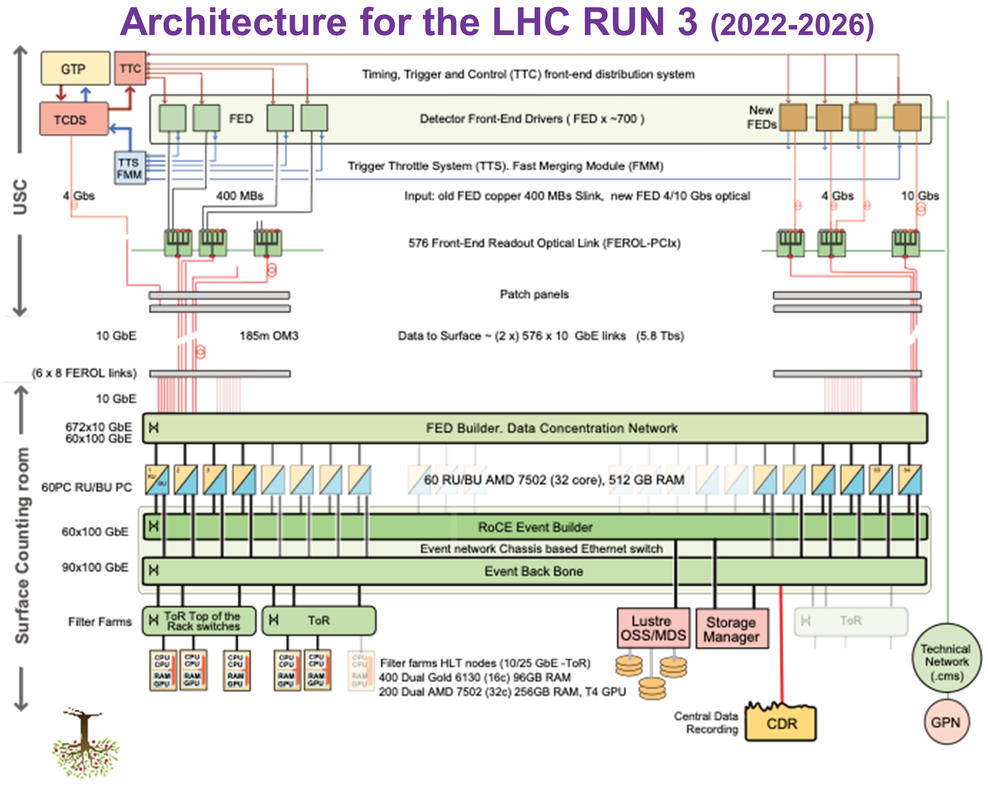
Evaluating a File-based Event Builder to enhance the Data Acquisition in the CMS Experiment

J. Alawieh [1], K. Arutjunjan [1], M. Bacharov Durasov [1], U. Behrens [2], A. Bocci [1], J. Branson [3], P. M. Brummer [1], J. A. Bugajski [1], E. Cano [1], S. Cittolin [3], A. Corominas I Mariscot [1], G. L. Darlea [4], C. Deldicque [1], M. Dobson [1], A. Dvorak [1], C. Emmanouil [1], A. Gaile [1], D. Gigi [1], F. Glege [1], G. Gomez-Ceballos [4], P. Gorniak [1], J. Hegeman [1], G. Izquierdo Moreno [1], T. O. James [1], T. Jayakumar [1], W. Karimeh [1], R. Krawczyk [2], W. Li [2], K. Long [4], F. Meijers [1], E. Meschi [1], M. Migliorini [1,7], S. Morović [3], B. J. Odetayo [1,5], L. Orsini [1], C. Paus [4], **A. Petrucci** [3], M. Pieri [3], D. S. Rabady [1], A. Racz [1], T. Rizopoulos [1], H. Sakulin [1], C. Schwick [1], D. Šimelevičius [1,6], P. Tzanis [1], C. Vazquez Velez [1], P. Žejdl [1]

[1] **CERN**, Geneva, Switzerland; [2] **Rice University**, Houston, Texas, USA; [3] **UCSD**, San Diego, California, USA; [4] **MIT**, Cambridge, Massachusetts, USA; [5] **University of Benin**, Benin City, Nigeria; [6] **Vilnius University**, Vilnius, Lithuania, [7] **University of Padova**, Padova, Italy

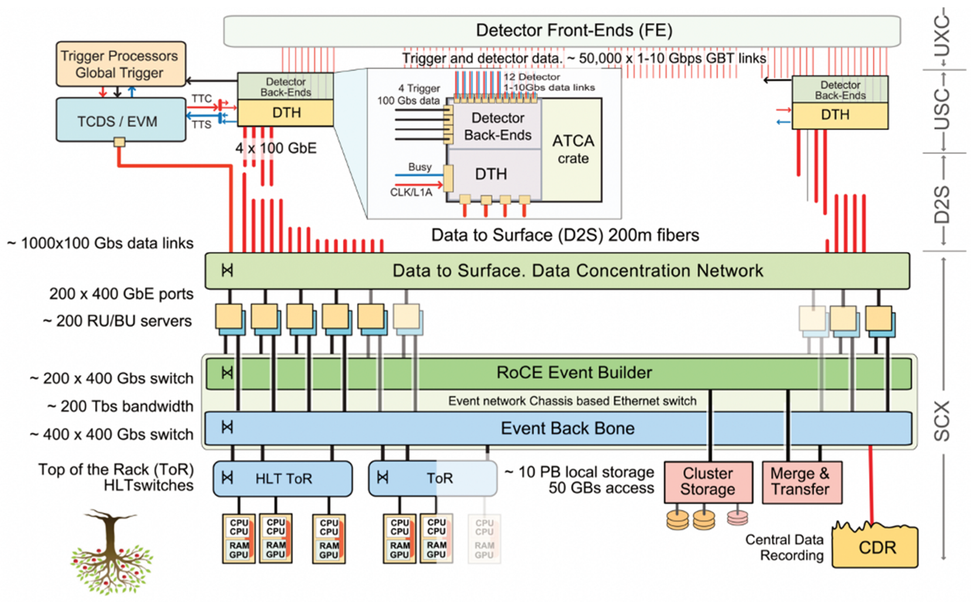
CMS Data Acquisition System

Architecture for the LHC RUN 3 (2022-2026)



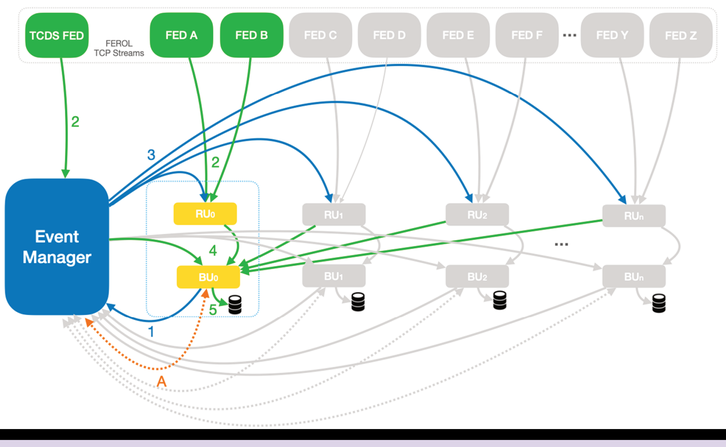
The **event builder** in the CMS DAQ is responsible for assembling **events** or **orbits**. Both the current and future DAQ architectures make use of state-of-the-art network technologies, employing Ethernet switches that support **RoCE** protocols. The **DAQ front-end** hardware is **custom-designed**, using a **reduced TCP/IP** protocol implemented in **FPGA** for reliable data transport between custom electronics and commercial computing hardware.
The **files** created by the event builder, each containing **around 100 events** or **one orbit**, are distributed over **NFS** among filter farm processes using **RAM disks**.

Architecture for the LHC Phase-2 (starting in 2030)



DAQ Building Information	LHC RUN 3	LHC Phase-2
L1 accept rate (maximum)	115 kHz	750 kHz
Event Size	2 MB	8.4 MB
Event network traffic	1.84 Tb/s	51 Tb/s
D2S modules	FEROL	DTH-400/DAQ-800
D2S module Ethernet ports	10 GbE	100 GbE
Number of D2S source ports	650	850
D2S network, EVB ports	100 GbE	400 GbE
EVB Network	100 GbE RoCE	400 GbE RoCE
Number of EVB nodes	~ 60	~200
Aggregation	Super-fragment	LHC Orbit (~67 Super-fragments)

Event Builder (EVB) Protocol



1) BU requests to EVM N resource to build
2) Asynchronously Event fragments from FEROL to EVM & RUs
3) EVM schedule & Control:
   A. EVM receives enough event fragment from TCDS FED to satisfy N of request from BU
   B. EVM sends a message to all RUs with L1T numbers of the events together with the BU ID and all IDs of RU contributing with super-fragment
4) EVM and RUs build the super-fragments of the requested events and send them to the BU
5) After receiving all super-fragments for an event the BU builds the event and writes it to the RAM disk

A) BUs get the number of events for a given lumisection from EVM

RUBU machine

Alternative File-Based Event Builder

In the LHC Phase-2, **filter farm** processes need to **assemble events** from the given orbit for analysis. Why not **build** the **event** or **orbit** directly within the **filter farm**? By doing so, the step of moving data between super-fragments and the event or orbit building process can be eliminated.

Benefits of building events or orbits in the filter farm process:

1. **No** need for the **Event Backbone network** resulting in **cost savings** on network line cards and NICs;
2. **Simplifying** the event builder **code** reduces its task to only building super-fragments;
3. **Building events** or **orbits** entirely in **RAM** disk on filter farm PCs **reduces** the overall number of **memory copies** in the data flow, which would not be the case if the BU applications were moved to the filter farm PCs.

File-based Event Builder

Super-fragment Builder (SFB) — The SFB constructs multiple super-fragments based on the grouping of FEDs and stores them in local RAM disks.

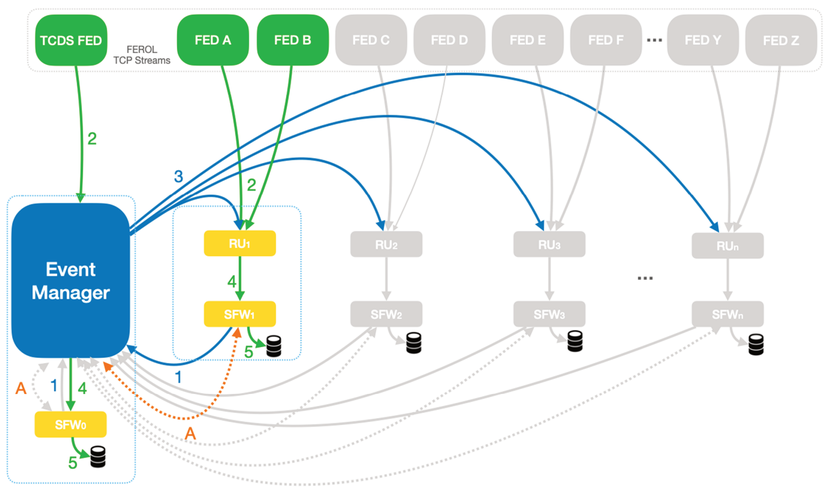
Builder Filter Farm File Based (B3F) — The BF3 accesses super-fragments from all RU machines via the NFS over Ethernet and builds complete events within the HLT farm.

Super-fragment Builder

A **super-fragment** consists of the data read by **one** or **more Front-Ends** and corresponding to the same **L1 accept** or **orbit**, and the SFB **constructs** multiple **super-fragments** corresponding to the number of Read-Unit (RU) machines in the DAQ system, storing them in local RAM disks. In the LHC Phase-2, the DAQ **custom-designed** hardware will **identify** potential **issues** during data taking. In contrast, in the LHC Run 3, this task is handled by the EVB, which needs to be integrated with the SFB.

Preliminary **performance** tests of the **SFB** using DAQ 3 hardware were conducted at the beginning of the year with the **DAQ 3 Emulator**.
By **dropping** data in the **RAM disk** (without HLT), the SFB was able to reach the nominal **115 kHz** L1 rate for the LHC Run 3.

Super-fragment Builder Protocol

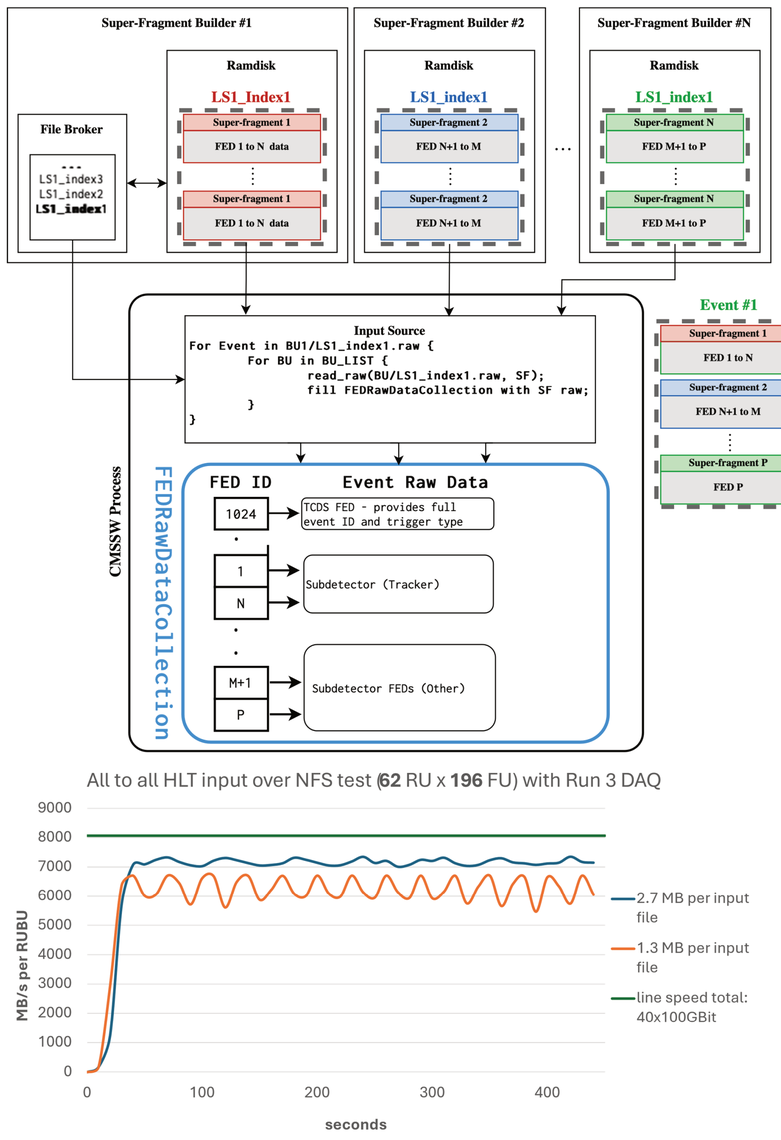


1) SFW requests to EVM N resource to build
2) Asynchronously Event fragments from FEROL to EVM & RUs
3) EVM schedule & Control:
   A. EVM receives enough event fragment from TCDS FED to satisfy one request from N SFWs
   B. EVM sends a message to all RUs with L1T numbers of the events together with the SFW ID and all IDs of RU contributing with super-fragment
4) EVM and RUs build the super-fragments of the requested events and send them to their SFW
5) The SFW after receiving a super-fragments writes it to the RAM disk

A) SFWs get the number of events for a given lumisection from EVM

RU machine

Builder Filter Farm File Based (B3F)



- **Super-Fragment Builder:**
  - In this node, the SFB runs and creates the stream files in the RAM disk.
- **File Broker:**
  - Manage the assignment of Super-Fragment stream files to Filter Farm processes
- **CMSSW DAQ** modules:
  - **INPUT**:
    - Read super-fragments and meta data from all RU RAM disks
    - Build events
    - Write booking metadata for HLTD
  - **OUTPUT**:
    - Write output streams and meta data to local RAM disk
- **High Level Trigger Daemon** (HLTD):
  - Merge output streams to Storage and Transfer System
  - Monitoring, scheduling CMSSW processes

All to all HLT input over NFS test (**62** RU x **196** FU) with Run 3 DAQ

This plot shows the B3F performance using DAQ 3 RUBU nodes of **all-to-all** HLT NFS using TCP/IP operation with a script running in each node to **generate RAM disk files** using real event sample:

- Total throughput Limited by top of the rack switch uplinks 40 x 100 GbE (8 links per rack);
- Better result with larger files (nearly 90% of the line speed);
- Next step to explore NFS over RDMA.

Future Work

In the **LHC Run 3**, the main goal is to develop and commission the File-based Event Builder during periods when data is not being taken. The target for the **File-based Event Builder** is to go **operational** for data taking during **the final years** of the LHC Run 3.

Looking ahead to the **LHC Phase-2**, depending on the **success** of the **File-based Event Builder** during the LHC Run 3, the **DAQ** architecture will need to be **updated** and take advantage of **new technologies** as part of a **redesigned** system.

October 19 - 25, 2024 CHEP 2024

"**Track 2 - Online and real-time computing**" as Poster 14 in session "Poster" on Tuesday, the 22nd of October 2024

CONTACTS

Name: Andrea Petrucci
Organization: UCSD
Email: andrea.petrucci@cern.ch
Phone: +41227662563

Name: Srećko Morović
Organization: UCSD
Email: srecko.morovic@cern.ch
Phone: +41754116276