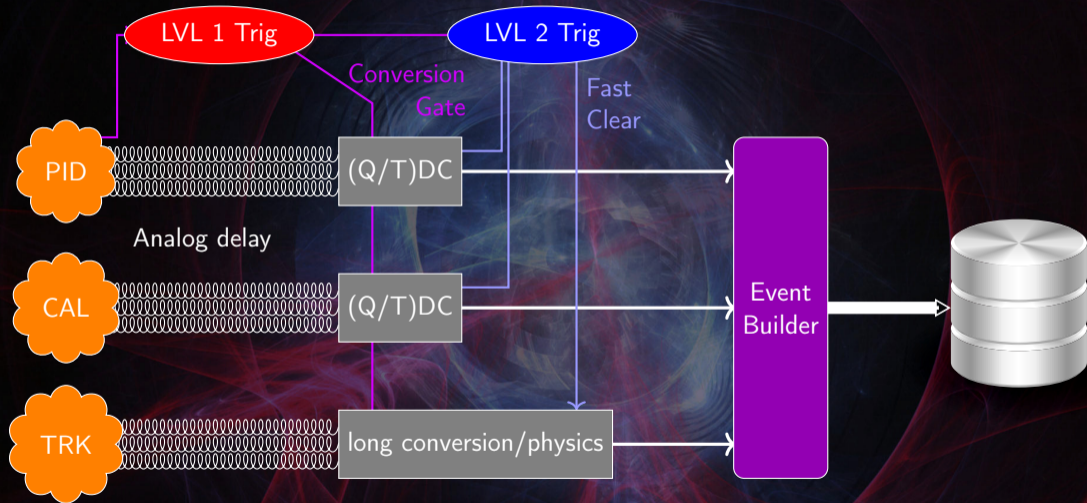


Why not put up a video camera instead?

A classic readout system



A classic readout system



Reasons we ever did hardware triggers

Originally:

- QDCs needed a gate
- xDCs all used slope techniques to convert. Slow!
- TDCs actually measured durations (start-stop)

Reasons we ever did hardware triggers

Originally:

- QDCs needed a gate
- xDCs all used slope techniques to convert. Slow!
- TDCs actually measured durations (start-stop)

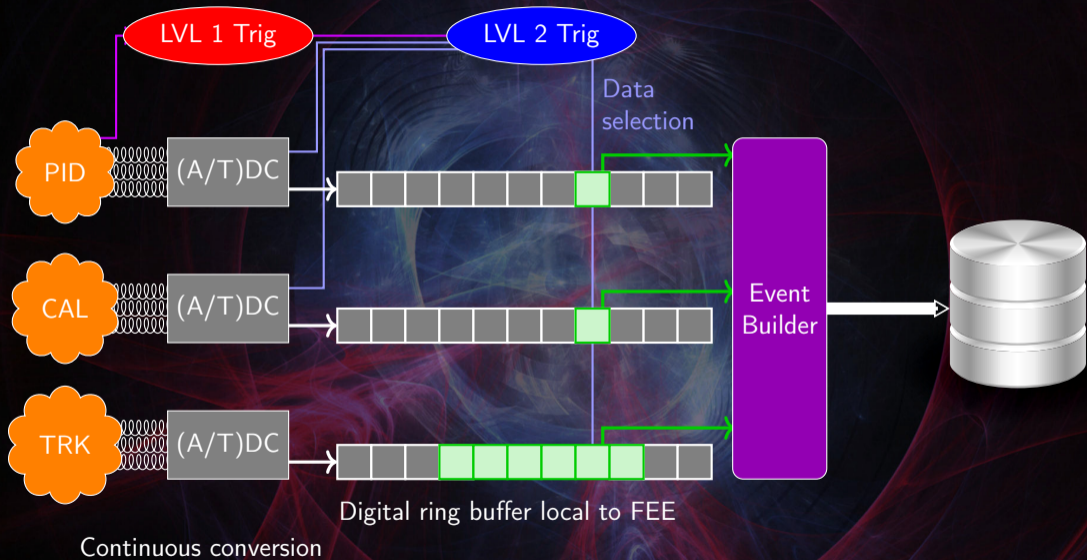
Later:

- Data rate limit on buses, in networks and to tape.

Hardware we need for streaming readout

- Continuously converting xDCs
- Fast buses
- Fast networks
- Fast/vast storage

A more modern triggered readout



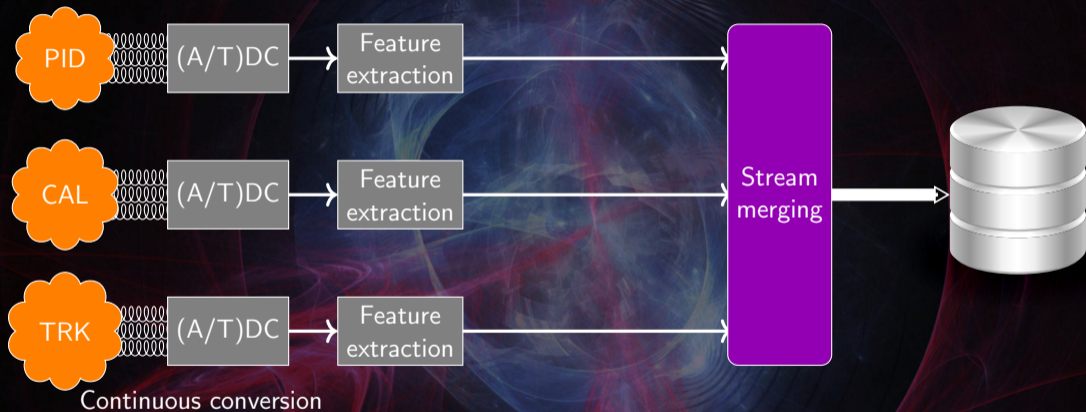
Hardware we need for streaming readout

- Continuously converting xDCs \Leftarrow Modern setups have that!
- Fast buses
- Fast networks
- Fast/vast storage

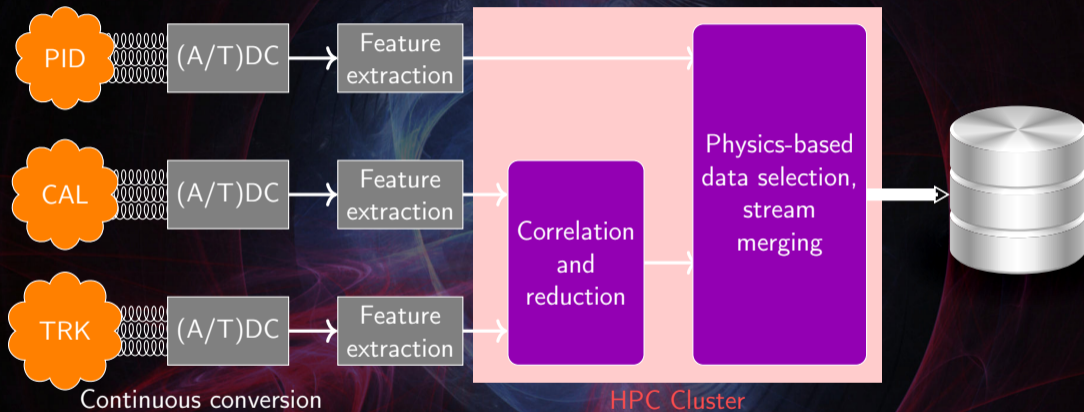
Hardware we need for streaming readout

- Continuously converting xDCs
- **Fast buses** \leftarrow Point-to-point networking and switches
- **Fast networks** \leftarrow 1Gbit/s \sim free, 10Gbit/s and 100Gbit/s cheap-ish
- **Fast/vast storage?!?**

A simple streaming example



A complex streaming example



Storage requirements

- A streaming readout with a "software defined trigger" will produce **the same rate** as the hardware triggered equivalent.

Storage requirements

- A streaming readout with a "software defined trigger" will produce **the same rate** as the hardware triggered equivalent.
- But: More information available:
 - can write better data selection
 - can suppress noise better
- And: Can only save preprocessed information, drop raw data

Storage requirements

- A streaming readout with a "software defined trigger" will produce **the same rate** as the hardware triggered equivalent.
- But: More information available:
 - can write better data selection
 - can suppress noise better
- And: Can only save preprocessed information, drop raw data
- Can save more, without dead-time

Storage requirements

- A streaming readout with a "software defined trigger" will produce **the same rate** as the hardware triggered equivalent.
- But: More information available:
 - can write better data selection
 - can suppress noise better
- And: Can only save preprocessed information, drop raw data
- Can save more, without dead-time

Can stuff more physics in same space!

Different experiment work in different regimes

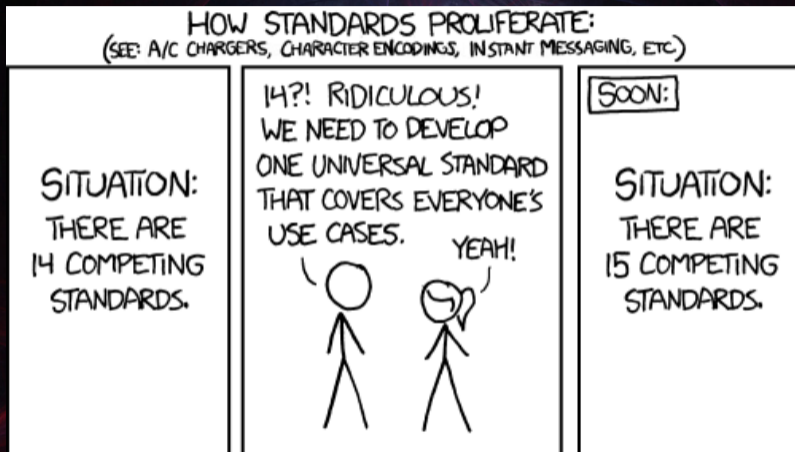
- LHC ALICE and LHCb: Move analysis online to save only analyzed data
- DarkLight / CBM: Software defined trigger to only save relevant data
- sPHENIX: Kill deadtime for TPC readout
- EIC? It depends

- **ePHENIX**: Most likely possible to save everything after zero suppression. Less rate than sPHENIX!
- **TopSide**: Will need streaming readout to reduce data rate.
- **Benefits of streaming readout:**
 - No online event builder
 - Less complex hardware, less tricky timing etc.
 - Better online monitoring
 - Faster to-publication times
 - Convergence of online and offline

Democratization of DAQ

- Streaming readout moves many tasks from hardware to firmware to software
- More people can contribute.
- "Trigger" can be changed after data taking. Data mining!

We want to push a standard



(Source: XKCD)

Why?

- There is no standard so far. HEP has many resources to throw at problem, **NP doesn't**
- Want to **pool resources**
- Want to solve a pain point: people bring their own detectors to national labs.
- Better educate students:
 - What you have at your institute at home is a small version of what you have at the labs.
 - **Software** instead of hardware
- **Avoid** vendor lock-in
- Use road to EIC as driving force, but profit for **ALL** labs and exps.

Implications for analysis

To exploit benefits of streaming readout optimally, **move as much as possible of the analysis online**

- **time-stamped streams** instead of full events
- **specialized nodes** instead of monolithic analysis program
- **heterogen hardware** FPGAs, GPUs, etc for specialized functions.

Software (adjacent) tasks in streaming readout

From low-level to high-level

- Definition of wire / disk protocol **detector agnostic**
- FEE firmware **modularized**
- Server to host **operators** on the data streams and a framework to write them in.
language agnostic / multiple frameworks?
- Device, node, and link configuration **Can we use DNS?**
- Orchestration **must scale**
- Time-stamp oriented analysis as a network of operators
- MC which can produce streaming-like data with correct timing structure.