Real-time Level-1 Trigger Data Scouting at CMS using CXL Memory Lake

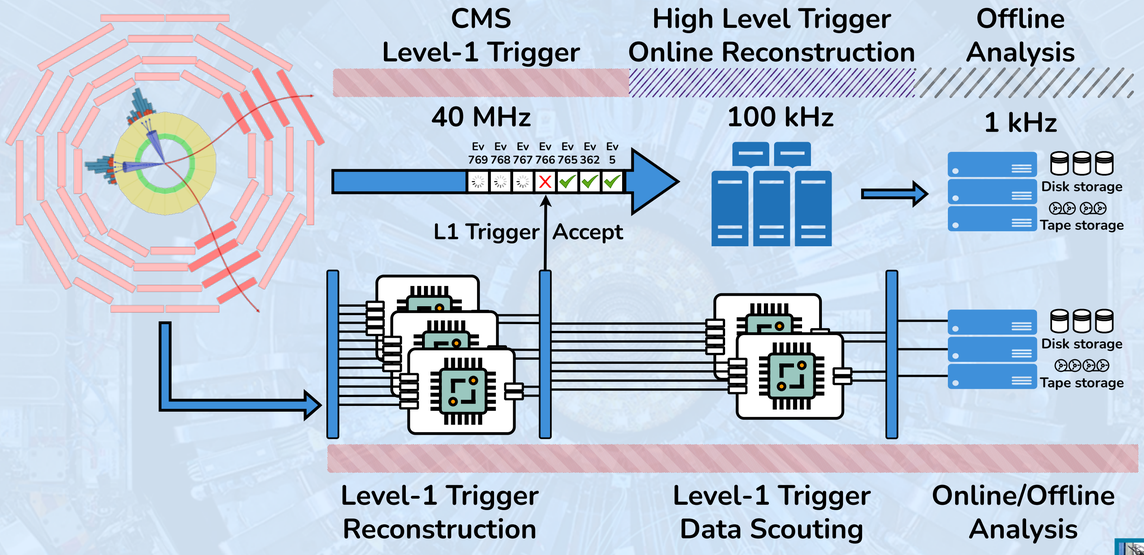
Giovanna **Lazzari Miotto**, Thomas **James**, Emilio **Meschi**, for the **CMS L1 Data Scouting Team**
CERN, Geneva, Switzerland

What is CMS L1 Data Scouting (L1DS)?

Full readout at the **LHC**'s **40 MHz rate** is infeasible. CMS employs a 2-tier trigger system, pre-selecting events with a fast, FPGA-based **Level-1 (L1) Trigger that fires at 100 kHz**, guided by ***trigger primitives*** and ***physics signatures that are inherently biased***. CMS Phase 2 will enhance L1 resolution to cope with HL-LHC pileup.

Thus, **analysing L1 trigger primitives at 40 MHz can pave the way for new physics, bunch-crossing correlations** and **diagnostics**.

L1 Data Scouting collects muon, calorimeter and BRIL data since LHC Run 3 (1.2 PB in 2024) [1, 2].



Detector-to-analysis workflow with the CMS Trigger System and L1 Data Scouting. Adapted from [1].

VCU128 FPGA boards

Scouting rack at CMS Underground Service Cavern

How a memory lake can help

Our demonstrator currently stages data on a ramdisk exposed via NFS to processing unit daemons [3]. We need a novel approach for **real-time, zero-bias analysis** in **Run 4 and beyond:**

- **Heterogeneous computing** (more FPGA and GPU accelerators before CMS Phase 2)
- **More scalable ephemeral storage**
- Close-to-DRAM **low-latency, high-bandwidth**

⟹ **Memory lake architectures (and new memory standards) can provide that!**

Why Compute Express Link (CXL)

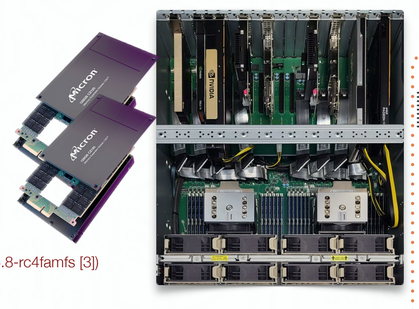
Open standard [4] for **high-bandwidth memory sharing** with **heterogeneous devices**

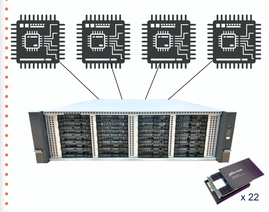
- **Single interface** for CPUs, GPUs, FPGAs, smart NICs
- **Shared**, **fabric-attached memory pooling**
  - Less data copying and movement
- **Hardware-level cache-coherence**
  - **Shared data** in cache is invalidated if stale
  - **Cache-line granularity**: *O*(64 bytes)
- **PCIe 5.0 physical layer** with extended protocols:
  `cxl.io`: modified, PCIe 5.0 **load/store interface**
  `cxl.mem`: host → device (**memory expansion**)
  `cxl.cache`: device → host (**accelerator caching**)

Our prototype at CMS

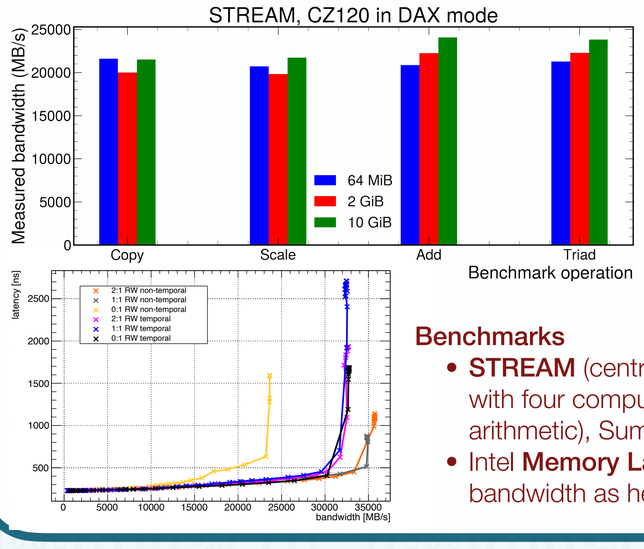
Supermicro server at CMS' Online Computing Centre (OLC), with:

- 2x **Micron CZ120 256GB** (*engineering samples*)
  - **'Type 3' CXL memory expansion modules**
  - PCIe 5.0, x8 data lanes
  - **36 GB/s peak bandwidth** (R/W)
- 2x AMD EPYC 9454 Genoa 48-Core @ 2.75 GHz, SMT
- 2x 96 CPU, 460 GB/s peak bandwidth per socket
- 2x 256 GB L3 shared cache
- 24x 16 GB DDR5-4800 RDIMM
- RHEL 9.3 (kernel 6.8-rc4famfs [3])
- AutoNUMA on
- libcxl 78





Falcon C5024 memory chassis



Arriving 2024-25

- H3 Falcon C5024 memory chassis
  - With Xconn XC50256 **CXL switch**
  - Up to **22 E3.S CXL memory modules**
  - **5.5 TB memory lake**
- CXL FPGA-base custom boards for NMC
- More CXL-compliant HW to come in ~months

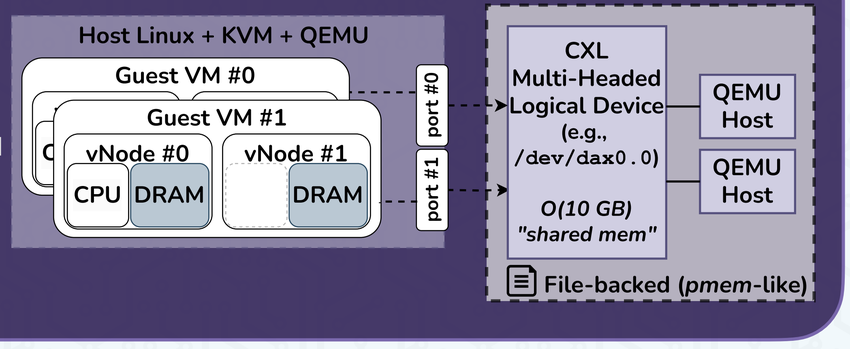
Until then... **emulation tools**

Benchmarks

- **STREAM** (centre left): sustained main-memory bandwidth as DAX with four computation workloads. Copy (transfer), Scale (transfer + arithmetic), Sum (+ load/store), Triad (chained MUL+ADD).
- Intel **Memory Latency Checker** (**MLC**) (lower left): latency and bandwidth as headless NUMA domain for various RW patterns

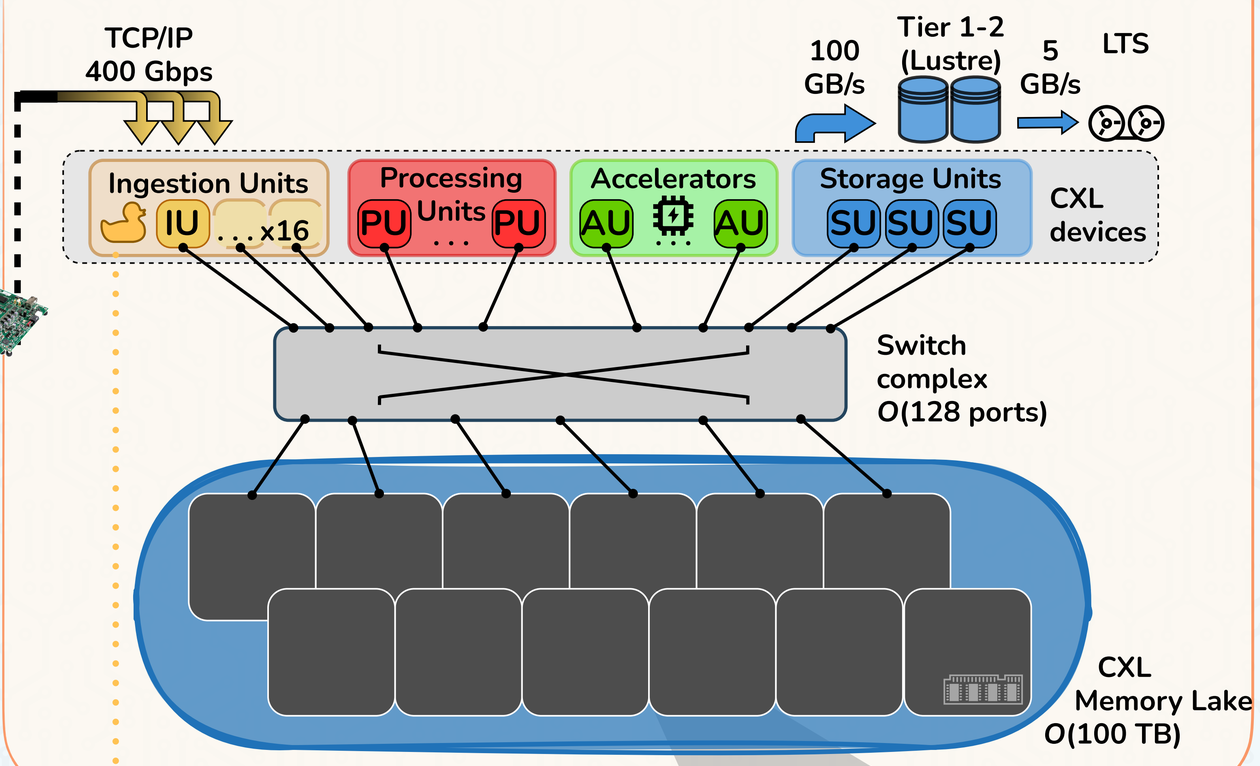
Emulation of CXL topologies

- QEMU-based [5]
- Multiple memory modules interconnected by an emulated switch ("pooling")
- Interoperates with **kernel-based virtual machines** (KVMs) as independent hosts
- Approximates behaviour of cache-coherent CXL access

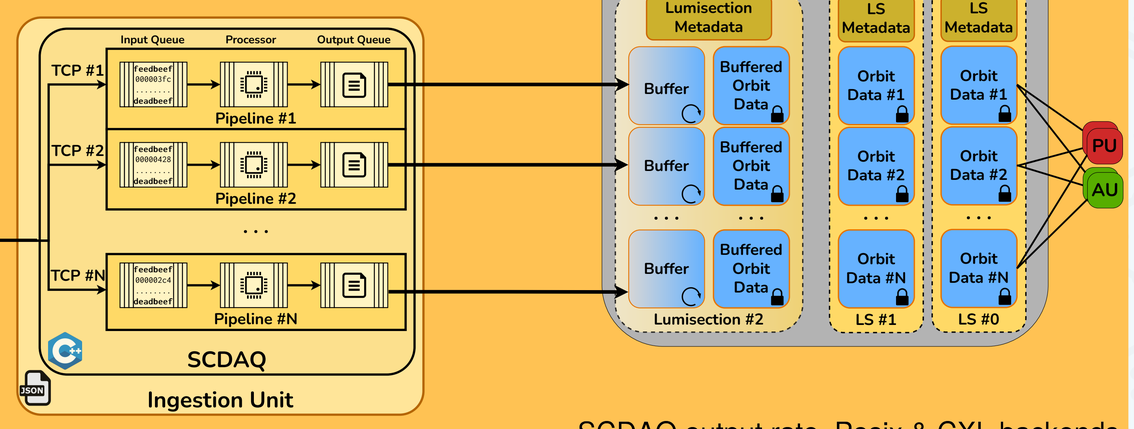


A CXL memory lake for 40 MHz analysis in the HL-LHC era

A **shared memory lake** will pool **several TBs of memory for efficient, zero-copy access**, and **cache-coherent online analysis at 40 MHz by heterogeneous computing accelerators**

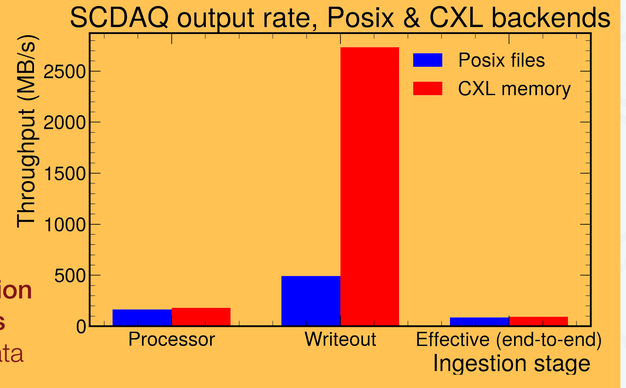


SCDAQ: scouting data acquisition



Scouting Data Acquisition (SCDAQ) [6]

- **Intel TBB**-based ingestion and preprocessing software on L1DS DAQ units
  - TCP/IP receive-buffering (from FPGAs)
  - Zero-suppression at the bunch-crossing level
  - Lumisection / orbit metadata



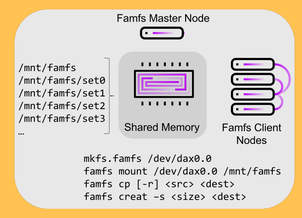
How to manage ephemeral data?

Prototype: **minimal, in-memory directory organization**

- Metadata descriptors with memory range `iovecs`
- Non-contiguous lumisections, contiguous orbit data
- Lowest-level: flexible development but onerous
- DAX: cache-bypassing direct access with huge-page alignment (2 MiB)

Exploring: **a more expressive filesystem - Famfs**

- *Fabric-attached memory filesystem*
- Lightweight, **FS-DAX**-like FS for shared, disaggregated contexts
- Multi-host mounting
- POSIX IO API-compatible, optimized for **DAX** + `mmap`
- Kernel patch proposed, FUSE version being developed



Famfs scheme and usage [7]

Outlook

We propose a **CXL-based memory lake architecture for CMS L1 Data Scouting (L1DS):**

- **Heterogeneous, disaggregated online processing of L1 primitives** (targeting LHC Run 4+)
- **Prototype installed at CMS OLC**: **512GB of CXL memory** with near-DRAM latency and bandwidth
- Significant extension and rewrite of L1DS data-taking software, with new **buffered DAX and Famfs-based output backends** with improved write throughput currently not exploited due to processing bottlenecks
- New QEMU-based emulation testbed to simulate CXL topologies

Next steps

Milestones for a full, **CXL-based detector-to-analysis demonstrator at CMS** within the next months:

1. Integrate **DAX+Famfs SCDAQ** with downstream CMSSW-based **processing units**
2. Connect **CMS OLC prototype** to spare **L1DS links at CMS USC**
3. Introduce **CXL `Type 2` accelerators** and **validate cache-coherence in concurrent settings**

References

[1] Ardino et al. "***Design and perspectives of the CMS Level-1 trigger Data Scouting system***". NIMA vol. 1067. Oct 2024. [doi]
[2] Sieder et al. "***CMS L1 Data Scouting for HL-LHC***". CHEP 2024. [doi]
[3] Migliorini et al. "***An online data processing system for the CMS Level-1 Trigger data scouting demonstrator***". CHEP 2024. [doi]
[4] **CXL 3.1 Specification**. https://computeexpresslink.org/cxl-specification
[5] **QEMU**. https://www.qemu.org
[6] **SCDAQ repository**: *https://gitlab.cern.ch/scouting-demonstrator/scdaq*
[7] **Famfs Shared Memory Filesystem Framework.** *https://github.com/cxl-micron-reskit/famfs*

This project benefitted from funding provided by the **CERN openlab** - **Micron collaboration.** *Thanks to our external collaborators at Micron and to* **Guilherme Paulino** *(2024 openlab summer student).*

Giovanna **Lazzari Miotto**
Software Engineer, **CERN EP-CMD**
giovanna.lazzari.miotto@cern.ch