A design study for the upgraded ALICE O² computing facility

Matthias Richter for the ALICE collaboration
CERN - European Organization for Nuclear Research, Geneva, Switzerland;
Department of Physics, University of Oslo, Norway
E-mail: Matthias.Richter@cern.ch

Abstract. An upgrade of the ALICE detector is currently prepared for the Run 3 period of the Large Hadron Collider (LHC) at CERN starting in 2020. The physics topics under study by ALICE during this period will require the inspection of all collisions at a rate of 50 kHz for minimum bias Pb-Pb and 200 kHz for pp and p-Pb collisions in order to extract physics signals embedded into a large background.

The upgraded ALICE detector will produce more than 1 TByte/s of data. Both collision and data rate impose new challenges onto the detector readout and compute system. Some detectors will not use a triggered readout, which will require a continuous processing of the detector data. The challenging requirements will be met by a combined online and offline facility developed and managed by the ALICE O² project. The combined facility will accommodate the necessary substantial increase of data taking rate.

In this paper we present first results of a prototype with estimates for scalability and feasibility for a full scale system.

1. Introduction
In Run 3, the ALICE collaboration will introduce a novel approach by inspecting all collisions and discards the concept of event selection via triggering. As some of the detectors in ALICE have an intrinsic signal collection time which exceeds the average time between two collisions at a rate of 50 kHz, those detectors will be read out in a trigger-less continuous mode. This paradigm shift has a significant impact onto the design of the computing facility. Although various online systems are existing for event-based reconstruction, the application of a production system for processing of time-intervals containing O(1000) collisions is a novel case in High Energy Physics.

The prototype study for the ALICE O² project has been carried out to check the feasibility of the data processing strategy. The main components to run a small scale but yet realistic processing topology are addressed in this paper, namely (1) a flexible and modular software framework, (2) a data transportation solution, (3) some realistic data processing, and (4) process distribution and deployment on the test cluster nodes.

2. The ALICE O² project
The ALICE O² project as presented in its Technical Design Report (TDR) [1] will design, develop and operate a combined online and offline computing system for the upgraded ALICE detector.
In this section we outline the data model and data flow and the software framework design, and introduce the ALICE O^2 prototype development.

2.1. Data Model and Data Flow
Instead of triggered events, data will be read out and processed according to a Timeframe (TF) structure, which is characterized by fixed time intervals. Global reconstruction of Timeframes requires information of the full ALICE detector as input. All data belonging to the same time interval has to be aggregated on a single processing node. The individual components of the data flow are sketched in Figure 1, more details are described in [1].

- **Detector Electronics**: Detector Front-end electronics (FEE) send data according to time intervals and/or trigger over approximatively 8100 read-out links
- **First Level Processor (FLP)**: Local aggregation of data from up to 48 optical links per FLP in Timeframes (TF), the Timeframe data granularity is defined by heartbeat trigger events (HBE); Local reconstruction and calibration
- **Switching Network**: Physical transportation of timeframe data
- **Event Processing Node (EPN)**: Aggregation of all sub-timeframes and global reconstruction of the full TF data sample
- **Storage**: Intermediate/permanent storage of pre-processed and compressed data

![Figure 1. Schematic overview of the ALICE O^2 computing facility [1].](image)

2.2. Software Framework
The ALICE O^2 software framework is currently under development. It has to be capable of supporting multiple and heterogeneous detector systems. The framework builds on ALFA, the common ALICE and FAIR software project.

Data processing in ALICE O^2 is split into multiple processing entities, each entity referred to be a *device*. Devices implement different functionality like algorithms for detector reconstruction, calibration and global reconstruction. Other devices implement data transport functionality. All devices exchange data via a messaging system and bind to the flow of data through the same interface. The current framework prototype uses the FAIR messaging framework *FairMQ*. FairMQ provides functionality to establish connections between devices. Data is exchanged in
the form of messages. Each channel provides message queues of configurable depth which allows asynchronous processing. As soon as a device has finished the processing of a data sample, the result is sent to the output queue for transmission and the next sample is read from the input queue. FairMQ can use different transport layers. ZeroMQ is used in the prototype.

The framework supports two paradigms of parallel processing. In the utilized multi-process strategy, processing of the full data stream is distributed over multiple devices on multiple nodes. Furthermore, multi-threading is utilized in the devices, both framework functionality and implemented algorithms make use of threads on the specific host machine.

2.3. Prototype processing topology: TPC online reconstruction

The prototype processing topology models online reconstruction for the ALICE Time Projection Chamber (TPC). The setup includes the following devices in various multiplicities as outlined in Figure 2:

1. ClusterPublisher devices read the binary cluster data from disk and publish them into the system. There are 216 publishers.
2. DataMerger devices collect the input of 6 publishers running on the same physical node and provide one data stream to the FLPsender device.
3. The FLPsender and EPNreceiver devices build the data distribution network.
4. Every EPNreceiver device forwards the complete sample to a Tracker devices and further on to a TrackMerger device. All three devices are running in a group on the same node. The EPN group is multiplied.
5. A DataCollector component collects the data streams which have been split to separate branches.

![Figure 2](image.png)

**Figure 2.** Schematic overview of the prototype processing topology. The shadowed boxes depict processing nodes, while the horizontal rectangular boxes indicate processes.
A couple of assumptions build the basis for the prototype development strategy. (1) The focus is on processing of data from the TPC because 92.5% of the data is generated by the TPC. (2) The data from the TPC front-end will arrive via multiple links in the FLP nodes. For a realistic setup the present readout layout with 216 optical input links is used. (3) Local cluster reconstruction is running on hardware accelerator cards in real-time on the input streams. Processing in the facility starts with clusters (space points) in the main memory of FLP nodes.

3. Performance of Processing Topology

3.1. Hardware infrastructure

The ALICE O$^2$ prototype has been tested on a small scale cluster consisting of about 40 nodes from the ALICE HLT cluster purchased in 2011. The production cluster nodes have been replaced in preparation for Run 2, the former nodes are now being used in a test cluster. The available nodes are partly 16 core Intel® Xeon® 2.26 GHz and 24 core AMD Opteron® 2.1 GHz machines. The AMD nodes are equipped with NVIDIA GTX480 GPUs. These are used as accelerator cards for particle track finding. Every machine in the cluster is interconnected with an InfiniBand network in a flat topology for the main application data flow using IP over InfiniBand as network protocol. An additional Gigabit Ethernet is used for management or monitoring. The hardware setup is described in [2].

3.2. Test data

Test data samples have been prepared from recorded data of the 2011 Pb-Pb campaign. Raw data have been extracted from Grid storage and binary data blocks for TPC online clusters have been produced for a sample of $\sim 1.5 \cdot 10^4$ events of run 167808. The distribution of number of clusters reflects the centrality distribution of a minimum bias data sample, see Figure 3. The same distribution is expected for individual collisions in the upgraded detector. Due to continuous readout in time intervals, input data for the O$^2$ facility will be a superposition of data from individual collisions plus effects from the detector response.

3.3. Event processing performance

As a test case, the TPC online track reconstruction of the ALICE HLT has been integrated into the processing topology [3]. The track finding component is based on the Cellular Automaton (CA) algorithm which delivers candidates for particle tracks by connecting the reconstructed spacepoints of particle trajectories in the sensitive detector volume. The algorithm is highly parallizable and suited for running on hardware accelerator cards. One Graphics Processing Unit (GPU) is used per node in the test setup. In addition, two to four supporting threads on CPU cores of the processing node take care of data organization and interfacing to the framework. Figure 4 shows the processing time observed with the minimum bias test data sample. Processing time is linear in the event size given by number of spacepoints (TPC clusters) to be processed.

3.4. Processing Topology Performance using individual events

In the present processing topology, the CA Tracker device is the most time consuming part and is thus limiting the processing rate. Following the described strategy, the data path is split and
We use 25 processing groups where EPNReceiver devices aggregate data and pass them onto CA Tracker devices. A realistic situation can be simulated by using a fixed input sample rate. The system has been tested with various rates using always the same test data sample. The system handles fluctuations in CPU/GPU utilization and sample size by buffering parts of data samples in message queues. Without investing in further optimization, the highest sample rate for a stable processing topology was found to be 250 Hz in a setup with 25 parallel EPN groups. This leaves a factor of about 2.5 for the process multiplicity with respect to the theoretical limit. Using the average processing time of the CA Tracker as the slowest device, 40 ms theoretically allow for a rate of $25 \times 25 = 625$ Hz. Close to this rate, high processing load of the machines does not allow for compensation of fluctuations, which results in saturation of message queues. No stable processing topology can be achieved in such a situation and average sample rate drops significantly. Those effects are under study to further estimate parameters for stable topologies.

With an average raw data sample size of 16 MByte, the topology is processing an aggregated rate of 1.6 GByte/s for publishing data at ~100 Hz, and 4 GByte/s for a sample rate of 250 Hz. The message queuing system takes care of buffering data samples and provides the next sample to the algorithm as soon as the previous one has been processed. The algorithm has no influence on the running environment and in an ideal framework implementation, it does neither need nor get information about that. A continuous stream of data samples is provided by the framework for processing. The time which elapses from the end of processing of the previous sample to arrival of the next one is an important monitoring parameter and is referred to be system turnaround time.

Figure 5 shows the measurement of system turnaround time for the individual processing devices of the test topology. Following the fixed sample rate of 250 Hz, turnaround time is 4.1 ms at ClusterPublishers with very little variation. The CA Tracker and CA GlobalMerger devices are running in 25 processing branches and see on average ~100 ms turnaround time. The Data Collector receives data samples from all branches, a turnaround time of again 4.1 ms is measured. As expected, variations are increasing from stage to stage while turnaround times are consistent with the setup and stable over running time.

Figure 6 shows the system turnaround time of the CA Tracker devices with respect to the size of the sample described by the number of clusters. The observed dependency is an indication for slightly non-optimal data flow. This is a subject of further optimization and development.
3.5. Performance with extrapolated Timeframe Data

The existing TPC tracking algorithm is a solid foundation for the track reconstruction strategy of the ALICE O2 project. In order to test the algorithm with bigger data samples, timeframe-like data has been produced by overlaying clusters of individual events from the test data sample. In a first simple approach, clusters of each event are shifted in z (drift direction) by a constant offset which corresponds to the time between collisions. The shifted clusters of several events are then overlaid in data samples. Simulation of realistic detector data including distortion effects and the stochastic nature of collisions is underway and will be available for future measurements. The produced test data is still realistic in terms of sample size of a time frame. Figure 7 shows results from measurements of CA Tracker processing time.

Figure 5. System turnaround vs. sample number for the four processing stages in the prototype topology. Measurement for a fixed sample rate of 250 Hz.

Figure 6. System turnaround time of the CA Tracker vs. number of clusters. We observe a dependency on the sample size. Ideally, turnaround time does not depend on the data size as the message queues hold data samples for all stages. The observed dependency is weak compared to the actual processing time of the algorithm but an indication for a slightly non-optimal data flow.
Figure 7. CA Tracker processing time vs. number of clusters. The measurement indicates that linear trend with sample size is continuing. The tests are currently limited by the internal memory of GPU cards on the test cluster. As a next step, CA Tracker device can be extended to provide cluster data to the GPU in a sliding window approach. Such an implementation will be necessary even with more modern hardware as the internal memory will not be sufficient for the data of a complete time frame.

3.6. Data Transportation

The FLP to EPN data transportation can split the downstream data processing into a variable number of data flow branches. While the number of FLPsender devices is determined by the detector input and thus fixed, the number of EPNreceiver devices can be adjusted according to available resources and processing requirements. The data transportation network makes sure that all parts of a time frame are transported to the same EPNreceiver device for processing. Both FLPsender and EPNreceiver are implemented as devices in the FairMQ framework.

The performance of data transportation has been tested using a processing topology with a fixed input data rate. The number of FLPsender devices is fixed while the number of EPNreceiver devices has been changed in the different measurement cycles. Using a larger number of EPNreceiver devices allows to distribute data samples to many downstream processing branches. Consequently, data rate and processing load in individual EPN groups decrease with increasing number of EPNreceiver devices in the configuration.

Table 1 and Figure 8 summarize the results for EPNreceiver scaling. A small collection of central events from the test data sample with an average sample size of 90 MByte has been used at different fixed input sample rates. Multiple EPN groups running on a single physical node have been used to allow more EPN groups than the number of available machines. The nodes can handle up to 4 parallel groups. The input data rate is limited by the CPU consumption of the EPNreceiver device, the effect can be seen as decreasing slope at small n in the top left plot of Figure 8. A sustained data rate of up to 510 MByte/s has been measured. We conclude a linear scaling in 1/n.

Table 1. Input data rates of EPNreceiver devices in MByte/s for a fixed sample rate of 6 Hz and an average sample size of 90 MByte.

| Setup      | nEPN | 1   | 2   | 4   | 8   | 16  | 24  | 32  | 40  | 64  |
|------------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 EPN/node |      | 511.0 | 255.5 | 127.7 | 63.8 | 31.8 | 21.3 |     |     |     |
| 2 EPN/node |      | 255.5 | 127.7 | 63.6 | 31.8 | 21.2 | 15.9 | 12.8 | 10.6 |
| 4 EPN/node |      | 127.1 | 63.6 | 31.8 | 21.2 | 15.9 | 12.7 | 10.6 | 8.0  |

1 A. Rybalchenko: Efficient time frame building for online data reconstruction in ALICE experiment, presented at 21st International Conference on Computing in High Energy and Nuclear Physics, Okinawa, Japan 2015.
Figure 8. Data rate on a single EPNreceiver device in a setup of variable number of EPNreceiver devices at fixed sample rate. Top left: Fixed sample rate of 100 Hz, top right and right: 6 Hz adjusted to a stable topology using one EPNreceiver. Right: data rate vs. $1/n$ shows linear scaling.

4. Summary
A prototype for the ALICE O$^2$ computing facility has been successfully developed and tested. The setup includes online reconstruction devices for data samples from the Time Projection Chamber in a system with 36 FLPsender and 25 EPNreceiver devices. We have been running realistic processing topologies based on reconstruction algorithms of the current ALICE High Level Trigger and new data transport technology provided by FairMQ with the underlying ZeroMQ transport layer. The prototype implementation proves suitability of the transport layer and the concept of distributed processing topologies.

A realistic processing topology for the TPC track reconstruction is stably running at 250 Hz for real minimum bias Pb-Pb events. The framework provides efficient tools to measure performance, investigate operation, and further optimize the system. Data have been extrapolated to realistic sample sizes. As one example, the suitability of the TPC CA tracking algorithm has been shown here.

Tests with prototype versions of FLPsender and EPNreceiver devices show that the performance of the data transport is close to the requirements. A sustained data aggregation rate up to 510 MByte/s per EPNreceiver can be achieved in the system. The limitations are given by the CPU consumption of the processes on a fairly old hardware setup. The measured performance shows that a target data rate like in the final system can be achieved with 2000 EPN nodes of the current hardware.

References
[1] The ALICE Collaboration. Technical Design Report for the Upgrade of the Online-Offline Computing System. ALICE- TDR-019, 2015. CERN-LHCC-2015-006.
[2] A. Szostak et al. The ALICE Collaboration. Operational experience with the ALICE High Level Trigger. Journal of Physics: Conference Series 396 (2012). doi: 10.1088/1742-6596/396/1/012048
[3] D. Rohr et al. The ALICE Collaboration. ALICE HLT TPC Tracking of Pb-Pb Events on GPUs. Journal of Physics: Conference Series 396 (2012). doi: 10.1088/1742-6596/396/1/012044