EPJ Web of Conferences **295**, 02029 (2024)
CHEP 2023
https://doi.org/10.1051/epjconf/202429502029

The new ALICE data acquisition system (O^2/FLP) for LHC Run 3

Vasco Barroso[1] [*], *Konstantinos* Alexopoulos[1], *Martin* Boulais[1], *Sylvain* Chapeland[1], *Filippo* Costa[1], *Roberto* Divià[1], *Ulrich* Fuchs[1], *Claire* Guyot[1], *Jik* Huijberts[2], *Piotr* Konopka[1], *Iosif-Charles* Legrand[3], *Teo* Mrnjavac[1], *George* Raduta[1], *Adam* Wegrzynek[1], *Pierre* Vande Vyvre[1], and *Barthélémy* von Haller[1]

[1]CERN, 1211 Geneva 23, Switzerland
[2]Amsterdam University of Applied Sciences, 1101 CD Amsterdam, Netherlands
[3]Horia Hulubei National Institute of Physics and Nuclear Engineering, Măgurele 077125, Romania

Abstract. ALICE (A Large Ion Collider Experiment) has undertaken a major upgrade during the LHC Long Shutdown 2. The increase in the detector data rates led to a hundredfold increase in the input raw data, up to 3.5 TB/s. To cope with it, a new common Online and Offline computing system, called O^2, has been developed and put in production. The O^2/FLP (First Level Processor) system, successor of the ALICE DAQ system, implements the critical functions of detector readout, data quality control and operational services running in the CR1 data centre at the experimental site. Data from the 15 ALICE subdetectors are read out via 8000 optical links by 500 custom PCIe cards hosted in 200 nodes. It addresses novel challenges such as the continuous readout of the TPC detector while keeping compatibility with legacy detector front-end electronics. This paper discusses the final architecture and design of the O^2/FLP system and provides an overview of all its components, both hardware and software. It presents the selection process for the FLP nodes, the different commissioning steps and the main accomplishments so far. It will conclude with the challenges that lie ahead and how they will be addressed.

1 Introduction

1.1 The ALICE experiment

A Large Ion Collider Experiment (ALICE) [1] is the detector designed to study the physics of strongly interacting matter at extreme energy densities at the CERN Large Hadron Collider (LHC). After two successful data taking periods (LHC Run 1 [2010-2013] and Run 2 [2015-2018]), ALICE underwent a major upgrade [2] of the detector apparatus and the online systems during Long Shutdown 2 (LS2, from 2019 to 2021), which significantly improved the experiment capability to probe the quark gluon plasma (QGP). Regular data taking

---

[*] Corresponding author: vmcb@cern.ch

operations resumed in 2022 and will continue during LHC Run 3 [2022-2025] and Run 4 [2029-2032].

1.2 The ALICE Online-Offline computing system

As part of the ALICE LS2 upgrade, a brand new computing system called Online-Offline (O^2) [3] has been designed, built and installed. It is based on a completely new model, which aims at maximum compression of the data volume by reconstructing tracks online to minimize the costs for data processing and storage. To achieve this, two new computing farms have been built: the First Level Processor (FLP) farm located in Counting Room 1 (CR1, in the experiment access shaft) and the Event Processing Node (EPN) farm located in Counting Room 0 (CR0, on the surface). The FLP farm performs the traditional Data Acquisition (DAQ) functions and reduces the aggregated detector data rate from 3.5 TB/s to 900 GB/s (mainly by zero suppression). Data is then shipped to the EPN farm via an Infiniband network where a first reconstruction pass is performed, further reducing the data rate to 130 GB/s which is then written to permanent storage. The vast majority of the O^2 software has been written from scratch to meet the challenges of this new data model and to introduce modern computing technologies.

This paper will mainly focus on the hardware and software components under the responsibility of the ALICE O^2/FLP project, which inherits its name from the FLP farm where most of its components reside.

2 The O^2/FLP computing system

The O^2/FLP computing system consists of a set of hardware and software components, which provide three main functions: detector readout, data quality control and operational services. This section describes them in detail.

2.1 The O^2/FLP hardware

2.1.1 Data center infrastructure

CR1, located in the experimental site, hosts the O^2/FLP computing farm. It is equipped with 42 racks, out of which 24 are used for the present system consisting of 260 servers. Each rack is equipped with a rear cooling door connected to 14°C facility water that can absorb 12-14kW. The whole cooling circuit is limited to 280kW, so careful planning was put in the layout of the computing room, concerning the total number of machines, their placement in the racks and the respective arrival of readout fibers.

The power for the computing room is provided by a 500kVA uninterruptible power supply (UPS), delivering a maximum of 14kW to each rack on two power distribution units. For redundancy purposes all racks can be switched over to a second, non-UPS-protected power network in case of a UPS failure. Depending on the usage, the power consumption of the whole system varies between 75kW and 82kW, provided by a UPS backed-up power network.

The readout part is realized by roughly 9000 fibers coming from the detector’s front end electronics (FEE) and entering the computing room in more than 50 fiber trunks of 144 fibers each. The readout cards in the servers are connected to the trunks through patch panels, at least one per rack. There is a mixture of fiber trunks arriving on each patch panel, providing readout links on MTP12 connectors and access to the trigger PON network through single-mode connections that need further fan-out to the Optical Network Units (ONU) on all readout cards.

All servers are connected to the O^2/FLP management network, providing 25GbE connectivity to all readout and service nodes, as well as 100GbE for hypervisors and servers receiving huge amounts of traffic.

2.1.2 Readout cards

The new ALICE detector readout chain includes two PCIe readout cards: the new Common Readout Unit (CRU) [4] and the legacy Common Readout Receiver Card (C-RORC) [5], already used during LHC Run 2.

The CRU is a dual (x8) gen3 PCIe card equipped with an Intel Arria 10 FPGA which is connected to up to 4x12 channel bi-directional 10.3125 Gb/s optical transceivers to ensure the connections to the front-end electronics. It communicates with the trigger system using 2 Small Form Pluggable (SFP+) connectors and implements two different readout modes: continuous and triggered. The firmware, written in VHDL, is divided into two parts: first the common firmware which provides the interfaces to PCIe, trigger and timing, and FEE configuration; second the user logic used by detector systems requiring detector-specific data processing (such as baseline correction or zero suppression). There is a total of 472 CRU cards used in the production system.

The C-RORC is a gen2 PCIe card equipped with a Xilinx Virtex-6, interfacing to 12 serial full duplex optical links via three QSFP modules, with each link supporting a maximum rate of 6.6 Gbps. It is used by three subdetectors that could not upgrade their FEE during LS2 and there is a total of 16 C-RORCs in the production system.

Both cards are responsible for moving the input detector data into the memory of the host server via Direct Memory Access (DMA) transfers.

2.1.3 First Level Processor nodes

The First Level Processor (FLP) nodes hosts the readout cards (up to 3 CRUs or 4 C-RORCs) and runs a multi-process topology that reads the detector data, performs an initial processing and/or quality control and ships the data to the EPN farm. The selected hardware model (via a competitive tender [6]) was the Dell PowerEdge R740 server with dual-Xeon CPUs and all memory banks equipped to maximize the memory bandwidth and therefore the DMA capabilities of the nodes. In addition, all nodes are equipped with a 100G HDR Infiniband network interface for data transport to the EPN processing farm.

As there is very little computing done on most readout nodes, the Xeon 4210 CPU was chosen for most subdetectors except two that run more CPU-hungry processes, where gold-shelf CPUs of the type 6230 were necessary. With all memory lanes equipped with 8 GB modules, all nodes have 96GB of RAM.

There is a total of 203 FLP nodes used in the production system.

2.1.4 Quality Control nodes

The Quality Control (QC) nodes are used to run QC-related topologies. Equally based on the Dell PowerEdge R740 server, they are equipped with lower-gold-shelf CPUs of the 5218/5220 types, 64 GB to 128 GB of RAM and 25 GbE to 100 GbE network cards. There is a total of 17 QC nodes used in the production system.

2.1.5 Infrastructure nodes

On top of FLP and QC nodes, the O^2/FLP computing system includes a multitude of virtualized infrastructure nodes hosted on three powerful VMware hypervisors – with dual Xeon Gold 6248 CPUs and 384 GB of memory each - that are used to run operational services. Having been designed to run 25 Virtual Machines (VM), they currently host 45 VMs with great efficiency and reliability.

In terms of storage, a live-migration capable iSCSI backend provides storage space to the virtualization cluster via two iSCSI arrays in a Live-Volume setup, each providing 50TB of full-flash storage through 4 redundant links of 100GbE each. Originally meant as storage back-end for the hypervisors, backups and databases, this storage setup providing 100TB of full-flash space with 800Gbps access bandwidth quickly found more use cases and today serves 27 individual systems.

2.2 The O^2/FLP software

The software portfolio of the O^2/FLP computing system is diverse and includes a mix of in-house components and 3rd party tools to fulfil the required functional tasks. This section provides a survey of that portfolio.

2.2.1 Detector readout

Simply known as *readout,* the detector readout software [7], written in modern C++, is in charge of controlling the readout cards (CRU and C-RORC) and managing the memory buffers into which the DMA transfers are performed. Interaction with the readout cards is done via the ReadoutCard library [8], a C++ high level interface based on the Portable Driver Architecture. Once the data is in the host memory, *readout* aggregates the data from multiple links, performs a set of checks, formats the outgoing data and slices it up into time-based data units that are then sent to the data distribution software.

To satisfy the different subdetector teams needs during early commissioning, the detector readout software comes with a rich set of operational functionalities such as saving output to local file, LZ4 compression of the data and replay of data from file.

On the production system, each FLP node contains one single running instance of *readout.*

2.2.2 Experiment Control System

The experiment control system of the ALICE experiment, *AliECS* [9], is responsible for orchestrating the data taking and online processing operations. It directly manages the lifecycle and configuration of all the tasks running on FLPs and QC nodes and interfaces with the dedicated control system of the EPN farm to deploy the reconstruction and calibration topologies that run there. It also interfaces with the other ALICE online systems for an automated and centralized operational control.

AliECS aggregates all tasks, their configuration and required hardware components into a logical entity called environment. One environment can then be used to perform (usually) one or more runs of different types and is represented by a state machine which represents the aggregated state of its tasks.

Written mainly in Go [10] to benefit from its advanced synchronization and threading capabilities, *AliECS* uses Apache Mesos [11] as Resource Management System and gRPC [12] for inter-process communication. A C++ library is provided to interface with the tasks that are controlled directly by *AliECS*.

The shift crews that operate the ALICE experiment use the *AliECS GUI* [13], a web-based user interface that allows them to set up and deploy complex data taking and online processing workflows in an easy and intuitive way. Once the run is started, users can also check and monitor the state of the environment in detailed views.

2.2.3 Data Quality Control

The data quality control software (QC) [14], written in C++, provides a single and comprehensive solution for both online data quality monitoring and offline quality assurance. It includes features to sample physics data, execute user-defined algorithms, merge the resulting histograms, evaluate the results and store the QC objects in a repository. Visualization is provided by a dedicated web-based user interface which allows for the definition of operational layouts combining many objects in a single view. The JSROOT library is used to draw and interact with the objects.

During online operations, multi-process QC topologies are launched on FLPs, EPNs and QC nodes to perform all these functions on different types of physics data, using message passing channels to transfer the data between the different tasks. During offline operations, the output of all related jobs is merged by a multi-layer topology, with intermediate results stored in local files.

At the time of writing, ALICE has 180 QC tasks that produce 32 000 types of objects for inspection by both shifters and experts.

2.2.4 Operational services - Monitoring

The monitoring subsystem [15] provides an overview of the entire data taking system by collecting, processing, storing and visualising metrics and alerting users when abnormal situation occurs.

The metrics are fed to the system from over 100k sources via a C++ interface with an interval of between 1 and 15 seconds. They are aggregated on each node by a Telegraf agent which also scrapes counters from the operating system, services and hardware components. All metrics are stored in an InfluxDB time-series database. InfluxDB supports data retention and down sampling, which decreases the value resolution over time and brings down the total database size. Certain metrics are published to a Kafka streaming platform. Metrics in Kafka are pre-processed, before being stored in InfluxDB to speed up the query time and reduce the load on the database engine. Kafka also allows other subsystems to subscribe for desired metric streams. The metric visualisation is provided by Grafana. It serves real-time dashboards, by subscribing to a Kafka stream, and historical dashboards by simply querying the database. The dashboards provide high level status for shift crews in the ALICE control room and drill-down view for subsystem experts. Grafana also defines the alerting rules, such as node health or incorrect readout throughput for a given detector. The alerts are routed to a Mattermost channel or an email group.

The production monitoring system consists of:

- 5 InfluxDB nodes: 3 for “hot” metrics (up to 7 days) and 2 for “cold”, historical storage (up to 6 months, down sampled), in total over 700 GB of metrics; all instances serve, in average, 9k queries per hour,
- Kafka cluster operating in High Availability mode with 6 data processors,

- 1 Grafana instance providing 27 operational dashboards.

2.2.5 Operational services – Configuration

The configuration of the tasks relies on a C++ library that allows them to retrieve their configuration from different backends. In production, a cluster of 3 Consul instances provide a key/value store that serves as main configuration repository. FLPs and QC nodes run Consul agents and all configuration changes are committed in a git repository for historical record. Consul is also used for service discovery of QC tasks and for health checking of long running services.

2.2.6 Operational services – Logging

The *infoLogger* [16] is the package used in the online system to generate, transport, collect, store, and consult software log messages. It was used already during LHC Run 1 and 2 and has been adapted to fulfil the requirements of the O^2 computing system.

It provides a library to inject logs, a central repository to store the messages, and user interfaces to display and query them. Any distributed process can create a message and set the associated metadata: some fields are defined automatically (timestamp, hostname, pid, etc) and some can be customized (detector, system, facility, etc). A daemon collects all the logs of the local node, and sends them to a central server, where they are stored in a database. The GUI allows reading and filtering messages, either received in real time by the server or queried from the database archive. The tool is written in C/C++, and bindings are generated for various languages (Python, Go, TCL, nodejs). The database is implemented with a single-host MariaDB server having InnoDB partitioning (one partition per day), and can keep 1 year of logs. It handles 7M messages per day in average, and the system can sustain occasional peaks of over 200M messages per day. In 2023, 2 billion logs have been collected, corresponding to 1.5 TB on disk including indexes.

2.2.7 Operational services – Bookkeeping

Bookkeeping [17] is the electronic logbook solution for ALICE during Run 3 and 4. Written in JavaScript, it uses Node.js on the backend and Mithril.js on the single page application frontend. HTTP and gRPC APIs allow read/write access and MariaDB is used as relational database.

In terms of features, *Bookkeeping* allows operators and experts to log their interventions via markdown enriched Log Entries. It stores the list of all environments and runs created, together with a list of associated configuration parameters and several counters from running tasks. Dedicated pages with pre-filled information simplifies the end of shift reporting by shift crews and plots with running efficiency statistics provide managers with a global view of the experiment's operations.

2.2.8 FLP Suite

All the O^2/FLP software portfolio and associated 3rd party tools are bundled together into a single product called the *FLP Suite,* which uses Ansible to automate the installation and configuration of all components. More than 100 Ansible roles combine with an inventory of all production hosts to provide an infrastructure-as-code solution for the deployment of new releases in the production cluster which is further simplified by a gitlab pipeline. A full deployment of a new *FLP Suite* release in production can be done in less than one hour.

3 First operational experience

The O^2/FLP computing farm, including FLPs, QC nodes and infrastructure nodes were procured, installed and commissioned between the end of 2019 and mid 2020. Following an initial period of standalone subdetector tests, 2021 marked the first year of significant data taking activity at the experimental site. In the first half of the year, a set of milestone weeks tested different parts of the system and provided useful input for the developers to improve their components. This was followed by global commissioning, where all subdetectors and online systems were successfully operated together. In October 2021, ALICE saw the first LHC collisions in the form of pilot beams, successfully collecting physics data for the first time with the upgraded detector.

The year 2022 was the first operational year of LHC Run 3. Throughout the year, different types of runs were regularly performed to both test the system and collect physics data:

- SYNTHETIC runs: based on the replay of Monte Carlo simulated data, it allowed to validate the online systems at different data rates, ranging from 1 GB/s to 1.2 TB/s;
- COSMICS runs: based on the interaction of cosmic rays with the ALICE detector, it was used for commissioning and alignment of subdetectors at data rates of 85 GB/s detector readout and 2 GB/s to permanent storage;
- PHYSICS pp 500 KHz: based on proton-proton collisions, it was used to accumulate physics data for analysis at data rates of 500 GB/s detector readout and 30 GB/s to permanent storage;

The year 2023 continued this program and at the time of writing the ALICE community was eagerly awaiting the start of the first heavy ion run of LHC Run 3. Throughout all this period, continuous improvements were applied to the O^2/FLP computing system, greatly contributing to the increase of operational efficiency from 20% in Q1 2022 to 80% by mid-2023, with peaks above 99% during some periods. Since the start of global data taking, ALICE performed more than 40 000 data taking runs which read out more than 4000 PB of data of which more that 150 PB were recorded on permanent storage.

4 Conclusion

The ALICE collaboration successfully designed and built a brand-new data acquisition system for LHC Run 3 and Run 4. All software components have been rewritten from scratch and, following a series of tests and commissioning exercises at the experimental area, are performing within the required operational parameters since the start of LHC Run 3 in 2022.

References

1. The ALICE Collaboration, *The ALICE experiment at the CERN LHC* JINST **3** S08002 (2008)
2. The ALICE Collaboration, *ALICE upgrades during the LHC Long Shutdown 2* arXiv:2302.01238 (2023)

3. P. Buncic, M. Krzewicki, P. Vande Vyvre et al, *Technical Design Report for the Upgrade of the Online-Offline Computing System* Technical Design Report ALICE **19** (2015)
4. O. Bourrion, J. Bouvier, F. Costa, E. David, J. Imrek, T.M. Nguyen, S. Mukherjee, *Versatile firmware for the Common Readout Unit (CRU) of the ALICE experiment at the LHC* JINST **16** P05019 (2021)
5. A. Borga, F. Costa, G.J. Crone, H. Engel, D. Eschweiler, D. Francis, B. Green, M. Joos, U. Kebschull, T. Kiss, *The C-RORC PCIe card and its application in the ALICE and ATLAS experiments* JINST **10** C02022 (2015)
6. F. Costa, S. Chapeland, K. Alexopoulos and U. Fuchs, *Assessment of the ALICE O2 readout servers*, EPJ Web Conf. **245**, 01013 (2020)
7. S. Chapeland, *Commissioning of the ALICE readout software for LHC Run 3*, in 26th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2023) (to be published)
8. K. Alexopoulos and F. Costa, *The ReadoutCard Userspace Driver for the New Alice O2 Computing System* IEEE Trans. Nucl. Sci. **68**, 8 (2021)
9. T. Mrnjavac, K. Alexopoulos, V. Barroso, C. Guyot, P. Konopka and G. Raduta, *The ALICE Experiment Control System in LHC Run 3,* in 26th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2023) (to be published)
10. The Go programming Language, https://golang.org (2023)
11. Apache Mesos, http://mesos.apache.org (2023)
12. gRPC A high performance, open-source universal RPC framework, https://grpc.io (2023)
13. G.C. Raduta, *From design to production: State-of-the-art web user interfaces to operate ALICE offline and online system* J. Phys. Conf. Ser. **2438**, 012041 (2023)
14. B. von Haller and P. Konopka, *The ALICE Data Quality Control,* in 26th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2023) (to be published)
15. A. Wegrzynek, G. Vino, *The evolution of the ALICE O^2 monitoring system,* EPJ Web Conf. **245**, 01042 (2020)
16. S. Chapeland et al., *The ALICE DAQ infoLogger,* J. Phys. Conf. Ser. **513**, 012005 (2014)
17. M. Boulais, G. C. Raduta and J. Huijberts, *Bookkeeping, a new logbook system for ALICE,* in 26th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2023) (to be published)