EPJ Web of Conferences **295**, 02011 (2024) https://doi.org/10.1051/epjconf/202429502011
CHEP 2023

The CMS Orbit Builder for the HL-LHC at CERN

Vassileios Amoiridis[1], *Ulf* Behrens[2], *Andrea* Bocci[1], *James* Branson[3], *Philipp* Brummer[1], *Eric* Cano[1], *Sergio* Cittolin[3], *Joao* Da Silva Almeida Da Quintanilha[1], *Georgiana-Lavinia* Darlea[4], *Christian* Deldicque[1], *Marc* Dobson[1], *Antonin* Dvorak[1], *Dominique* Gigi[1], *Frank* Glege[1], *Guillelmo* Gomez-Ceballos[4], *Patrycja* Gorniak[1], *Neven* Gutić[1], *Jeroen* Hegeman[1], *Guillermo* Izquierdo Moreno[1], *Thomas Owen* James[3], *Wassef* Karimeh[1], *Miltiadis* Kartalas[1], *Rafał Dominik* Krawczyk[2*], *Wei* Li[2], *Kenneth* Long[4], *Frans* Meijers[1], *Emilio* Meschi[1], *Srećko* Morović[3], *Luciano* Orsini[1], *Christoph* Paus[4], *Andrea* Petrucci[3**], *Marco* Pieri[3], *Dinyar Sebastian* Rabady[1], *Attila* Racz[1], *Theodoros* Rizopoulos[1], *Hannes* Sakulin[1], *Christoph* Schwick[1], *Dainius* Šimelevičius[1,5], *Polyneikis* Tzanis[1], *Cristina* Vazquez Velez[1], *Petr* Žejdl[1], *Yousen* Zhang[2], and *Dominika* Zogatova[1]

[1]CERN, Geneva, Switzerland
[2]Rice University, Houston, Texas, USA
[3]UCSD, San Diego, California, USA
[4]MIT, Cambridge, Massachusetts, USA
[5]Vilnius University, Vilnius, Lithuania

Abstract. The Compact Muon Solenoid (CMS) experiment at CERN incorporates one of the highest throughput data acquisition systems in the world and is expected to increase its throughput by more than a factor of ten for High-Luminosity phase of Large Hadron Collider (HL-LHC). To achieve this goal, the system will be upgraded in most of its components. Among them, the event builder software, in charge of assembling all the data read out from the different sub-detectors, is planned to be modified from a single event builder to an orbit builder that assembles multiple events at the same time. The throughput of the event builder will be increased from the current 1.6 Tb/s to 51 Tb/s for the HL-LHC orbit builder. This paper presents preliminary network transfer studies in preparation for the upgrade. The key conceptual characteristics are discussed, concerning differences between the CMS event builder in Run 3 and the CMS Orbit Builder for the HL-LHC. For the feasibility studies, a pipestream benchmark, mimicking event-builder-like traffic has been developed. Preliminary performance tests and results are discussed.

1 Introduction

The Compact Muon Solenoid (CMS) is one of four main Large Hadron Collider (LHC) experiments at CERN [1]. CMS is currently taking data in Run 3, started in 2022 and expected to continue until 2025 at a centre of mass energy of 13.6 TeV and a luminosity of $2 \times 10^{34}\text{cm}^{-2}\text{s}^{-1}$. After the end of Run 3, CMS is expected to undergo the Phase-2 upgrades in view of the High Luminosity phase of the LHC (HL-LHC) at a centre of mass energy

*Corresponding author e-mail: rafal.dominik.krawczyk@cern.ch
**Corresponding author e-mail: Andrea.Petrucci@cern.ch

of 14 TeV and a luminosity of $7.5 \times 10^{34}\text{cm}^{-2}\text{s}^{-1}$. Together with the substantial changes in the sub-detectors, the data acquisition system (DAQ) will also be upgraded to handle the increased rates [2]. The event size, the L1 trigger (L1) accept rate, and the high level trigger (HLT) accept rate will increase from the current 2 MB, 100 kHz and 2 kHz in Run 3 to 8.4 MB, 750 kHz, and 7.5 kHz, respectively, at the HL-LHC.

This work covers preliminary studies of event building, which is one of the components in the DAQ system. Its purpose is assembling events, which are collections of data representing a full, detector-wide snapshot of CMS response to the collision of two individual proton bunches. Event data fragments, produced by all sub-detectors, are conveyed to the event builder for events accepted by the L1 trigger. The process of event building then involves gathering fragments into complete events for further selection in the HLT. To adapt to the new operational parameters, a different form of data aggregation is planned for the increased data rate. Instead of building a single event, an alternative orbit structure is proposed that can incorporate multiple events at a time and that represents a collection of events in a complete revolution of the LHC beams. The main scope of the presented feasibility study was a preparation of the software to evaluate the impact of a new framework and the containerization in Kubernetes (K8s) on the all-to-all network transfers.

The remainder of the paper is structured as follows. Section 2 covers the details of DAQ for the HL-LHC. In section 3 the event builder and its upgrade to per-orbit aggregation are discussed. Section 4 demonstrates the software study of the Orbit Builder for the HL-LHC, with performance tests and the results. Section 5 covers a conclusion of the studies and planned future work.

2 The CMS DAQ architecture in Run 3 and for the HL-LHC

The CMS DAQ architecture in Run 3 is presented in Figure 1 [3], with the following steps:

1. approximately 760 Detector Front-End Drivers (FEDs) collect the data from the sub-detector front-end electronics;
2. each of FEDs is point-to-point connected either to one of 510 Front-End Readout Optical Link boards (FEROL) or to one of 32 FEROL-40 boards;
3. FEROL and FEROL-40 boards gather FED data and output them in TCP/IP streams using 10 Gb Ethernet links;
4. using the FED Builder Data Concentration Network, the FEROL, and FEROL-40 TCP/IP streams are fed to the event builder;
5. 62 RU/BU server nodes assemble events using the 100 Gb/s Ethernet RDMA over Converged Ethernet (RoCE) Core Event Builder Network. Assembled events are stored in RAM disks of the RU/BU nodes;
6. complete events are sent to 200 HLT nodes for selection;
7. selected events are stored back in the RAM disks of the RU/BU nodes;
8. the HLT output is temporarily stored in Lustre and then sent to persistent storage outside of the CMS online infrastructure for offline analysis.

The proposed CMS DAQ architecture for HL-LHC [2] is presented in Figure 2. The design follows the same principle of operation. In comparison with the system in Run 3 in Figure 1, several changes are envisaged. There will be a general increase in link speeds, data rate, and computing power. For the event builder, the estimated number of RU/BU nodes is 200. 400 Gb/s Ethernet is planned to be used for the RoCE Event Builder Network. Depending on the needed computing power, the HLT farm is planned to consist of between 800 to 1600 servers. New custom acquisition boards, the DAQ and TCDS Hub (DTH) will

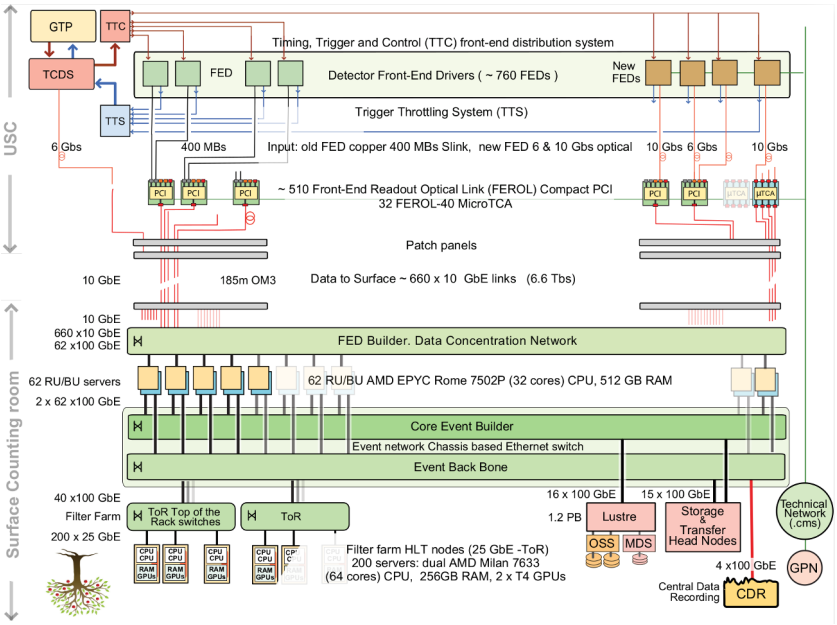

Figure 1. CMS DAQ architecture in Run 3 [3].

replace the FEROL and FEROL40. They will also incorporate the functionality of the Trigger Throttling System, and of the Timing, and Trigger and Control system. The DTH boards will also handle Trigger and Clock Distribution System (TCDS) functions. The DTH boards are estimated to use approximately $1000 \times 100\ Gb/s$ output links in total. The DTH boards will output orbit structures (versus outputting event structures in FEROL and FEROL-40 boards).

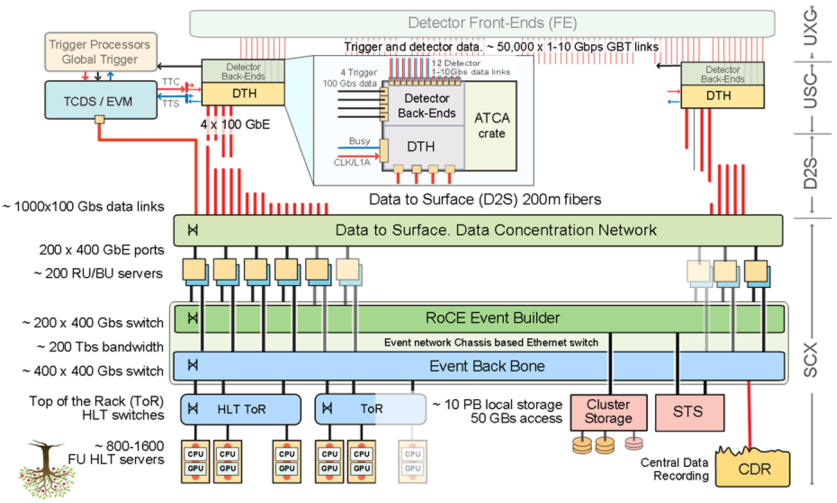

Figure 2. CMS DAQ architecture for the HL-LHC [2].

3 The CMS event builder upgrade for the HL-LHC

The scope of the presented work was the preparation of the transition from events to orbits in the event builder.

3.1 The event building protocol

The principle of operation of the Run 3 event builder and the HL-LHC orbit builder remains the same. The purpose is to assemble events for the HLT farm from the fragments produced by all sub-detectors. In Run 3 event builder there are three types of units:

1. *Readout Unit* (RU) - It receives event fragments from FEROL or FEROL-40 boards via TCP/IP streams and assembles them into superfragments. RUs buffer superfragments and, when requested by the EVM, send them to BUs.
2. *Builder Unit* (BU) - It receives superfragments from all of the participating RUs, assembles them into complete events, and stores them in a RAM disk. Events are thus made available for further filtering in the HLT nodes. In Run 3 a folded configuration is used where RUs and BUs share the same node.
3. *The Event Manager* (EVM) - Based on BUs requests and the TCDS trigger data, it assigns events to BUs and broadcasts requests to all RUs to send their data.

Figure 3 presents the CMS event-building protocol in Run 3. Control messages are marked with blue dashed arrows. Messages with events are marked with black arrows. Builder Units also receive the TCDS data (not shown in the picture for clarity). The following five steps can be distinguished:

1. BUs send event requests to the EVM;
2. the EVM maps the BU requests with the TCDS data;
3. after the EVM scheduling is complete, the EVM broadcasts requests to RUs;
4. based on the requests, RUs send super fragments to BUs;
5. BUs assemble complete events and store them in the RAM disk for filtering in the HLT.

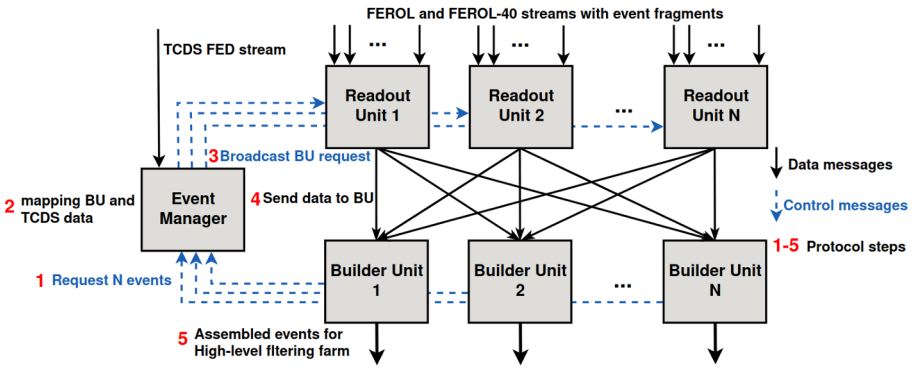

Figure 3. The protocol of the CMS event building in Run 3.

3.2 CMS orbit aggregation for the HL-LHC

The main quantifiable CMS event builder differences between Run 3 and the HL-LHC are the increase of total throughput from 1.6 Tb/s to 51 Tb/s, entailing an increase of event building

nodes from approximately 60 to approximately 200. The main functional difference is a set of new structures received by RUs and BUs. In Run 3, FEROL and FEROL-40 boards are sending event fragments to RUs. At the HL-LHC, the DTH will output orbit fragments instead. The orbit structure represents a collection of events from a complete revolution of the LHC beams. Orbit data are fed to the event builder from the DTH using the information from TCDS. The purpose of aggregating event fragments into an orbit structure is to increase the amount of data per transmission, to maximally exploit the available link speeds, and to reduce the number of control messages in the event-building. While the expected HL-LHC L1 accept rate is up to 750 kHz, the orbits will be fed at a rate of 11.2 kHz.

Figure 4 illustrates how the introduction of an orbit fragment impacts the building process. In Run 3, RUs receive event fragments from FEROL and FEROL-40 boards and construct event super fragments, and BUs build complete events. At the HL-LHC, DTH will construct and feed orbit fragments to RUs. RUs will then construct orbit super fragments and send them to BUs that will assemble orbits.

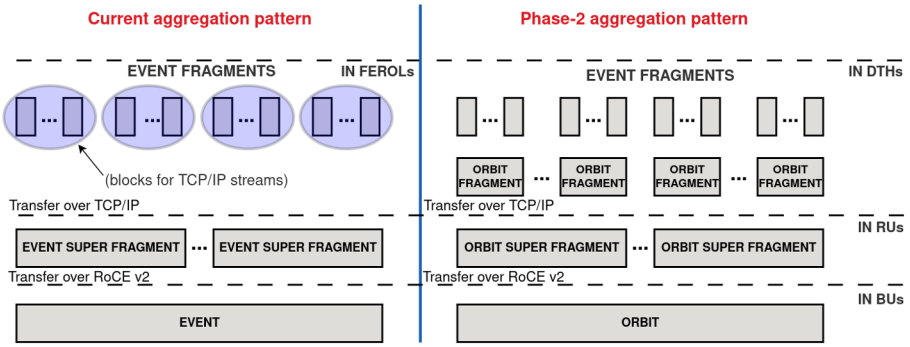

Figure 4. The event construction in CMS in Run 3 versus the orbit construction in CMS for the HL-LHC.

The aggregation differences in RUs can be seen in Figure 5. In Run 3, each FEROL and FEROL-40 receives data from the following possible link configurations: 2×400 MB/s, or 2×6 Gb/s, or 1×10 Gb/s, or $1\times$ 400 MB/s and $1\times$ 6 Gb/s for FEROL, and 4×10 Gb/s for FEROL-40. Then the boards sends TCP/IP streams to the RU. RUs obtain event fragments from the streams and construct event super fragments. The event super fragment is a structure containing event fragments from all FEROL and FEROL40 boards connected to a given RU. At the HL-LHC, the DTH will receive data from up to 24×25 Gb/s links and will output TCP/IP streams using up to 5×100 Gb/s links. The RU only analyzes the orbit header to produce orbit super fragments. There is no need to analyze each of the event fragments individually, because all consistency checks for the event fragment are performed in the DTH. The orbit fragment size is estimated to be in the range of 50-250 kB. Since the front-end links to the DTH will be arranged to produce balanced data sizes, an average orbit fragment size of 125 kB can be expected [2]. It is assumed that data transfers close to the link speed of 400 Gb/s will be reached. One 400 Gb/s link per RU is planned, and with a maximum of 24 TCP/IP DTH streams, the estimated average orbit super fragment size will be $24\times125kB \approx 3MB$. In a conservative estimate, including the overhead of control messages, a rate of 16 kHz can be potentially reached. This is above the required 11.2 kHz HL-LHC orbit rate.

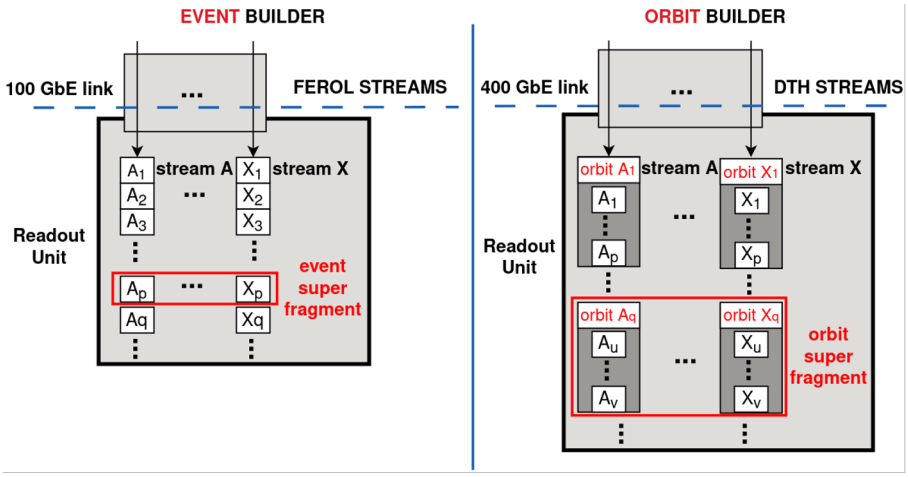

Figure 5. Event versus orbit aggregation in Readout Unit.

4 Feasibility studies for the evolution of the CMS event builder software

New software has been developed to evaluate the network performance of event-builder-like, all-to-all traffic.

4.1 The pipestream benchmark

A benchmarking software, referred to as pipestream, has been developed and used in the network performance tests. It is based on the XDAQ 2nd generation, a framework developed for CMS [4], that supports RoCE. Pipestream relies on a protocol of scheduled data sending from clients to servers over the network. Two setups can be used, with clients and servers running on separate nodes or sharing nodes. In the former scenario, the number of clients and servers can be different. The latter pattern is referred to as the folded configuration and mimics the Run 3 event builder with an instance of RU and an instance of BU sharing the same node. Two sender scheduling policies have been implemented: a best-effort scheduling driven by completion of transmissions, and a policy optimized for fairness, waiting for all nodes to complete a minimal number of transmissions.

The pipestream source code has been incorporated into the XDAQ code [5]. Pipestream relies on a Representational State Transfer (REST) paradigm for control and monitoring. A Finite-State Machine (FSM) is used for runtime control and YAML is used for configuration. The FSM states are used to control connection and disconnection of nodes, as well as to start and stop the data transmission. The benchmark can be launched as an independent application (referred to as the standalone configuration), or it can be executed in K8s. The message rates of clients and servers are periodically probed through REST. Several parameters can be adjusted and have been tuned for the tests: maximal size of transmitted message, buffer size that node allocates when establishing connection to other node, total network burst size, number and affinity of threads for servicing RoCE transmissions, and affinity of the RoCE transmission buffers.

Python and Bash scripts have been implemented for launching pipestream in the standalone configuration or in K8s. Using REST commands, the scripts make pipestream instances transition through FSM states of clients and servers. The scripts iterate over a given

set of message sizes to probe the message rates through REST. A Python script has also been developed to plot the resulting output using ROOT.

4.2 Tests and result

The pipestream benchmark was used for the tests with an existing Run 3 validation system with 100 Gb/s Ethernet and 14 test nodes. These servers were dual-socket machines with two Intel(R) Xeon(R) Gold 6130 CPUs @ 2.10GHz, 256 GiB DDR4 @ 2666 MT/s and with Mellanox ConnectX-6 NICs running in the RoCE mode. The nodes were connected with a single Juniper QFX10000-30C line card in a Juniper QFX10008 chassis. One node was used as an orchestrator where the scripts were launched. Two preliminary tests were made: the assessment of the impact of containerization on the pipestream performance, and the evaluation of the impact of the number of pipestream communication threads on the all-to-all network traffic. Affinity, burst size, and buffer size were optimized to achieve the best performance in either of the tests.

The first test was made for a configuration where each of the sending clients and receiving servers ran on separate nodes. The results are presented in Figure 6. There was one thread servicing transmissions per connection between a client and a server. The average client throughput was measured for different message sizes and for different numbers of clients and servers. As seen in Figure 6, the results are almost the same for the runs in the standalone configuration (left plot) and in K8s (right plot). This proves that the containerization can be considered for the development of the event building for HL-LHC.

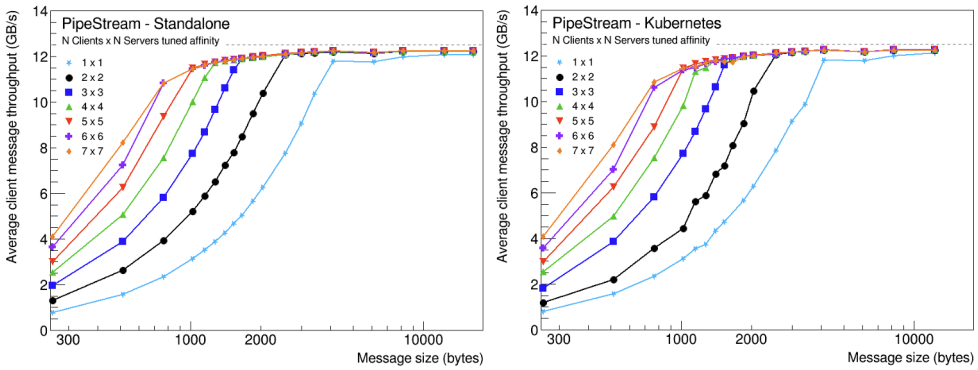

Figure 6. Throughput versus message size in pipestream runs with one thread per connection for different numbers of nodes in the standalone configuration (left) and in K8s (right).

The second test was conducted for an 8×8 folded configuration, with both the sending client and the receiving server running on each of the 8 nodes. The average client throughput was measured for different message sizes and for different numbers of threads servicing transmissions. The results are presented in Figure 7, where the throughput versus the message size for a sending client is shown. The dashed line is the 100 Gb/s link speed. At a message size of 4 KB, with one thread used for transmissions, sending clients reached an average throughput that was close to 100 Gb/s.

When operating below the 4KB message threshold, the throughput was below the link speed and depended on the number of threads servicing transmission. This was likely due to an increase in message rate that the system needed to sustain, resulting in an additional overhead to handle the control messages. This feature is well known and documented for the CMS use case [6, 7].

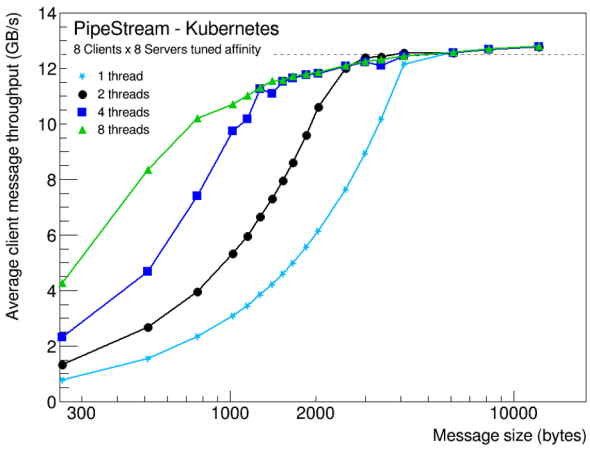

Figure 7. Throughput versus message size for a sending client in a K8s test.

5 Conclusions and future work

The upgrade of the CMS data acquisition system for the High-Luminosity phase of LHC involves an increase in the event size, the L1 trigger accept rate, and the high level trigger accept rate. We investigated the performance of the CMS event-building-like, all-to-all network traffic. Using the DAQ Online C++ framework, a proof of concept was made using the pipestream benchmark. The software performance was the same when containerized in Kubernetes. The decrease in throughput, observed below 4 KB message size, indicates that using larger messages may be advantageous. Higher throughput was reached with more pipestream threads. For the future evaluation, scaling the network up for increased traffic and 400 Gb/s links have to be tested for event-building-like data transmissions. Additionally, since pipestream only simulated the network traffic, the performance of the building process has to be also evaluated.

References

[1] S. Chatrchyan, et al. The CMS experiment at the CERN LHC, JINST 3 S08004 (2008)
[2] The Phase-2 Upgrade of the CMS data acquisition and High Level Trigger TDR (2022)
[3] The CMS Collaboration, Development of the CMS detector for the CERN LHC Run 3, CERN-EP-2023-136, submitted to JINST (2023)
[4] D.Simelevicius, L.Orsini, et al., Towards a container-based architecture for CMS data acquisition, in these proceedings
[5] R. Krawczyk, A.Petrucci, et al., Pipestream, https://gitlab.cern.ch/cmsos/core/-/tree/feature_272_pipestream_imp/benchmark
[6] G. Bauer et al., A comprehensive zero-copy architecture for high performance distributed data acquisition over advanced network technologies for the CMS experiment, IEEE Trans. Nucl. Sci. vol. 60 no.6 (2013)
[7] T.Bawej, et al. Achieving High Performance With TCP Over 40 GbE on NUMA Architectures for CMS data acquisition, IEEE Trans. Nucl. Sci., vol. 62, no. 3 (2015)