Computer Physics Communications 242 (2019) 25–48

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Real-time data processing in the ALICE High Level Trigger at the LHC

ALICE Collaboration[1]

ARTICLE INFO

Article history:
Received 4 February 2019
Received in revised form 9 April 2019
Accepted 19 April 2019
Available online 26 April 2019

Keywords:
FPGA
GPU

ABSTRACT

At the Large Hadron Collider at CERN in Geneva, Switzerland, atomic nuclei are collided at ultra-relativistic energies. Many final-state particles are produced in each collision and their properties are measured by the ALICE detector. The detector signals induced by the produced particles are digitized leading to data rates that are in excess of 48 GB/s. The ALICE High Level Trigger (HLT) system pioneered the use of FPGA- and GPU-based algorithms to reconstruct charged-particle trajectories and reduce the data size in real time. The results of the reconstruction of the collision events, available online, are used for high level data quality and detector-performance monitoring and real-time time-dependent detector calibration. The online data compression techniques developed and used in the ALICE HLT have more than quadrupled the amount of data that can be stored for offline event processing.

Outline of this article

In the following, after introducing the ALICE (A Large Ion Collider Experiment) apparatus and highlighting specific detector subsystems relevant to this article, the ALICE High Level Trigger (HLT) architecture and the system software that operates the compute cluster are presented. Thereafter, the custom Field Programmable Gate Array (FPGA) based readout card, which is employed to receive data from the detectors, is described. An overview of the most important processing components employed in the HLT follows. The updates made to the HLT for LHC Run 2, that provided the capability to operate at twice the event rate compared to LHC Run 1, are discussed. The track and event reconstruction methods used, along with the quality of their performance are highlighted. The presentation of the ALICE HLT is concluded with an analysis of the maximum feasible data and event rates, along with an outlook in particular to LHC Run 3.

1. The ALICE detector

The ALICE apparatus [1] comprises various detector systems (Fig. 1), each with its own specific technology choice and design, driven by the physics requirements and the experimental conditions at the LHC [2]. The most stringent design constraint is the extreme charged particle multiplicity density ($\mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta$) in heavy-ion collisions, which was measured at midrapidity to be 1943 ± 54 in the 5% most central (head-on) Pb–Pb events at $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV [3]. The main part of the apparatus is housed in a solenoidal magnet, which generates a field of 0.5 T within a volume of 1600 m^3. The central barrel of ALICE is composed of various detectors for tracking and particle identification at midrapidity. The main tracking device is the Time Projection Chamber (TPC) [4]. In addition to tracking, it provides particle identification information via the measurement of the specific ionization energy loss ($\mathrm{d}E/\mathrm{d}x$). The momentum and angular resolution provided by the TPC is further enhanced by using the information from the six layer high-precision silicon Inner Tracking System (ITS) [5], which surrounds the beam pipe. Outside the TPC there are two large particle identification detectors: the Transition Radiation Detector (TRD) [6] and the Time-Of-Flight (TOF) [7]. The central barrel of ALICE is augmented by dedicated detectors that are used to measure the energy of photons and electrons, the Photon Spectrometer (PHOS) [8] and ElectroMagnetic Calorimeter (EMCal) [9]. In the forward direction of one of the particle beams is the muon spectrometer [10], with its own large dipole magnet. In addition, there are other fast-interaction detectors including the V0, T0 [11], and Zero Degree Calorimeter (ZDC) [12]. As the TPC is the most relevant for the performance of the HLT a more detailed description of it follows.

The TPC is a large cylindrical, gas-filled drift detector with two readout planes at its end-caps. A central high voltage membrane provides the electric drift field and divides the total active volume of 85 m^3 into two halves. Each charged particle traversing the gas in the detector volume produces a trace of ionization along its own trajectory. The ionization electrons drift towards the readout planes, which are subdivided into 18 trapezoidal readout sectors. The readout sectors are segmented into 15 488 readout pads each, arranged in 159 consecutive rows in radial direction. Upon their arrival at the readout planes, ionization electrons induce electric signals on the readout pads. For an issued readout trigger, the signals are digitized by a 10 bit ADC at a frequency of 10 MHz, sampling the maximum drift time of about 100 μs into 1000 time bins. This results in a total of $5.5 \cdot 10^8$ ADC samples containing

[1] See Appendix for the list of collaboration members.

https://doi.org/10.1016/j.cpc.2019.04.011
0010-4655/

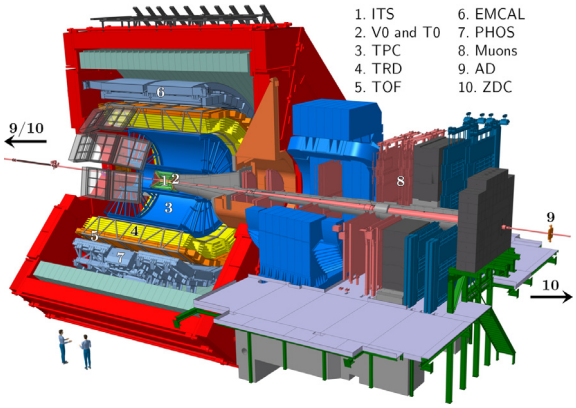


Fig. 1. The ALICE detector system at the LHC.

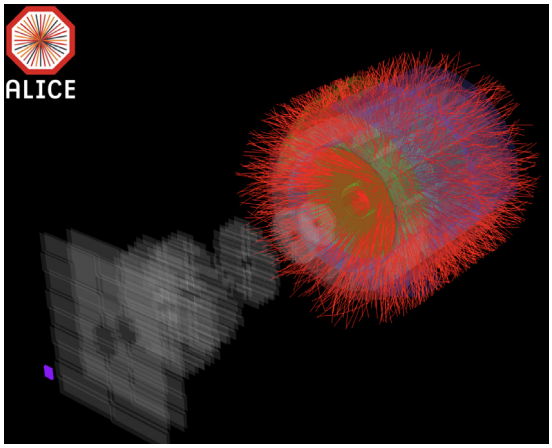


Fig. 2. Visualization of a heavy-ion collision recorded in ALICE with tracks reconstructed in real time on the GPUs of the HLT.

the full digitized TPC pulse height information. The size of data corresponding to a single collision event is about 700 MB. A zero-suppression algorithm implemented in an ASIC reduces the proton–proton TPC event size to typically 100 kB. The exact event size depends on the background, trigger setting, and interaction rate. Central Pb–Pb collisions produce up to 100 MB of TPC data, which can grow up to around 200 MB with pile-up. The TPC is responsible for the bulk of the data rate in ALICE. In Run 2, when operated at event rates of up to 2 kHz (pp and p–Pb) and 1 kHz (Pb–Pb), it reads out up to 40 GB/s. In addition, the total readout rate has a contribution of a few GB/s from other ALICE detectors, some of them operating at trigger rates up to 3.5 kHz. The volume of data taken at these rates exceeds the capacity for permanent storage considerably.

The amount of data that is stored can be reduced in a number of ways. The most widely used methods are compression of raw data (using either lossless or lossy schemes) and online selection of a subset of physically interesting events (triggering), which discards a certain fraction of the data read out by the detector [13–15]. A hierarchical trigger system performs this type of selection by having the lower hardware levels base their decision only on a subset of the data recorded by trigger detectors. The highest trigger level is the software-based High Level Trigger (HLT), which has access to the entire detector data set.

2. The High Level Trigger (HLT)

2.1. From LHC Run 1 commissioning to LHC Run 2 upgrades

A first step in transforming raw data to fully reconstructed physics information in real time was achieved with the beginning of LHC Run 1 on November 23rd, 2009, when protons collided in the center of the ALICE detector for the first time. On the morning of December 6th, stable beams at an energy of 450 GeV per beam were delivered by the LHC for the first time, and the HLT reconstructed the first charged-particle tracks from pp collisions by processing data from all available ALICE detectors. Though the HLT was designed as trigger and was operated as such at the start of Run 1, the collaboration found that by using it for data compression one could record all data to storage, thus optimizing the use of beam time. This was possible due to the quality of the online reconstruction and the increased bandwidth to storage. Throughout Run 1 the HLT was successful as an online reconstruction and data compression facility.

After the LHC Run 1, that lasted to the beginning of 2013, parts of the ALICE detector were upgraded for LHC Run 2, which started in 2015. The most important change was the upgrade of the TPC readout electronics, employing a new version of the Readout Control Unit (RCU2) [16] which uses the updated optical link speed of 3.125 Gbps instead of the previous readout rate of 2.125 Gbps. The upgrades, along with an improved TPC readout scheme, doubled the theoretical maximum TPC readout data rate to 48 GB/s, thus allowing ALICE to record twice as many events. In addition, the HLT farm underwent a consolidation phase during that period in order to be able to cope with the increased data rate of Run 2. This update improved several parts of the HLT based on the experience from Run 1. While the HLT processed up to 13 GB/s of TPC data in Run 1 [17], the new HLT infrastructure allows for the processing of the full 48 GB/s (see Section 4). Fig. 2 shows a screenshot of the online event display during a Run 2 heavy-ion run[2] with active GPU-accelerated online tracking in the HLT, of which will be described in the following.

2.2. General description

The main objective of the ALICE HLT is to reduce the data volume that is stored permanently to a reasonable size, so to fit in the allocated tape space. The baseline for the entire HLT operation is full real-time event reconstruction. This is required for more elaborate compression algorithms that use reconstructed event properties. In addition, the HLT enables a direct high-level online Quality Assurance (QA) of the data received from the detectors, which can immediately reveal problems that arise during data taking. Several of the ALICE sub-detectors (like the TPC) are so called drift-detectors that are sensitive to environmental conditions like ambient temperature and pressure. Thus a precise event reconstruction requires detector calibration, which in turn requires results from a first reconstruction as input. It is natural then to perform as much calibration as possible online in the HLT, which is also immediately available for offline event reconstruction, and thus reduces the required offline compute resources. In summary, the HLT tasks are online reconstruction, calibration, quality monitoring and data compression.

The HLT is a compute farm composed of 180 worker nodes and 8 infrastructure nodes. It receives an exact copy of all the data from the detector links. After processing the data, the HLT sends its reconstruction output to the Data Acquisition (DAQ) via dedicated optical output links. Output channels to other systems for QA histograms, calibration objects, etc., are described later in this paper. In addition the HLT sends a trigger decision. The

[2] A run is defined as a limited period of data taking with similar detector and data-taking conditions.

Table 1
Overview of the HLT Run 1 and Run 2 production clusters.

	Run 1 farm	Run 2 farm
CPU cores	Opteron/Xeon	Xeon E5-2697
	2784 cores, up to 2.27 GHz	4480 cores, 2.7 GHz
GPUs	64 × GeForce GTX480	180 × FirePro S9000
Total memory	6.1 TB	23.1 TB
Total nodes	248	188
Infrastructure nodes	22	8
Worker nodes	226	180
Compute nodes (CN)	95	172
Input nodes	117	*(subset of CNs)* 66
Output nodes	14	8
Bandwidth to DAQ	5 GB/s	12 GB/s
Max. input bandwidth	25 GB/s	48 GB/s
Detector links	452	473
Output links	28	28
RORC type	H-RORC	C-RORC
Host interface	PCI-X	PCI-Express
Max. PCI bandwidth	940 MB/s	3.6 GB/s
Optical links	2	12
Max. link bandwidth	2.125 Gbps	5.3125 Gbps
Clock frequency	133.3 MHz	312.5 MHz
On-board memory	128 MB	up to 16 GB

decision contains a readout list, which specifies the output links that are to be stored and are to be discarded by DAQ. A collision event is fully accepted if all detector links are allowed to store data and rejected if the decision is negative for all links. Data on some links may be replaced by issuing a negative decision for those links and injecting (reconstructed) HLT data instead. DAQ buffers all the event fragments locally and waits for the readout decision from the HLT, which has an average delay of 2–4 s for Pb–Pb data, while in rare cases the maximum delay reaches 10 s. Then, DAQ builds the events using only the fraction of the links accepted by the HLT plus the HLT payloads and moves the events first to temporary storage and later to permanent storage. Fig. 3 illustrates how the HLT is integrated in the ALICE data readout scheme.

The compute nodes use off-the-shelf components except for the Read Out Receiver Card (RORC — outlined in Section 2.3), which is a custom FPGA-based card developed for Run 1 and Run 2. During LHC Run 1 the HLT farm consisted of 248 servers including 117 dedicated Front-End Processor (FEP) nodes equipped with RORCs for receiving data from the detectors and sending data to DAQ. The remaining servers were standard compute nodes with two processors each, employing AMD Magny-Cours twelve-core CPUs and Intel Nehalem Quad-core CPUs. A subset of 64 compute nodes was equipped with NVIDIA Fermi GPUs as hardware accelerators for track reconstruction, described in Section 3.3. In addition, there were around 20 infrastructure nodes for provisioning, storage, database service and monitoring. Two independent networks connected the cluster: a gigabit Ethernet network for management and a fast fat-tree InfiniBand QDR 40 GBit network for data processing. Remote management of the compute nodes was realized via the custom developed FPGA-based CHARM card [18] that emulates and forwards a VGA interface, as well as the BMC (Board Management Controller) iKVM (Keyboard, Video, Mouse over IP) available as IPMI (Intelligent Platform Management Interface) standard in new compute nodes [19].

In 2014, a new HLT cluster was installed for Run 2 replacing the older servers, in particular the Run 1 FEP nodes, which were operational since 2008, during system commissioning. The availability of modern hardware, specifically the faster PCI Express interface and network interconnect, allowed for a consolidation of the different server types. The Run 2 HLT employs 188 ASUS ESC4000 G2S servers with two twelve-core Intel Xeon IvyBridge E5-2697 CPUs running at 2.7 GHz and one AMD S9000 GPU each. In order to exclude possible compatibility problems before purchase, a full HLT processing chain was stress tested on the SANAM [20] compute cluster at the GSI Helmholtz Centre for Heavy-Ion Research using almost identical hardware. The front-end and output functionality was integrated into 66 input nodes and 8 output nodes, where the input nodes serve also as compute nodes. They were equipped with RORCs for input and output allowing for a better overall resource utilization of the processors, while the infrastructure nodes of the same server type were kept separate. This reduction in the total number of servers also reduced the required rack-space and number of network switches and cables. Furthermore, the fast network was upgraded to 56 GBit FDR InfiniBand. Table 1 gives an overview of the Run 1 and Run 2 computing farms.

Considering the requirement of high reliability, which is driven among other things by the operating cost of the LHC, a fundamental design criterion is the robustness of the overall system with regard to component failure. Therefore, all the infrastructure nodes are duplicated in a cold-failover configuration. The workload is distributed in a round-robin fashion among all compute nodes, so that if one pure compute node fails it can easily be excluded from the data-taking period. Potentially the failover requires a reboot and a restart of the ALICE data taking. This scenario only takes a few minutes, which is acceptable given the low failure rate of the system; for instance, there were only 9 node failures in 1409 h of operation during 2016. A more severe problem would be the failure of an input node, because in that case the HLT is unable to receive data from several optical links. Even though there are spare servers and spare RORCs, manual intervention is needed to reconnect the fibers if the FEP node cannot be switched on remotely. However, this scenario occurred only twice in all the years of HLT operation (from 2009 to 2017). Since the start of Run 2, the entire production cluster is connected to an online uninterruptible power supply.

Since the installation of the Run 2 compute farm, parts of the former compute infrastructure are reused as a development cluster, to allow for software development and realistic scale testing without disrupting the data taking activities. Additionally, the development cluster is used as an opportunistic GRID compute resource (see Section 2.5) and an integration cluster for the ALICE Online–Offline (O^2) computing upgrade foreseen for Run 3 [21]. The O^2 project includes upgrades to the ALICE computing model, a software framework that integrates the online and offline data processing, and the construction of a new computing facility.

2.3. The Common Read-Out Receiver Card

The Read-Out Receiver Card (RORC) is the main input and output interface of the HLT for detector data. It is an FPGA-based server plug-in board that connects the optical detector links to the HLT cluster and serves as the first data processing stage. During Run 1 this functionality was provided by the HLT-dedicated RORC (H-RORC) [22], a PCI-X based FPGA board that connects to up to two optical detector links at 2.125 Gbps. The need for higher link rates, the lack of the PCI-X interface on recent server PCs, as well as the limited processing capabilities of the H-RORC with respect to the Run 2 data rates required a new RORC for Run 2. None of the commercially available boards were able to provide the required functionality, which led to the development of the Common Read-Out Receiver Card (C-RORC) as a custom readout board for Run 2. The hardware was developed in order to enable the readout of detectors at higher link speeds, extend the hardware-based online processing of detector data, and provide state-of-the-art interfaces with a common hardware platform.

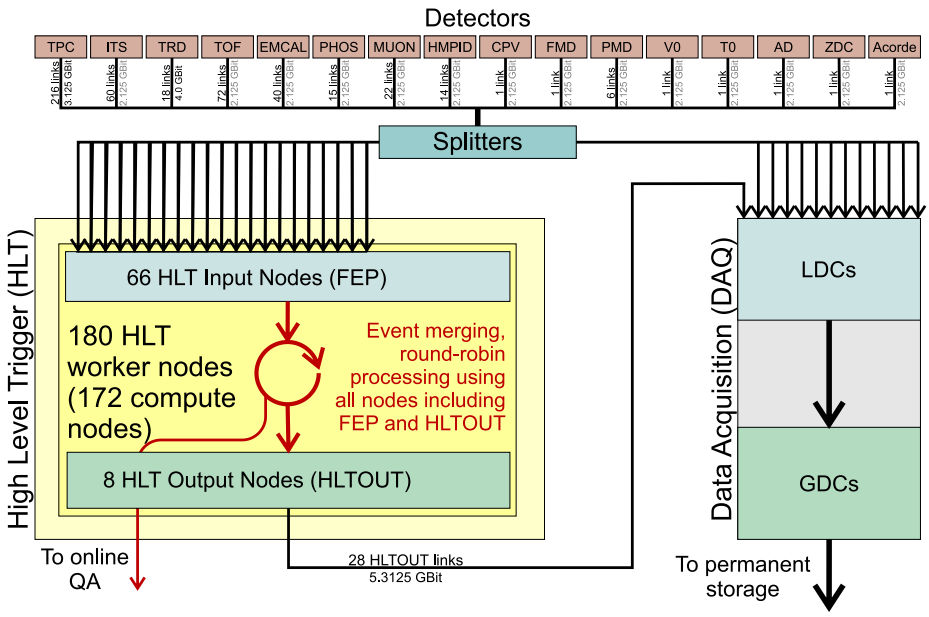


Fig. 3. The ALICE HLT in the data readout scheme during Run 2. In the DAQ system the data flow through the local and the global data concentrators, LDC and GDC, respectively. In parallel, HLT ships QA and calibration data via dedicated interfaces.

Additionally, technological advancements enabled a factor six higher link density per board and therefore reduced the number of boards required for the same amount of optical links compared to the previous generation of RORCs. One HLT C-RORC receives up to 12 links. A photograph of the board is shown in Fig. 4. The C-RORC has been part of the production systems of ALICE DAQ, ALICE HLT and ATLAS trigger and data acquisition since the start of Run 2 [23]. The FPGA handles the data stream from the links and directly writes the data into the RAM of the host machine using Direct Memory Access (DMA). A minimal kernel adapter in combination with a user space device driver based on the Portable Driver Architecture (PDA) [24] provides buffer management, flow control, and user-space access to the data on the host side. A custom DMA engine in the firmware enables a throughput of 3.6 GB/s from device to host. This is enough to handle the maximum input bandwidth of the TPC as the biggest data contributor (1.9 GB/s per C-RORC), the TRD as the detector with the fastest link speed (6 links at 2.3 GB/s per C-RORC), and a fully equipped C-RORC with 12 links at 2.125 Gbps (2.5 GB/s). The C-RORC FPGA implements a cluster finding algorithm to process the TPC raw data at an early stage. This algorithm is further described in Section 3.2. The C-RORC can be equipped with several GB of on-board memory, used for data replay purposes. Generated, simulated, previously recorded, or even faulty detector data can be loaded into this on-board RAM and played back as if it were coming via the optical links. The HLT output FPGAs can be configured in a way to discard data right before it would be sent back to the DAQ system. The data replay can be operated independently from any other ALICE online system, detector, or LHC operational state. In combination with a configurable replay event rate, the data replay functionality provides a powerful tool to verify, scale, and benchmark the full HLT system. This feature is essential for the optimizations presented in Section 4. The C-RORCs are integrated into the HLT data transport framework as data-source components for detector data input via optical links and as sink components to provide the HLT results to the DAQ system. The C-RORC FPGA firmware and its integration into the HLT is further described in [25]. The data from approximately 500 links, at link rates between 2.125 Gbps and 5.3125 Gbps, is handled via 74 C-RORCs that are installed in the HLT.

2.4. Cluster commissioning, software deployment, and monitoring

The central goal for managing the HLT cluster is automation that minimizes the need for manual interventions and guarantees that the whole cluster is in a consistent state that can be easily controlled and modified if needed. Foreman [26] is used to automatize the basic installation of the servers via PXE-boot. The operating system (OS) that is currently used on all of the servers is CERN CentOS 7. Once the OS is installed on these servers, Puppet [27] controls and applies the desired configuration to each server. Puppet efficiently integrates into Foreman and allows for servers to be organized into groups according to different roles and apply changes to multiple servers instantaneously. With this automatized setup the complete cluster can be rebuilt, including the final configuration, in roughly three hours. For both the production and development clusters several infrastructure servers are in place, providing different services like DNS, DHCP, NFS, databases, or private network monitoring. Critical services are redundant to reduce the risk of cluster failure in case there is a problem with a single infrastructure server.

The monitoring of the HLT computing infrastructure is done using the open source tool Zabbix [28]. It allows administrators to gather metrics, be aware of the nodes health status, and react to undesired states. More than 100 metrics per node are being monitored, such as temperature, CPU load, network traffic, free disk space, disk-health status, and failure rate on the network fabric. The monitoring system automatizes many tasks that would require administrators' intervention. These preemptive measures offer the possibility to replace hardware beforehand, i.e. during technical shutdowns, and to avoid failures during data taking. HLT administrators receive a daily report of the system status and, in addition, e-mail notifications when certain metrics exceed warning thresholds. For risky events there are automated actions in place. For instance, several shutdown procedures are performed when the node temperature reaches critical values, in order to prevent damage to the servers.

In addition to Zabbix, ALICE has developed a custom distributed log collector called InfoLogger. A parser script is employed that scans all error messages stored to the logs to find important problems in real time. These alerts can also help the

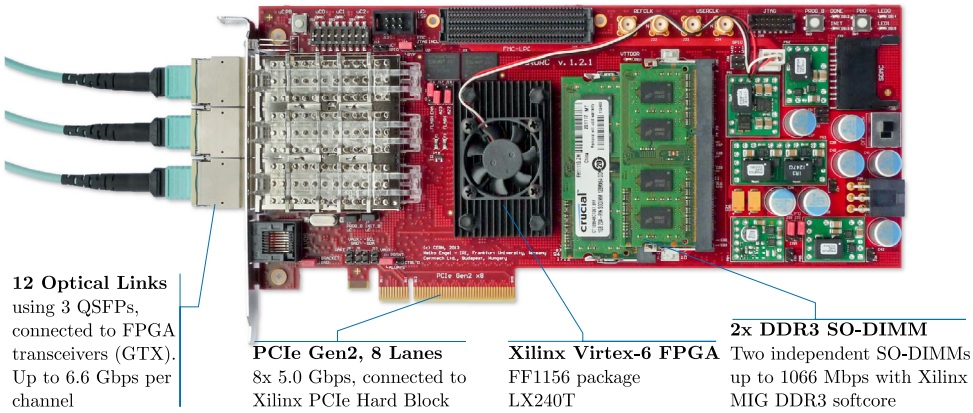


Fig. 4. The Common Read-Out Receiver Card.

detector experts with the monitoring of their systems, including automated alarms sent via e-mail or SMS.

This configuration lowers the complexity of managing a heterogeneous system with around 200 nodes for a period of at least 10 years, reducing the number of trained on-site engineers required for operation.

2.5. Alternative use cases of the HLT farm

In order to maximize the usage of the servers during times when there are no collisions, a Worldwide LHC Computing Grid (WLCG) [29] configuration was developed for the cluster in cooperation with the ALICE offline team. The first WLCG setup used OpenStack [30] Virtual Machines (VM) to produce ALICE Monte Carlo (MC) simulations of particle collision events. In 2017, the WLCG setup was improved to use Docker [31] containers instead of OpenStack VMs, which allows for more flexibility and therefore improves efficiency with the available resources. The containers are spawned for just one job and destroyed after the job finishes. During pp data taking a part of the production cluster is contributed to the WLCG setup. During phases without data taking, like LHC year-end shutdowns and technical stops, the whole HLT production cluster is operated as a WLCG site as long as it is not needed for tests of the HLT system. Fig. 5 shows the aggregated wall time of the new Docker setup from March 2017 onward. The steeper slope represents periods when the complete cluster is assigned to WLCG operation, while the plateau indicates a phase of full scale framework testing. The HLT production cluster provides a contribution to the ALICE MC simulation compute time with this opportunistic use on a best-effort basis. The WLCG setup of the HLT focuses on MC simulations because these require less storage and network resources than general ALICE Grid jobs and are thus ideally suited for opportunistic operation without side effects.

The HLT development cluster, introduced in Section 2.2, is composed of approximately 80 older servers. Not only does it allow for ongoing development of the current framework, of which runs on the production cluster, but it can also used for tests of the future framework for Run 3. During periods when no development is taking place, 60 of the nodes act as second WLCG site, in addition to the opportunistic use of the production cluster, donating the compute resources to ALICE MC jobs. To guarantee that there is no interference with data taking, the HLT development cluster is completely separated from the production environment. The development cluster is installed in different racks and also uses a different private network, which has no direct connection to the production cluster. For WLCG operation, the HLT internal networks and the network used for WLCG communication were completely separated via VLANs configured at switch level.

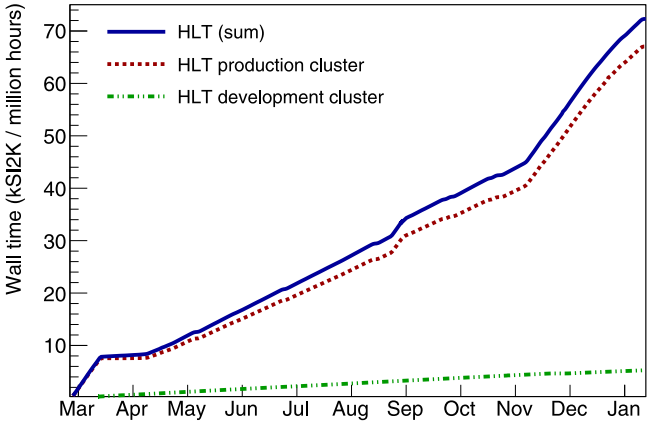


Fig. 5. Contribution of the HLT production (dotted line) and development (double-dotted dashed line) clusters to the WLCG between March 2017 and January 2018, with the sum of both contributions shown as the solid line.

2.6. HLT architecture and data transport software framework

In order to transform the raw detector signals into physical properties all ALICE detectors have developed reconstruction software, like TPC cluster finding (Section 3.2) and track finding (Section 3.3) algorithms. In the HLT the data processing is arranged in a pipelined data-push architecture. The reconstruction process starts with local clusterization of the digitized data, continues with track finding for individual detectors, and ends with the creation of the Event Summary Data (ESD). The ESD is a complex ROOT [32] data structure that holds all of the reconstruction information for each event.

In addition to the core framework described in the this section, a variety of interfaces exist to other ALICE subsystems [33]. These include the command and control interface to the Experiment Control System (ECS), the Shuttle system used for storing calibration objects for offline use, the optical links to DAQ, the online event display, and Data Quality Monitoring (DQM) for online visualization of QA histograms.

The ALICE HLT uses a modular software framework consisting of separate components, which communicate via a standardized publisher–subscriber interface designed to cause minimal overhead for data transport [34,35]. Such components can be data sources that feed into the HLT processing chain, either from the detector link or from other sources like TPC temperature and pressure sensors. Data sinks extract data from the processing chain and send the reconstructed event and trigger decision to DAQ via the output links. Other sinks ship calibration objects

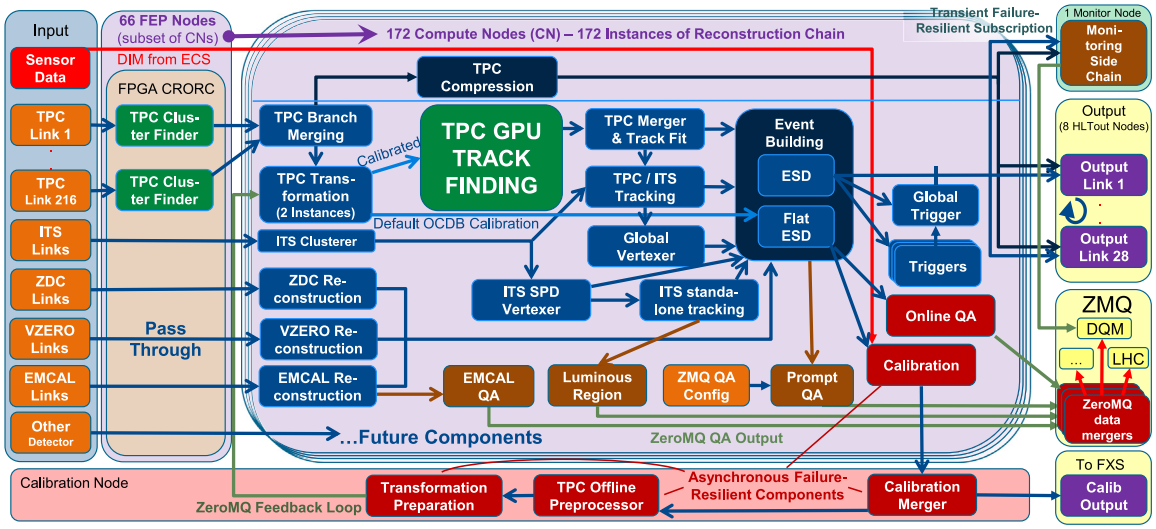


Fig. 6. Schema of the HLT components. The colored boxes represent processes accelerated by GPU/FPGA (green), normal processes (blue), processes that produced HLT output that is stored (dark blue), entities that store data (purple), asynchronous failure-resilient processes (dark red), classical QA components that use the original HLT data flow (brown), input (orange), and sensor data (red). Incoming data are passed through by the C-RORC FPGA cards or processed internally. The input nodes locally merge data from all links belonging to one event. The compute nodes then merge all fragments belonging to one event and run the reconstruction. The bottom of the diagram shows the asynchronous online calibration chain with a feedback loop as described in Section 3.4. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

or QA histograms, which are stored or visualized. In addition to source and sink components, analysis or worker components perform the main computational tasks in such a processing chain and are arranged in a pipelined hierarchy. Fig. 6 gives an overview of the data flow of the most relevant components currently running in the HLT. A component reads a data set (if it is not a source), processes it, creates the output and proceeds to the next data set. Although each component processes only one event at a time, the framework pipelines the events such that thousands of events can be either in-chain in the cluster or also on a single server. Merging of event fragments, scattering of events among multiple compute nodes for load balancing, and network transfer are all handled via special processing components provided by the framework and are transparent to the worker processes. Components situated on the same compute node pass data via a shared-memory based zero-copy scheme. With respect to Run 1 the framework underwent a revision of the interprocess-scheduling approach. The old approach, using POSIX pipes, began to cause a significant CPU load through many system calls and was consequently replaced by a shared-memory based communication.

Presently, the user simply defines the processing chain with reconstruction, monitoring, calibration, and other processing components. The user also defines the inputs for all components as well as the output at the end of the processing chain. The full chain is started automatically and distributed in the cluster. The processing configuration can be annotated with hints to guide the scheduling. In order to minimize the data transfer, the chain usually starts with local processing components on the front-end nodes (like the TPC cluster finder presented in Section 3.2). In the end, after the local steps have reduced the data volume, all required event fragments are merged on one compute node for the global event reconstruction.

The data transport framework is based on three pillars. There is a primary reconstruction chain which processes all the recorded events in an event-synchronous fashion. It performs the main reconstruction and data compression tasks and is responsible for receiving and sending data. This main chain is the backbone of the HLT event reconstruction and its stability is paramount for the data taking efficiency of ALICE.

The second pillar is the data monitoring side chains, which run in parallel at low rates on the compute nodes. These subscribe transiently to the output of a component of the main chain. In this way, the side chains cannot break or stall the HLT main chain.

For Run 2 a third pillar was added, based on Zero-MQ (Zero Message Queue) message transfer [36], which provides similar features compared to the main chain but runs asynchronously. Currently, it is used for the monitoring and calibration tasks and does not merge fragments of one event but instead it is fed with fully reconstructed events from the main chain. It processes as many events as possible on a best-effort basis, skipping events when necessary. Results of the distributed components are merged periodically to combine statistics processed by each instance. The same Zero-MQ transport is also used as an interface to DQM and as external interface which allows detector experts to query merged results of QA components running in the HLT.

The transport framework is not restricted to closed networks or computing clusters. A proof-of-principle test of the framework used locally in the HLT cluster deploys a global processing chain for a Grid-like real-time data processing. This framework was distributed on a North–South axis between Cape Town in South Africa and Tromsø in northern Norway, with Bergen (Norway), Heidelberg (Germany), and Dubna (Russia) as additional participating sites [37]. The concepts developed for the HLT are the basis for the new framework of the ALICE O^2 computing upgrade.

2.7. Fault tolerance and dynamic reconfiguration

Robustness of the main reconstruction chain is the most important aspect from the point of view of data taking efficiency. Therefore, the HLT was designed with several failure resiliency features. All infrastructure services run on two redundant servers and compute node failures can be easily compensated for. Experimental and non-critical components can run in a side-chain or asynchronously via Zero-MQ, separate from the main chain.

Also the main chain itself has several fault tolerance features. Some components use code from offline reconstruction, or code written by the teams responsible for certain detector development, and hence they are not developed considering the high-reliability requirements of the HLT. Nevertheless, the HLT must still ensure stable operation in case of critical errors like segmentation faults. Thus, all components run in different processes,

which are isolated from each other by the operating system. In case one component fails, the HLT framework can transparently cease the processing of that component for a short time, and then later restart the component. Although the event is still processed, the result of that particular component for this event and possibly several following events are lost. This loss of a single instance causes only a marginal loss of information.

3. Fast algorithms for fast computers

Since the TPC produces 91.1% (Pb–Pb) and 95.3% (pp) of the data volume[3] and, also because of the sheer data volume, event reconstruction of the TPC data including clusterizing and tracking is the most compute intensive task of the HLT. This makes the TPC the central detector for the HLT. Its raw data are the most worthwhile target for data compression algorithms. Since a majority of the compute cycles are spent processing TPC data, it is mandatory that the TPC reconstruction code is highly efficient. It is the TPC reconstruction that leverages the compute potential of both the FPGA and GPU hardware accelerators in the HLT. Furthermore, since it is an ionization detector, TPC calibration is both challenging and essential.

Here, a selection of important HLT components, following the processing of the TPC data in the chain is described. The processing of the TPC data starts with the clusterization of the raw data, which happens in a streaming fashion in the FPGA while the data are received at the full optical speed. Two independent branches follow, where one component compresses the TPC clusters and replaces the TPC raw data with compressed HLT data. The second branch starts with the TPC track reconstruction using GPUs, continues with the creation of the ESD, and runs the TPC calibration and QA components.

3.1. Driving forces of information science

The design of the ALICE detector dates back two decades. At that time, the LHC computing needs could not be fulfilled based on existing technology but relied on extrapolations according to Moore's Law [38]. Indeed the performance of computers has improved by more than three orders of magnitude since then, but the development of microelectronics has reached physical limits in recent years. For example, processor clock rates have not increased significantly since 2004. To increase computing power various levels of parallelization are implemented, such as the use of multi- or many-core processors, or by supporting SIMD (Single Instruction, Multiple Data) vector-instructions. At this point in time computers do not become faster for single threads but they can become more powerful if parallelism is exploited. Although these developments were only partially foreseeable at the beginning of the ALICE construction phase, they have been taken into account for the realization of the HLT.

3.2. Fast FPGA cluster finder for the TPC

At the beginning of the reconstruction process the so-called clusters of locally adjacent signals in the TPC have to be found. Fig. 7 shows a schematic representation of a cross-section of a trapezoidal TPC sector, where the local coordinate system is such that in the middle of the sector the x-axis points away from the interaction point. One can imagine a stack of 2D pad-time planes (y–z plane in Fig. 7) in which a charged particle traversing the detector creates several neighboring signals in each 2D plane. The exact position of the intersection between the charged-particle trajectory and the 2D plane can be calculated by using

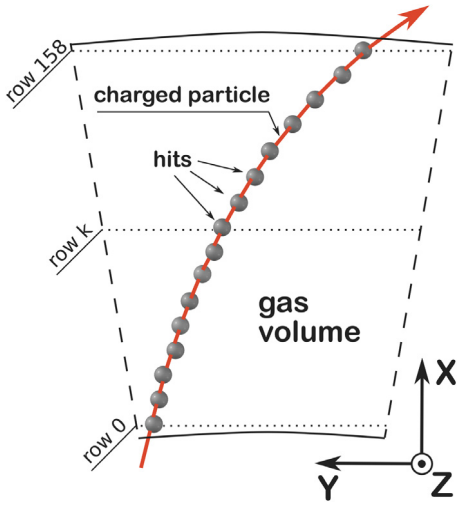


Fig. 7. Schematic representation of the geometry of a TPC sector. Local y and z coordinates of a charged-particle trajectory are measured at certain x positions of 159 readout rows, providing a chain of spatial points (hits) along its trajectory.

the weighted mean of the signals in the plane, i.e. by determining their center of gravity. The HLT cluster-finder algorithm can be broken down into three separate steps. Firstly, the relevant signals have to be extracted from raw data and the calibration factors are applied. Next, neighboring signals and charge peaks in time-direction are identified and the center of gravity is calculated. Finally, neighboring signals in the TPC pad-row direction (x-y plane) are merged to form a cluster. These reconstructed clusters are then passed on to the subsequent reconstruction steps, such as the track finding described in Section 3.3.

By design, the TPC cluster-finder algorithm is ideally suited for the implementation inside an FPGA [39], which supports small, independent and fast local memories and massively parallel computing elements. The three processing steps are mutually independent and are correspondingly implemented as a pipeline, using fast local memories as de-randomizing interfaces between these stages. In order to achieve the necessary pipeline throughput, each pipeline stage implements multiple custom designed arithmetic cores. The FPGA based RORCs are required as an interface of the HLT farm to the optical links. By placing the online processing of the TPC data in the FPGA, the data can be processed on-the-fly. The hardware cluster finder is designed to handle the data bandwidth of the optical link. Finally, a compute node receives the TPC clusters, computed in the FPGA, directly into its main memory.

An offline reference implementation of the cluster finding exists but is far too slow to be implemented online. Rather, the offline cluster finder is used as a reference for both the physics performance and the processing speed. In comparison to the hardware cluster finder executed on the FPGA, it performs additional and more complex tasks. These include checking TPC readout pads for baseline shifts and, if present, applying corrections and deconvoluting overlapping clusters using a Gaussian fit to the cluster shapes, which are simply split in the hardware version. Additional effects such as missing charge in the gaps between TPC sectors and malfunctioning TPC channels are considered. Finally, after the application of the drift-velocity calibration, cluster positions are transformed into the spatial x, y, and z coordinate system. In the HLT, a separate transformation component performs this spatial transformation as a later step. The evaluation in Section 3.2.1 demonstrates that the HLT hardware cluster finder delivers a performance comparable to the offline cluster finder.

[3] Values from the 8 kHz Pb–Pb and 200 kHz pp data taking runs of 2015.

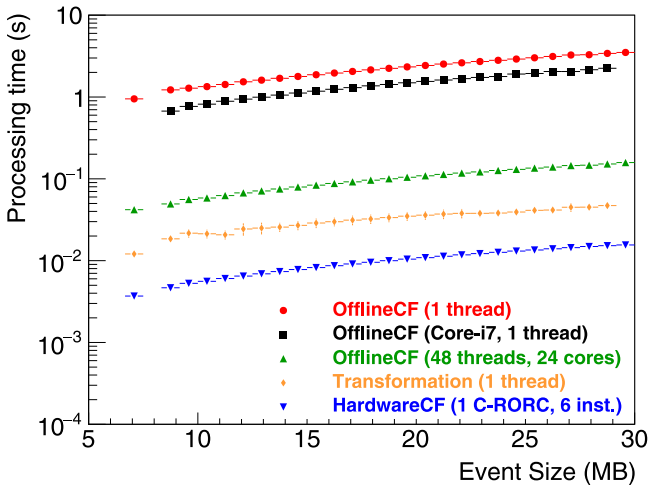


Fig. 8. Processing time of the hardware cluster finder and the offline cluster finder. The measurements were performed on an HLT node (circles, triangles, diamonds), a newer Core-i7 6700K CPU (squares), and on the C-RORC (inverted triangles).

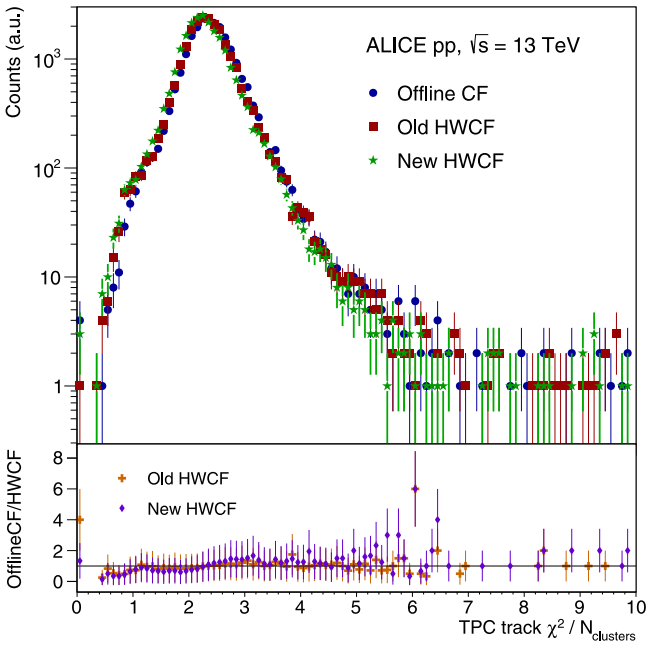


Fig. 9. The upper panel shows the distribution of TPC track χ^2 residuals from offline track reconstruction obtained using total cluster charges from offline cluster finder (Offline CF) and different versions of the HLT hardware cluster finder (HWCF). Tracks, reconstructed using the TPC and ITS points, satisfy the following selection criteria: pseudorapidity $|\eta| < 0.8$ and $N_{\text{TPC clusters}} \geq 70$. The ratios of the distributions obtained using the offline cluster finder and the HLT cluster finder are shown in the lower panel.

Benchmarks have shown that one C-RORC with six hardware cluster finder (HardwareCF) instances is about a factor 10 faster than the offline cluster finder (OfflineCF) using 48 threads on an HLT node, as shown in Fig. 8. The software processing time measurements were done on a HLT node with dual Xeon E5-2697 CPUs for the single-threaded variant, the multi-threaded variant as well as the cluster transformation component. The single-threaded variant was also evaluated on a Core-i7 6700k CPU to show the performance improvements of using the same implementation on a newer CPU architecture. The measurements were also performed on the C-RORC.

Several factors increase the load on the hardware cluster finder in Run 2. The C-RORC receives more links than the former H-RORC of Run 1, with the FPGA implementing six instead of the previously two instances of the cluster finder. The TPC RCU2 sends the data at a higher rate, up to 3.125 Gbps. In addition, during 2015 and 2016, the TPC was operated with argon gas instead of neon yielding a higher gain factor, which resulted in a higher probability of noise over the zero-suppression threshold. In this situation, the cluster finder detects a larger number of clusters, though a significantly large fraction of these are fake. In addition, the readout scheme of the RCU2 was improved, disproportionately increasing the data rate sent to the HLT compared to the link speed, yielding a net increase of a factor of 2. These modifications also required the clock frequency of the hardware cluster finder to be disproportionately scaled up compared to the link rate in order to cope with the input data rates. Major portions of the online cluster finder were adjusted, further pipelined, and partly rewritten to achieve the required clock frequency and throughput. The peak-finding step of the algorithm was replaced with an improved version more resilient to noise. This filtering reduces the number of noise induced clusters found, relaxes the load on the merging stage, and thus reduces the cluster finder output data size. The reduced output size, in combination with improvements to the software based data compression scheme, increases the overall data compression factor of the HLT (see Section 3.5).

3.2.1. Physics performance of the HLT cluster finder for the TPC

In order to reduce the amount of data stored on tape, the TPC raw data are replaced by clusters reconstructed in the HLT. The cluster-finder algorithm must be proven not to cause any significant degradation to the physical accuracy of the data. The offline track reconstruction algorithm was improved by better taking into account the slightly different behavior of the HLT cluster finder and its center of gravity approach compared to the offline cluster finder. The performance of the algorithm has been evaluated by looking at the charged-particle tracks reconstructed with the improved version of the offline track-reconstruction algorithm, described in Section 3.3.

The important properties of the clusters are the spatial position, the width, and the charge deposited by the traversing particle. Fig. 9 compares the χ^2 distribution of TPC tracks reconstructed by the offline tracking algorithm using TPC clusters produced using either the HLT hardware cluster finder or the offline version. Since the cluster errors coming from a fit to the track are parameterized and not derived from the width of the cluster, the χ^2 distribution is proportional to the average cluster-to-track residual. On a more global level, the cluster positions in the ITS are used to evaluate the track resolution of the TPC. The TPC track is propagated through the ITS volume and the probability of finding matching ITS spatial points is analyzed. Since the ITS cluster position is very precise it is a good metric for TPC track quality. However, because the occupancy for heavy-ion collisions is high, the matching requires an accurate position of the TPC track with a good transverse momentum (p_{T}) fit for precise extrapolation. It was found that there are no significant differences in track resolution and χ^2 between the offline cluster finder and the new HLT cluster finder, with the old HLT hardware cluster finder yielding a slightly worse result.

Fig. 10 shows the $\mathrm{d}E/\mathrm{d}x$ separation power as a measure of the quality of the HLT cluster charge reconstruction. Here, the separation power is defined as the $\mathrm{d}E/\mathrm{d}x$ separation between the pions and electrons scaled by the resolution. Since the $\mathrm{d}E/\mathrm{d}x$ is calculated from the cluster charge, an imprecise charge information would deteriorate the $\mathrm{d}E/\mathrm{d}x$ resolution and consequently separation power. Within the statistical uncertainty no substantial difference is observed between the offline and hardware cluster-finder algorithms.

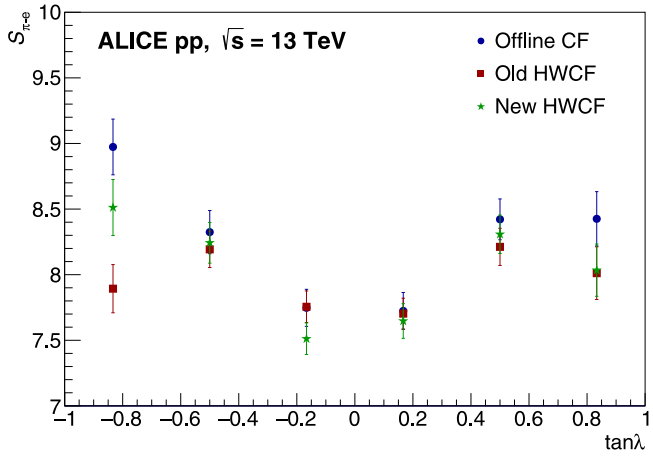


Fig. 10. Separation power ($S_{\pi-e}$) of pions and electrons (minimum ionizing particles, i.e. pions at 0.3 to 0.6 GeV/c versus electrons from gamma conversions at 0.35 to 0.5 GeV/c) as a function of the track momentum dip angle, where $\tan\lambda = p_z/p_T$. Comparison of dE/dx separation power using total cluster charges from Offline CF and different versions of the HWCF.

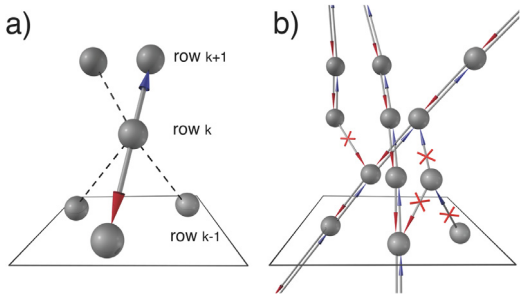


Fig. 11. Cellular automaton track seeding steps. (a) Neighbor finder. Each cluster at a row k is linked to the best pair of its neighbors from the next and the previous row. (b) Evolution step. Non-reciprocal links are removed, chains of reciprocal links define the tracklets.

3.3. Track reconstruction in the TPC

In ALICE there are two different TPC-track reconstruction algorithms. One is employed for offline track reconstruction and the other is the HLT track reconstruction algorithm. In this section, the HLT algorithm is described and its performance compared to that of the offline algorithm.

In the HLT, following the cluster finder step, the reconstruction of the trajectories of the charged particles traversing the TPC is performed in real time. The ALICE HLT is able to process pp collisions at a rate of 4.5 kHz and central heavy-ion collisions at 950 Hz (see Section 4), corresponding to a data rate of 48 GB/s, which is above the maximum deliverable rate from the TPC.

The TPC track reconstruction algorithm has two steps, namely the in-sector track-segment finding within individual TPC sectors and the segment merger, which concludes with a full track refit. The in-sector tracking is the most compute intense step of online event reconstruction, therefore it is described in more detail in the following subsection.

3.3.1. Cellular automaton tracker

Based on the cluster-finder information, clusters belonging to the same initial particle trajectory are combined to form tracks. This combinatorial pattern recognition problem is solved by a track finder algorithm. Since the potential number of cluster combinations is quite substantial, it is not feasible to calculate an exact solution of the problem in real time. Therefore, heuristic methods are applied. One key issue is the dependence of reconstruction time on the number of clusters. Due to the large combinatorial background, i.e. the large number of incorrectly combined clusters from different tracks, it is critical that the dependence is linear in order to perform online event processing. This was achieved by developing a fast algorithm for track reconstruction based on the cellular automaton principle [40,41] and the Kalman filter [42] for modern processors [43]. The processing time per track is 5.4 μs on an AMD S9000 GPU. The tracking time per track increases linearly with the number of tracks, and is thus independent of the detector occupancy, as shown in Section 3.3.4.

The track finder algorithm starts with a combinatorial search of track candidates (tracklets), which is based on the cellular automaton method. Local track segments are created from spatially adjacent clusters, eliminating non-physical cluster combinations. In the two-stage combinatorial processing, the neighbor finder matches, for each cluster at a row k, the best pair of neighboring clusters from rows $k+1$ and $k-1$, as shown in Fig. 11 (left). The neighbor selection criterion requires the cluster and its two best neighbors to form the best straight line, in addition to having a loose vertex constraint. The links to the best two neighbors are stored. Once the best pair of neighbors is found for each cluster, a consequent evolution step determines reciprocal links and removes all non-reciprocal links (see Fig. 11) (right).

A chain of at least two consecutive links defines a tracklet, which in turn defines the particle trajectory. The geometrical trajectories of the tracklets are fitted with a Kalman filter. Then, track candidates are constructed by extending the tracklets to contain clusters close to the trajectory. A cluster may be shared among track candidates; in this case it is assigned to the candidate that best satisfies track quality criteria like the track length and χ^2 of the fit.

This algorithm does not employ decision trees or multiple track hypotheses. This simple approach is possible due to the abundance of clusters for each TPC track and it results in a linear dependence of the processing time on the number of clusters.

Following the in-sector tracking the segments found in the individual TPC sectors are merged and the final track fit is performed. A flaw in this approach is that if an in-sector track segment is too short, e.g. having on the order of 10 clusters, it might not be found by the in-sector tracking algorithm. This is compensated for by a posterior step, that treats tracks ending at sector boundaries close to the inner or outer end of the TPC specially, by extrapolating the track through the adjacent sector, and picking up possibly missed clusters [44]. The time overhead of this additional step is less than 5% of the in-sector tracking time.

The HLT track finder demonstrates an excellent tracking efficiency, while running an order of magnitude faster than the offline finder, while also achieving comparable resolution. Corresponding efficiency and resolution distributions extracted from Pb–Pb events are shown in Section 3.3.4. The advantages of the HLT algorithm are a high degree of locality and the allowance of a massively parallel implementation, which is outlined in the following sections.

3.3.2. Track reconstruction on CPUs

Modern CPUs provide SIMD instructions allowing for operation on vector data with a potential to speed up corresponding to the vector width (to-date a factor up to 16 is achievable with the AVX512 instruction set). Alternatively, hardware accelerators like GPUs offer vast parallelization opportunities. In order to leverage this potential in the track finder, all the computations are implemented as a simple succession of arithmetic operations on single precision floats. An appropriate vector class and corresponding

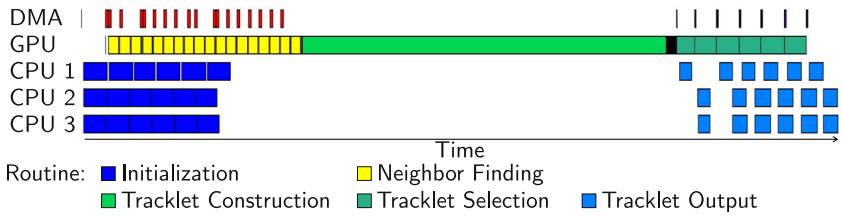


Fig. 12. Visualization of the pipelined GPU processing of the track reconstruction using multiple CPU cores to feed data to the GPU.

data structures were developed, yielding a vectorized version of the tracker that can run on both the Xeon Phi and standard CPUs using their vector instructions, or additionally in a scalar way. Data access is the most challenging part. The main difficulty is the fact that all tracklets have different starting rows, lengths, and number of clusters requiring random access into memory instead of vector loads. While the optimized and vectorized version of the Kalman filter itself yielded a speedup of around 3 over the initial scalar version, the overall speedup was however smaller. Therefore, the track reconstruction is performed on GPUs. Due to the random memory access during the search phase, it is impossible to create a memory layout optimized for SIMD. This poses a bottleneck for the GPU as well, but it is less severe due to the higher memory bandwidth and better latency hiding of the GPU. The vector library developed in the scope of this evaluation is available as the open source Vc library [45]. It was integrated into ROOT and is part of the C++ Parallelism technical specification [46]. The optimized data layout originally developed for fast SIMD access has also proven very efficient for parallelization on GPUs.

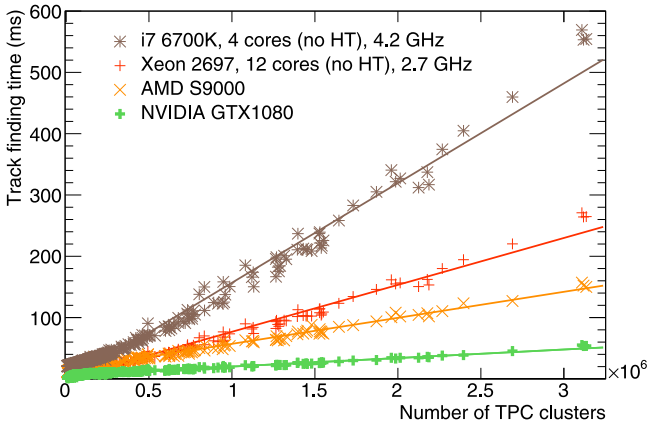


Fig. 13. Time required for execution of the tracking algorithm on CPUs and on GPUs as function of the input data size expressed in terms of the number of TPC clusters. The lines represent linear fits to the distributions. The merging and refitting times are not included in the track finding time.

3.3.3. Track reconstruction on GPUs

The alternative many-core approach using GPUs as general purpose processors is currently employed in the HLT. All steps of the cellular automaton tracker and the Kalman filter can be distributed on many independent processors. In order to be independent from any GPU vendor, the HLT code must not rely exclusively on a proprietary GPU programming framework. The fact that the reconstruction code is used in the ALICE offline framework, AliRoot, and that it is written in C++ poses several requirements on the GPU API. Currently, the HLT tracking can optionally use both the CUDA framework for NVIDIA GPUs or the OpenCL framework with C++ extensions for AMD GPUs. Even though OpenCL is an open, vendor-independent framework, the current HLT code is limited to AMD because other vendors do not yet support the C++ kernel language. C++ templates avoid code duplication for class instances residing in the different OpenCL memory scopes. The new OpenCL 2.2 standard specifies a C++ kernel language very similar to the extension currently used, which will allow for an easy migration. The tracking algorithm is written such that a common source file in generic C++ contains the entire algorithm representing more than 90% of the code. Small wrappers allow the execution of the code on different GPU models and also on standard processors, optionally parallelized via OpenMP. This aids in avoiding division between GPU and CPU code bases and thus reduces the maintenance effort [47] since improvements to the tracking algorithm are developed only once. All optimizations are parameterized and switchable, such that each architecture (CPU, NVIDIA GPU, AMD GPU) can use its own settings for optimum performance.

One such optimization for GPUs is pipelined processing: the execution of the track reconstruction on the GPU, the initialization and output merging on the CPU, as well as the DMA transfer, all happen simultaneously (Fig. 12). The pipeline hides the DMA transfer time and the CPU tasks and keeps the GPU executing kernels more than 95% of the time. On top of that, multiple events are processed concurrently to make sure all GPU compute units are always fully used [43]. One obstacle already mentioned in Section 3.3.2 is the different starting rows and lengths of tracks, which prevent optimum utilization of the GPU's single instruction, multiple thread units. A dynamic scheduling which, after processing a couple of rows, redistributes the remaining workload among the GPU threads was implemented. This reduces the fraction of wasted GPU resources due to warp-serialization due to a track that has ended while another track is still being followed.

3.3.4. Performance of the track reconstruction algorithm

The dependence of the tracking time on input data size expressed in terms of the number of TPC clusters is shown in Fig. 13. The hardware used for the HLT performance evaluation is the hardware of the HLT Run 2 farm, which consists of the already several years old Intel Xeon 2697 CPU and AMD FirePro S9000 GPU. The compute time using a modern system, i.e. an Intel Skylake CPU (i7 6700K) or NVIDIA GTX1080 GPU, is also shown and demonstrates that newer GPU generations yield the expected speedup. On both CPU and GPU architectures, the compute time grows linearly with the input data size. For small events, the GPU cannot be fully utilized and the pipeline-initialization time becomes significant, yielding a small offset for empty events. With no dominant quadratic complexity in the tracking algorithm an excellent scaling to large events is achieved. The CPU performance is scaled to the number of physical CPU cores via parallel processing of independent events, which scales linearly, while the tracking on GPUs processes a single event in one go. Only one CPU socket of the HLT Run 2 farm's server is used to avoid NUMA (Non Uniform Memory Architecture).

The overall speedup achieved by the HLT GPU tracking is shown in Fig. 14. It is computed as the ratio of the processing time of offline (CPU) tracking and the single-core processing time

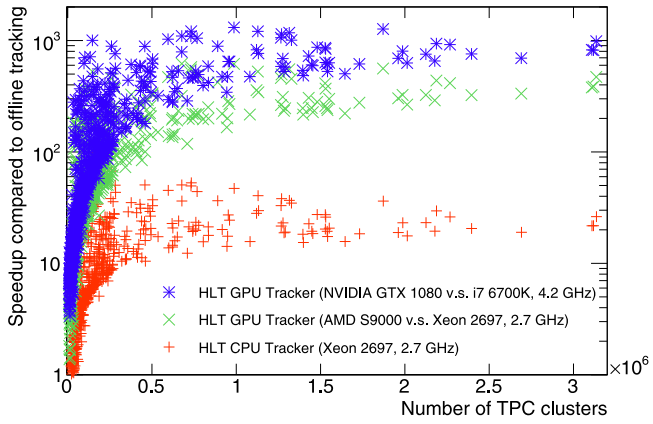


Fig. 14. Speedup of HLT tracking algorithm executed on GPUs and CPUs compared to the offline tracker normalized to a single core and corrected for the serial processing part that the CPU contributes to GPU tracking as a function of the input data size expressed in terms of the number of TPC clusters. The plus markers show the speedup as a function of the number of TPC clusters with the HLT tracking executed on the CPU. The cross(asterisk) markers show the speedup obtained with the tracking executed on a older(newer) GPU.

of GPU tracking. Here the CPU usage-time for pre- and post-processing of GPU tracking scaled by the average number of CPU cores used during the steps of GPU tracking is folded out of the total CPU-tracking time. For the CPU version of the HLT tracking algorithm, this is exactly the speedup. For the GPU version, this is the number of CPU cores equivalent, tracking-performance-wise, to one GPU. In this case, the full track reconstruction duration includes the merging and refitting time, whereas for Fig. 13 the non tracking-related steps of the offline tracking, e.g. dE/dx calculation, are disabled. Overall, the HLT tracking algorithm executed on the CPU is 15–20 times faster than the offline tracking algorithm. One GPU of the HLT Run 2 farm replaces more than 15 CPU cores in the server, for a total speedup factor of up to 300, with respect to offline tracking. The CPU demands for pre- and post-processing of the old AMD GPUs in the HLT server are significantly greater than for newer GPUs since the AMD GPUs lack the support for the OpenCL generic address space required by several processing steps. The newer NVIDIA GTX1080 GPU model supports offloading of a larger fraction of the workload and is faster in general, replacing up to 40 CPU cores of the Intel Skylake (i7 6700K) CPU, or up to 800 Xeon 2697 CPU cores when compared to offline tracking. Overall, in terms of execution time, a comparable performance is observed for the currently available AMD and NVIDIA GPUs. It has to be noted that HyperThreading was disabled for the measurements of Figs. 13 and 14. With HyperThreading, the Intel Core i7 CPU's total event throughput was 18% higher. The GPU throughput can also be increased by processing multiple independent events in parallel. A throughput increase of 32% is measured, at the expense of some latency on the AMD S9000 [43]. For Fig. 14, the better GPU performance would also require more CPU cores for pre- and post-processing, such that these speedups basically cancel each other out after the normalization to a CPU core. The tracking algorithm has proven to be fast enough for the LHC Run 3, in which ALICE will process time frames of up to 5 overlapping heavy-ion events in one TPC drift time.

GPU models used in the HLT farms of both Run 1 and Run 2 offered a tracking performance equivalent to a large fraction of the CPU cores on an HLT node. Thus, by equipping the servers with GPUs the required size of the farm was nearly reduced by a half. The cost savings compared to tracking on the processors in a traditional farm was around half a million CHF for Run 1 and is above one million CHF for Run 2, not including the savings accrued by having a smaller network, less infrastructure, and lower power consumption. If the HLT only used CPUs, online track reconstruction of all events, using the HLT algorithm, would be prohibitively expensive. Running the offline track reconstruction online would accordingly be even more expensive. This shows that fast tracking algorithms that exploit the capabilities of hardware accelerators are mandatory for future high luminosity heavy-ion experiments like ALICE in the LHC Run 3 or at the experiments that will be setup at the Facility for Antiproton and Ion Research (FAIR) at GSI [48].

The tracking efficiencies, in terms of the fraction of simulated tracks reconstructed by offline and HLT algorithms, are shown in Fig. 15. These efficiencies calculated using a HIJING [49] simulation of Pb–Pb collision events at $\sqrt{s_{\mathrm{NN}}} = 5.02$. The figure distinguishes between primary and secondary tracks as well findable tracks. Findable tracks are reconstructed tracks that have at least 70 clusters in the TPC, and both offline and HLT algorithms achieve close to 100% efficiency for findable primaries. In comparison, when the track sample includes tracks which are not physically in the detector acceptance or tracks with very few TPC hits the efficiency is lower. The minimum transverse momentum measurable for primaries reaches down to 90 MeV/c, as tracks with lower p_{T} do not reach the TPC. The HLT tracker achieves a slightly higher efficiency for secondary tracks because of the usage of the cellular automaton seeding without vertex constraint. In preparation for Run 3, the HLT tracking has also been tuned for the low-p_{T} finding efficiency in order to improve looper-track identification required for the O^2 compression [21]. Both offline and HLT trackers have negligible fake rates, while HLT shows a slightly lower clone rate at high-p_{T}, which is due to the approach used for sector tracking and merging. The clone rate increases significantly for low-p_{T} secondaries, in particular for the HLT. This is not a deficit of the tracker but rather is caused by looping tracks inside the TPC for which the merging of the multiple legs of the loop is not yet implemented.

The track resolution with respect to the track parameters of the MC track taken at the entrance of the TPC is shown in Fig. 16. These track parameters include the y and z spatial positions in the local coordinate system (see Fig. 7), the transverse momentum (p_{T}), the azimuthal (ϕ) and dip (λ) angles. The HLT tracker shows only a nearly negligible degradation compared to the offline algorithm. In order to provide a fair comparison of the tracking algorithms independent from calibration, the offline calibration was used in both cases. This guarantees the exact same transformation of TPC clusters from pad, row, and time to spatial coordinates and the same parameterization of systematic cluster errors due to distortions in the TPC that result from an accumulation of space charge at high interaction rates. Even though the calibration is the same, offline performs some additional corrections to account for the space-charge distortions, e. g. a correction of the covariance matrix that takes the correlation of systematic measurement errors in locally distorted regions into account. The mean values of the distributions obtained from the HLT and offline trackers are identical and the trackers do not show a significant bias for either of the track parameters. The remaining differences in the resolution originate from TPC space-charge distortions, since this correction is not yet implemented in the HLT tracker. This was verified by using MC simulations without the space-charge distortions, where differences in the resolution distribution mostly disappeared.

Overall, the HLT track reconstruction performance is comparable with offline track reconstruction. Speeding up the computation by an order of magnitude introduces only a minor degradation of the track resolution compared to offline. A comparison of efficiency and resolution of GPU and CPU version of the HLT tracking yields identical results. However, the bit-level CPU and GPU results are not 100% comparable because of different floating point rounding and concurrent processing.

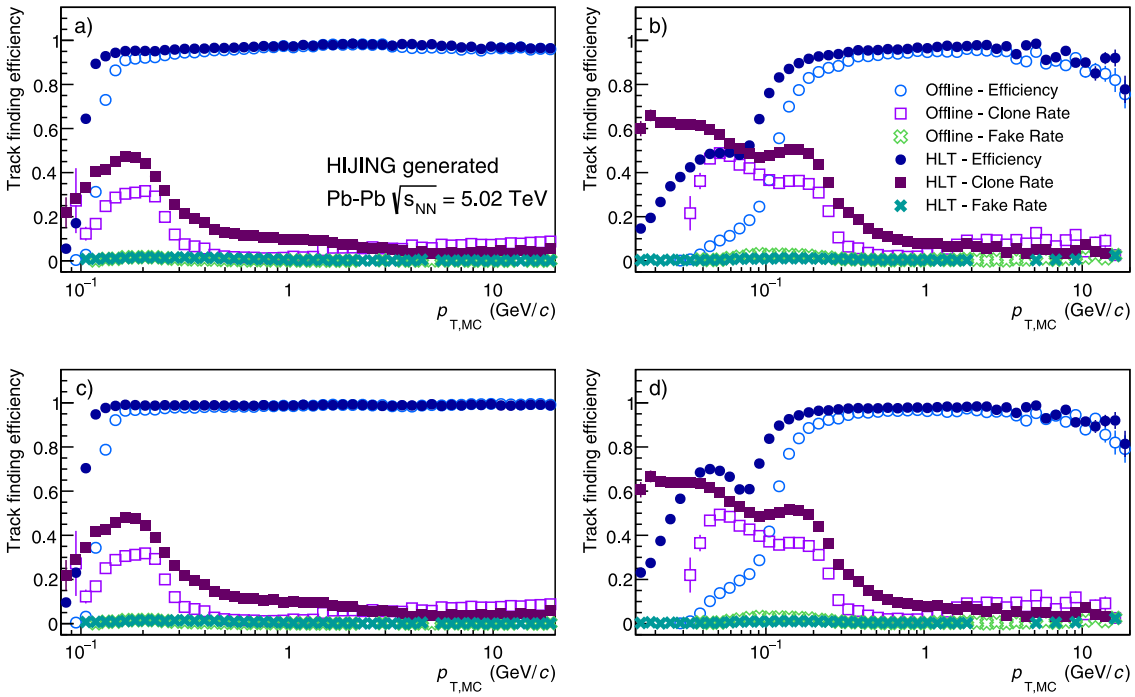


Fig. 15. Tracking efficiency of the HLT and offline trackers as function of the transverse momentum calculated as the ratio of reconstructed tracks and simulated tracks in HIJING generated Pb–Pb events at $\sqrt{s_{NN}} = 5.02$ TeV, shown for tracks that are (a) primary, (b) secondary, (c) findable primary, and (d) findable secondary. Findable tracks are defined as reconstructed tracks that have at least 70 clusters in the TPC.

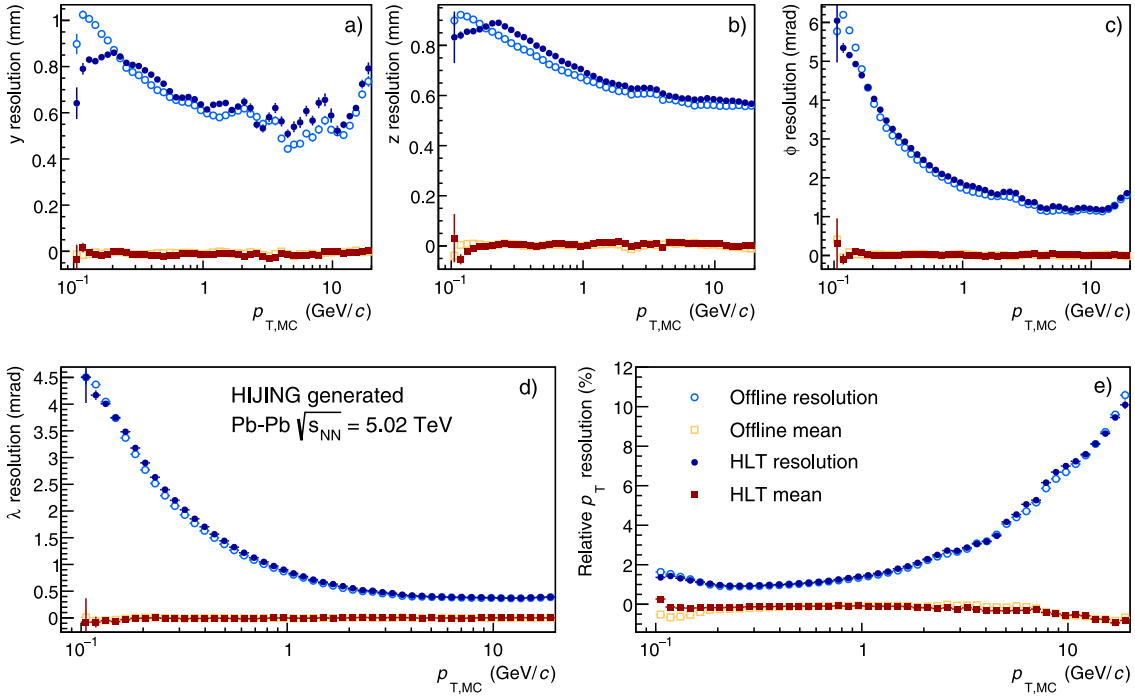


Fig. 16. Mean value and track parameter resolutions of the HLT and offline trackers as function of the transverse momentum measured in HIJING generated Pb–Pb events at $\sqrt{s_{NN}} = 5.02$ TeV. The resolution of (a) y and (b) z spatial positions, (c) azimuthal angle (ϕ), (d) lambda (λ), and (e) relative transverse momentum are shown.

3.4. TPC online calibration

High quality online tracking demands proper calibration objects. Drift detectors, like the TPC, are sensitive to changes in the environmental conditions such as the ambient pressure and/or temperature. Therefore, precise calibration of the electron drift velocity is crucial in order to properly relate the measured arrival time to the TPC end-caps spatial positions along the z axis. Spatial and temporal variations of the properties of the gas inside the TPC as well as the geometrical misalignment of the TPC and ITS contribute to misalignment of individual track segments belonging to a single particle. Corrections for these effects are found by comparing independently fitted TPC track parameters with those found in the ITS [50]. For the online calibration, the cycle

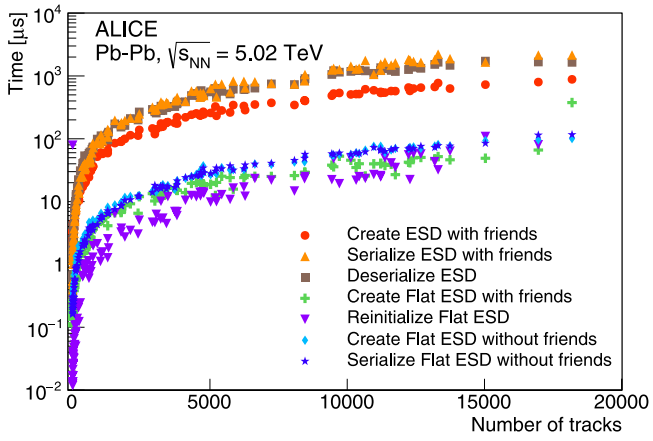


Fig. 17. Time required for the creation, serialization, and deserialization of the Flat ESD vs. the standard ESD for offline analysis as a function of the number of TPC tracks.

starts by collecting data from processing components, which run in parallel on all the HLT nodes. When the desired amount of events (roughly 3000 Pb–Pb events) is obtained, the resulting calibration parameters are merged and processed. To account for their time dependence, the procedure is repeated periodically. At the beginning of the run, no valid online calibration exists. Therefore, the HLT starts the track reconstruction with a default calibration until the online calibration becomes available after the first cycle.

The offline TPC drift-velocity calibration is implemented within the ALICE analysis framework, which is optimized for the processing of ESDs. In addition, the calibration algorithm produces a ROOT object called ESD friend, which contains additional track information and cluster data. Since it is relatively large, the ESD friend is not created for each event, rather it is stored for the events that are used for the calibration. Within the HLT framework the data are transferred between components via contiguous buffers. Hence these ESD objects must be serialized before sending and deserialized after receiving a buffer. Since this flow, comparable to online reconstruction, is resource-hungry a custom data representation was developed, called Flat ESD. Although the Flat ESD shares the same virtual interface with the ESD, the underlying data store of the flat structure is a single contiguous buffer. By design it has zero serialization/deserialization overhead. There is only a negligible overhead related to the virtual function table pointer restoration. Overall, creation, serialization, and deserialization of the Flat ESD is more than 10 times faster compared to the standard ESD used in offline analysis, as demonstrated in Fig. 17.

The HLT provides a wrapper to execute offline code inside the HLT online processing framework using offline configuration macros. The calibration components on each compute node process the calibration tasks asynchronously with respect to the main in-chain data flow. Once sufficient calibration data are collected, the components send their output to an asynchronous data merger. The merged calibration objects are then sent to a single asynchronous process which calculates the cluster transformation maps. These maps are used to correct the cluster position before the track finder algorithm is executed in order to avoid to interfere with the main HLT chain. Finally, finishing the cycle, these maps are distributed back to the beginning of the chain where they are used in the online reconstruction. The cycle is illustrated in Fig. 6. The calibration objects from a cycle are used until the following cycle finished and the output is available. The asynchronous transport uses ZeroMQ.

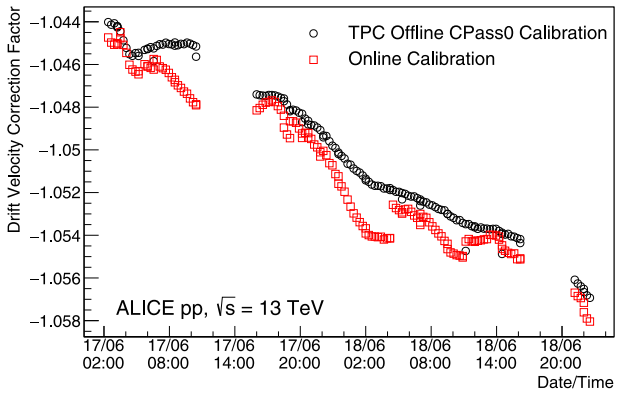
Depending on the availability of computing resources, which rely on beam and trigger conditions, the HLT runs up to 3 calibration worker processes per node on its 172 compute nodes. Events for the calibration are processed distributedly by the 3×172 instances of the calibration task. This number is a parameter: more instances would need more compute resources but in turn would yield more data for calibration in a shorter amount of time. A sufficient calibration precision requires approximately 3000 Pb–Pb event, which are collected in roughly 2 min with the number of instances mentioned above. The subsequent merging of the data, transformation map calculation and distribution to all reconstruction processes takes about another 30 s. While the TPC drift time calibration is stable within a 15 min time window, the total calibration cycle time never exceeds this stable calibration time window.

One difference between online and offline calibration is the availability of real-time ambient pressure and temperature values. Currently, the HLT only has access to the pressure value at the beginning of the run, and does not have access to the temperature at all. In contrast, offline has the full pressure and temperature data over time. This yields two effects shown in Fig. 18(a), in which the drift-velocity correction factor is reported as a function of time. First, the drift-velocity correction factor at the beginning of each run is shifted relative to the offline calibration, since the HLT calibration process uses an outdated temperature compared to offline. Second, during the run the change of the pressure slightly affects the drift velocity. In the offline case, this is accounted for, while HLT sticks to the pressure value at the beginning of the run. For the spatial TPC cluster positions, it does not play a role whether the temperature change is accounted for by the base drift velocity estimation or by the correction factor. Fig. 18(b) shows a run in which the HLT uses the correct temperature at the beginning of the run, obtaining the exact same calibration as offline. It should be noted that in the calibration procedure for Run 3, the temperature value will be available at the beginning of each run.

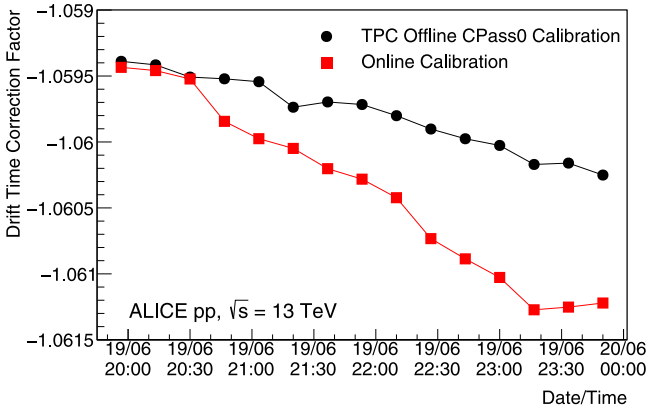
The drift-velocity calibration factors are in agreement for 90% of the runs. The remaining 10% of the runs are primarily composed of short data taking runs, where there was not enough time to gather enough data for online calibration, or test runs in special conditions that prevented a TPC calibration. Without online calibration, the TPC cluster position along the z-axis in the online reconstruction deviate by up to 3 cm from the calibration position available offline. The online calibration reduces this deviation down to 0.5 mm, which is in the order of the intrinsic TPC space point resolution, see Fig. 19. Online calibration objects can be used offline, but since the persistent data are not modified, calibration procedures can still run offline if needed.

3.5. TPC data compression

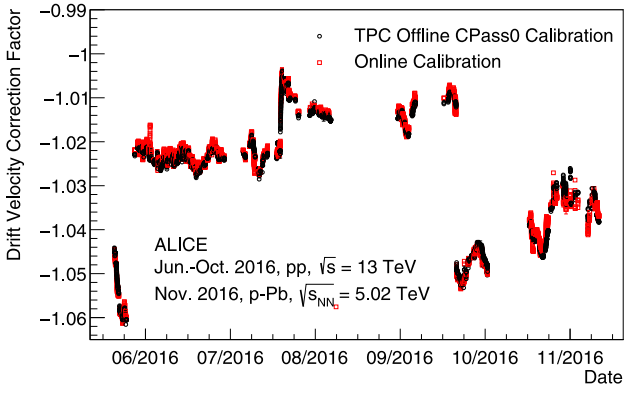
In parallel with the tracking and calibration, the data compression branch of the HLT chain compresses the TPC clusters and replaces the TPC raw data with these compressed clusters [51]. The backbone of the data compression is Huffman entropy encoding [52]. Entropy encoding of the pure TPC ADC values achieves only a maximum compression factor of two, which is less than the compression achievable on the cluster level. The data size is reduced in three consecutive steps. It begins with the hardware cluster finder converting raw data into TPC clusters, calculating properties like total charge, width, and coordinates. The second step converts the computed floating point properties into fixed point integers with the smallest unit equaling the detector resolution. Finally, Huffman encoding compresses the fixed size properties. During Run 1, the average total compression factor was 4.3. In preparation for Run 2 compression techniques were



(a) TPC drift-time correction factor for a period of 12 hours measured during pp data taking at $\sqrt{s}$ = 13 TeV.



(b) TPC drift-time correction factor for a duration of one run (4 hours) measured during pp data taking at $\sqrt{s}$ = 13 TeV. The correct temperature stored at the beginning of the run was included manually.



(c) TPC drift-time correction factor for a period of 5 months measure during pp data taking at $\sqrt{s}$ = 13 TeV and p–Pb data taking at $\sqrt{s_{\mathrm{NN}}}$ = 5.02 TeV.

Fig. 18. TPC drift-time correction factor obtained by the online and offline calibrations as a function of time for various periods of time. Gaps in the distributions correspond to periods without beam.

improved upon. Fig. 20 shows the compression ratio versus the input data size expressed in terms of the number of TPC clusters in 2017, when an average compression factor of 7.3 was achieved.

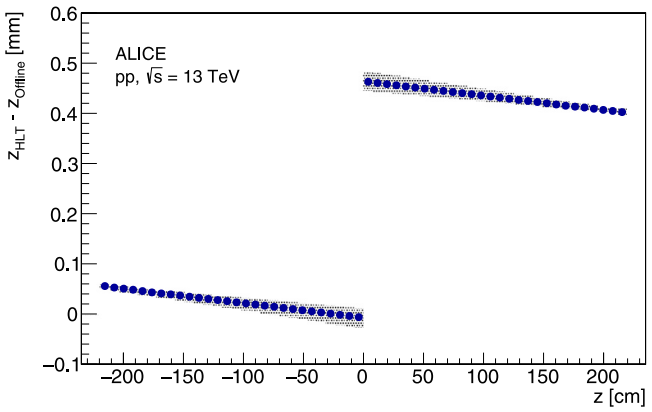


Fig. 19. Average differences of the TPC cluster position along z-axis calculated with drift-velocity correction factors from the online (HLT) and offline calibration. The differences are of the order of the intrinsic detector resolution. Calibration of the forward and backward halves of the TPC are computed independently. The error bands represent the statistical error, along with the r and φ dependent differences of online and offline calibration.

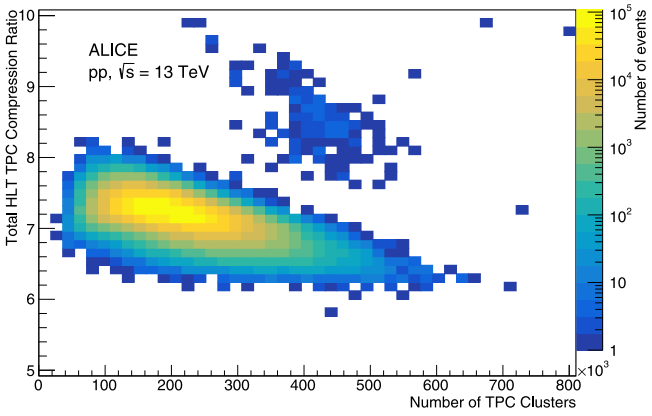


Fig. 20. Total HLT TPC data compression ratio including improved TPC online cluster finder and Huffman compression in Run 2 on 2017 pp data as a function of the input data size expressed in terms of the number of TPC clusters.

Table 2 gives an overview of the improvements of the HLT performance on the compression factors for different data-taking scenarios. The baseline is the compression ratio of 4.3 achieved during Run 1, shown in the leftmost column. In this case, the cluster finding and merging of clusters at readout branch borders yielded a compression factor of 1.2. Storing the cluster information in fixed point integer format reduced the size by a factor of 2.5, requiring 77 bits per cluster thereafter. The entropy coding using Huffman compression reduced the average number of bits per cluster down to 56.6.

Several boundary conditions changed at the beginning of Run 2. The TPC gas was changed from neon to argon in 2015 and 2016 which led to a higher gain. This increased the noise over the zero-suppression threshold, which led to a larger raw data size and an increase of the fake clusters. The compression factor of the cluster finder itself increases, because the fraction of noise in the raw data that is rejected is larger than that in Run 1. In addition, the readout hardware was changed to the RCU2 and the C-RORC (see Section 2.3), allowing all incoming data of one TPC pad-row to be processed together. Before, the pad-row was split into two branches which were processed independently and thus required a successive branch merging step to treat the clusters at the branch borders correctly. This is now obsolete with the new hardware, leading to a better physics performance and higher compression during the cluster finding stage.

Table 2
Compression factors of the different processing steps for the TPC.

Configuration							
Data taking period	2013	2015 Pb–Pb	2016 pp	2017 pp	2017 pp	*2016 pp*	*2015 Pb–Pb*
TPC gas	neon	argon	argon	neon	neon	*argon*	*argon*
RCU version	1	1	2	2	2	*2*	*2*
Cluster finder version	run 1	old	old	old	improved	*improved*	*improved*
Compression version	run 1/2	run 1/2	run 1/2	run 1/2	run 1/2	*run 3 prototype*	*run 3 prototype*
Compression step							
Cluster finder	1.20x	1.28x	1.50x	1.42x	1.81x	*1.72x*	*1.70x*
Branch merging	1.05x	1.05x	–	–	–	–	–
Integer format	2.50x	2.50x	2.50x	2.50x	2.40x	*2.40x*	*2.40x*
(bits per cluster)	77 bits	77 bits	77 bits	77 bits	80 bits	*80 bits*	*80 bits*
Entropy reduction							
(savings after entropy encoding)							
Position differences	–	16%/−7.2 bits	2%/−1.2 bits	2%/−1.0 bits	2%/−1.0 bits	*−1.0 bits*	*−4.5 bits*
Track model	–	–	–	–	–	*−14.5 bits*	*−14.3 bits*
Track model + differences	–	–	–	–	–	*−8.0 bits*	*−8.41 bits*
Logarithmic precision	–	–	–	–	15%/−6.6 bits	*−7.3 bits*	*−7.3 bits*
Entropy encoding							
Huffman coding	1.36x	1.75x	1.49x	1.46x	1.68x	*2.08x*	*2.12x*
Arithmetic coding	–	–	–	–	–	*2.18x*	*2.22x*
Total compression	4.26x	5.89x	5.58x	5.18x	7.28x	*9.00x*	*9.10x*
(bits per cluster)	56.6 bits	44.0 bits	51.7 bits	52.8 bits	47,7 bits	*36,7 bits*	*36,0 bits*

Additional processing steps can reduce the cluster entropy and improve the entropy encoding. In particular, for high occupancy Pb–Pb events, the spatial distribution of the clusters is mostly uniform, but the distances between adjacent clusters are small. Storing position differences instead of absolute positions reduces the entropy and yields a higher compression factor. On average this saves 7.2 bits for Pb–Pb data, reducing the size by 16%.

This is less efficient for pp data in which the occupancy is lower resulting in the position differences being much larger, leading to an average size reduction of only 1.2 bits. Overall, this and other format optimizations have improved the compression factors to 5.5 for pp and to 5.9 for Pb–Pb for Run 2. For clusters associated to tracks, an alternative approach consists of storing the track properties and the residuals of the cluster-to-track position are stored [51,53], listed as "Track model" in Table 2. These residuals also have a small entropy and are ideally suited for Huffman compression. This is particularly useful for pp data, where the position differences method does not perform well.

The two rightmost columns of Table 2 show compression factors obtained by a proof-of-concept prototype for the compression developed for Run 3, using data from Run 2. The prototype includes an advanced version of the track model compression [54] , which refits the track in a distorted coordinate system, yielding significantly smaller residuals than first track model compression during Run 1 [17]. The track-model compression saves on average more than 14 bits per cluster, both for pp and Pb–Pb data. In turn, it deteriorates the compression of the position differences method that is used for clusters not assigned to tracks because the occupancy of non-assigned clusters decreases, increasing the entropy of the differences. The "Track model + differences" row of the table shows the total average savings for all clusters, calculated as the weighted average of the savings achieved by the track-model and the position-differences methods. For Pb–Pb data, the result of 8.41 bits is only slightly better than the pure position differences method. However, the compression factor of pp data reaches the one of Pb–Pb data. There is an even more important benefit of track model compression. It maintains the cluster to track association of HLT tracks for the offline analysis without requiring additional storage or a special data format. In this way, the tracks that were found online by the HLT can be immediately used as seeds by the offline tracker. Having access to the cluster association, the offline tracker can run the slower but more sophisticated routines on the tracks. This approach saves memory and compute cycles during the offline track reconstruction and is currently being commissioned. The Run 3 prototype also shows that, by using arithmetic compression instead of Huffman compression to obtain optimal entropy encoding, a savings of roughly 5% is achieved.

The fixed point integer format is not ideal for all cluster properties. For the cluster width and charge, only a certain relative precision but no absolute precision is needed. Therefore, only a certain number of precision bits after the leading non-zero bit are allowed, and all following less significant bits are forced to zero, implementing proper rounding. This practically emulates a floating point format, while the entropy compression already guarantees the best storage, optimizing away the invalid values with more non-zero bits. By using only three non-significant bits for the cluster width and four for the charge, a savings of 15% of the 2017 pp data volume was obtained.

With the argon gas being used in the TPC at the beginning of Run 2, a significant overhead of fake clusters emerging from the increased noise was faced. The cluster finder searches for charge peaks and merges them, creating fake clusters if the total adjacent noise exceeds a minimum threshold. Therefore, the HLT cluster finder was improved for the 2017 data taking to reject this noise by an improved peak finding heuristic. This improved hardware cluster finder (see Section 3.2) reduced the amount of clusters reconstructed in the TPC in pp data collected in 2016 when argon was the TPC gas by 32%. The reduction was approximately 21% for the pp data collected in 2016 when the TPC gas was neon. It also sped up the tracking and yielded slightly better track parameters. Note that the gain in compression after the Huffman encoding can differ between the two data sets because noise clusters have different entropy.

Storage space is a limiting factor in data taking, even with the inclusion of HLT compression. Currently ALICE uses almost the entire allocated capacity, which is roughly 10 PB per year. TPC data are by far the largest contributor taking up more than 90% of the raw data volume. The offline software employs built-in ROOT file compression on the raw data from the other detectors. Their relative contribution increases significantly after the more than five-fold compression of the TPC data by the HLT in 2016. Overall, the HLT compression increases the total number of events ALICE can record and store by more than a factor of 4 within the given storage budget. In the case of Pb–Pb data taking, also the raw data bandwidth would exceed the available capacity necessitating the real-time compression in the HLT. Aggregating all compression

steps of the Run 3 prototype, a total compression factor of 9 was achieved for both pp and for Pb–Pb data. In the future, additional compression steps are foreseen, like rejecting TPC clusters attached to tracks with transverse momenta below 50 MeV/c, clusters attached to additional legs of looping tracks, and clusters attached to track segments with large inclination angles, which are not used in physics analyses. Using this cluster rejection, an additional compression factor of 2 is expected, bringing the compression factor close to the foreseen factor of 20, which is necessary for the O^2 computing upgrade.

3.6. Quality assurance for TPC, EMCAL, and other detectors

The HLT, in addition to online reconstruction and compression, also runs various types of QA and physics analysis components that allow for real-time monitoring of the physics performance of the ALICE apparatus. These frameworks gather and process various types of information: from event, track and vertex properties to data compression parameters. The HLT components executing these frameworks can be classified as fast, slow, and/or asynchronous. The fast components (e.g. EMCal and HLT's own QA) require the full data sample and therefore are considered prompt components, running in-chain. Slow components that simply sample some of the reconstructed events are executed out-of-chain, subscribing to the main chain transiently on a dedicated monitoring node, processing events on a best effort basis. Finally, some QA components run asynchronously on all nodes using the wrapper for the ALICE physics analysis task framework, which was developed for and is also used in online calibration [55] (compare Section 3.4).

In the asynchronous mode, the full statistics (or a subset proportional to the dedicated processing capacity) can be processed without disrupting the standard HLT operations. Several tasks from the TPC team are now running within the HLT in this mode. Another component that runs out-of-chain is the luminous region component, which provides information on the size and position of the region of the particle beams to the LHC team. In this case, all event information of interest are processed synchronously, with the merging and fitting stages being performed out-of-chain. The LHC is updated with these data in 30 s intervals.

In addition to running asynchronously the HLT also performs online monitoring synchronously. This allows access to the full data sample, however the components must be very stable to not interfere with HLT operations. With this infrastructure histograms can be created and modified on-the-fly to allow for e.g. prompt studies of trigger selections per histogram. This infrastructure also supports the correlation of arbitrary quantities like V0/T0/ZDC detector signals versus the number of ITS/TPC tracks. The benefit is that all events can be processed by running synchronously at the full event rate. The histograms produced by the in-chain components are continuously merged (asynchronously) and can be accessed at any time during the run. The detector teams of TPC and EMCal have also implemented similar QA tasks, which run in an analogous manner. The final monitoring histograms, run- and time-dependent, are published online for simple access. Furthermore, the data for HLT monitoring are available on the data quality monitoring station utilized by the ALICE shift crew.

Data exchange between asynchronous components and a part of external communication (e.g. related to QA) is handled by the ZeroMQ messaging library. The processing components need to exchange a multitude of data types related to a single entity (e.g. a triggered event). Data originates from different sources, e.g. shared memory holding raw or reconstructed data processed synchronously, buffers provided by serialization libraries and schema evolution data (e.g. ROOT streamers). Each data buffer needs to be uniquely identified to allow for correct decoding on the receiving end: metadata that annotates the contents and serialization strategy of a data buffer are constructed separately and sent together with the data. The association of metadata to data buffers is maintained at the transport layer level by ordering the header–payload pairs in a sequence. The ZeroMQ multi-part functionality allows for the atomic transport of multiple buffers and buffer ordering preservation; it is wrapped by a thin abstraction layer to provide an easy to use vectored input/output-like interface. In this scheme, many annotated data parts can efficiently be added to a single ZeroMQ message without the overhead being typically associated to message (de-)serialization [36].

4. Performance analysis of global HLT operation

In addition to system stability, the HLT must ensure that during normal operations any throttling of the data taking of the experiment is avoided. When one of the HLT processing components is too slow to process the incoming data, for example when the network cannot manage the data rate or when the framework cannot schedule the events, the HLT internal buffers become full and this results in the HLT sending back-pressure to the experiment and pausing data taking until there is again buffer space to accept more data.

Data rate and event rate are, although related, two different factors. For instance, small events at very high rate cause excessive load on the scheduling of related interprocess communication while the utilized network bandwidth can still be small. On the other hand, a few large events can saturate the network.

In 2016, the HLT caused on average less than 100 μs of back-pressure per run, an insignificant amount compared to the usual run duration of several hours. Therefore, the HLT has a negligible effect on the data taking efficiency. Besides observations during the operation, extensive data-replay based measurements were conducted to ensure that the HLT manages to process all data and event rates for all the foreseen data taking and trigger scenarios.

Data replay (see Section 2.3) allows for the evaluation of the HLT performance under a certain load scenario given the exact same conditions as in normal operation. The maximum input data rate into the HLT is limited by the number and the link speed of the optical link fibers coming from the detectors. The dominant contribution is from the TPC with 216 links, each running at 3.125 GBit/s. However, not all links can send data at full rate simultaneously. This is due to the geometry of the TPC resulting in the number of channels sent per link not being constant. Considering also the link protocol overhead, the maximum possible input rate from the TPC is 48 GB/s. Note that, during real operation, the TPC pauses the readout during sampling and that the TPC gating grid and detector busy time reduce the maximum rate. Therefore, the real rate is below 40 GB/s at less than 2 kHz, which gives some additional margin. Additionally 10 GB/s can originate from the other detectors. In the following, the data replay is analyzed using two data sets: pp events at high luminosity and maximum pile-up as well as minimum bias Pb–Pb events. Considering the data size, the replay of the Pb–Pb data set was run at 950 Hz and the pp data set at 2.5 kHz, which correspond to a TPC input rate of 48 GB/s in both cases. Other detectors can operate at higher event rates than the TPC. The ALICE trigger scenarios for Run 2 foresaw a rate below 2 kHz for the central barrel detectors with an additional few hundred Hz from both the fast-interaction detectors and the muon detector. This results in a total maximum aggregate event rate below 3.5 kHz when all trigger clusters are at maximum rate at the same time. A mixture of additional events without TPC contribution, to obtain a higher event rate for other detectors, was added to the replay data set.

Table 3 gives an overview of the maximum rate handled by the HLT for various scenarios. The HLT framework imposes an

Table 3
Maximum data rates and event rates in the HLT for different load scenarios in data replay.

Scenario	Detectors	Input size	TPC rate	Total event rate	Limiting factor
Single input link	ZDC	6 MB/s	0	10 kHz	Framework
pp 5.02 TeV	TPC, ITS, EMCAL, V0	8.3 GB/s	4.5 kHz	4.5 kHz	CPU load
pp 13 TeV	TPC, ITS, EMCAL, V0	48 GB/s	2.4 kHz	2.4 kHz	Optical link bandwidth
Pb–Pb 5.02 TeV	TPC, ITS, EMCAL, V0, ZDC	48 GB/s	950 Hz	950 Hz	Optical link bandwidth
Pb–Pb 5.02 TeV	ITS, EMCAL, V0, ZDC	3.5 GB/s	0	6 kHz	Framework
pp 13 TeV	All	49 GB/s	2.4 kHz	6 kHz	Optical link bandwidth/Framework
Pb–Pb 5.02 TeV	All	51 GB/s	950 Hz	3.75 kHz	Optical link bandwidth/CPU

event-rate limit of 10 kHz for a front-end node with a single input link, and a limit of 6 kHz for event-merging of the twelve links of a fully connected C-RORC [56]. Both limits do not apply in practice because the fastest foreseen trigger scenario peaks at an aggregate rate of 3.5 kHz. The table also shows that the CPU capacity will only become critical at event rates not supported by the detectors. The current GPU-based tracking achieves a peak TPC processing rate of 2.4 kHz for Pb–Pb data with an 8 kHz interaction rate, if it runs locally and standalone using all compute nodes. Currently, this leaves a 50% margin on the GPU capacity, which can be used for the implementation of additional compute-intensive online reconstruction steps.

The experience during Run 1 demonstrates that the performance of the event merging task is critical for the maximum achievable rate. Consequently, both the interprocess communication in the framework (see Section 2.6) as well as the HLT configuration was improved by having a more balanced layout, thereby reducing the load on the event fragment merger. These changes improved the maximum rates from 3 to 6 kHz for pp and from 500 to 950 Hz for Pb–Pb collisions.

In addition to the input and processing capacity, the HLT must have sufficient bandwidth of the internal network and of the output links to DAQ. The above scenarios lead to a maximum outgoing network bandwidth of 1.38 GB/s per input node, and for the current TPC data compression factor, a maximum of 1.53 GB/s received per output node. In total, 10.7 GB/s are sent to DAQ. Using the current HLT chain without processing, the framework has been tested up to input and output rates of 2.4 GB/s per node. This leaves already a margin of more than 50%, and close to 6 GB/s of network bandwidth per node accessible through the use of multiple transport streams. The optical link rates to DAQ and data storage systems have been tested to a maximum aggregate transfer speed of 12 GB/s, which is above the upper bound for the output rate of 10.7 GB/s. For the current HLT chain, the limiting factor is the actual link speed of the 28 fibers to DAQ, which are running at the highest possible speed of 5.3 GBit. The output rate could be increased by using more physical fibers, for which the HLT already has spare ports available. As presented in Section 2.3, the HLT C-RORC and the PCI Express bus in the FEP nodes are also able to handle any incoming data from the up to twelve optical links per node.

Overall, the current HLT farm handles all foreseen workloads for Run 2 without imposing backpressure. Looking ahead to Run 3, the available resources will be used to test and prototype the many new features planned for the O^2 system as early as possible under real conditions in the HLT.

4.1. HLT operation stability

The HLT is an integral part of the ALICE data taking chain and its operational stability is critical because a failure would interrupt the data taking. Moreover, without the HLT compression a maximum readout rate is no longer possible due to the bottlenecks described above. In addition, storage space becomes a problem due to the fact that uncompressed raw data quadruples storage requirements.

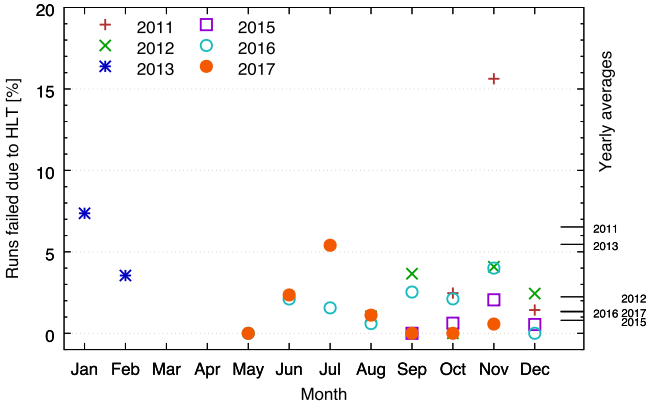


Fig. 21. Number of data taking runs terminated due to failure in the HLT during Run 1 and Run 2 since 2011, when the TPC data compression in the HLT was introduced. Missing months correspond to long shutdowns, end-of-year shutdowns, commissioning phases for the data compression and recommissioning for updated TPC readout. The yearly averages are shown as long tick marks along the right-side *y*-axis.

Fig. 21 shows the stability of the HLT over the period from 2011 to 2017. The figure includes only runs in which ALICE was collecting physics data and with the HLT performing TPC data compression. Overall, only a small percentage of data taking runs ended due to HLT problems. Since October 2011, there were only three months during which the HLT caused more than 5% of the data taking runs to fail. The largest percentage of failures occurred in November 2011. At that time the TPC compression was still in the commissioning phase and for the first time Pb–Pb data were collected at a higher interaction rate with respect to Run 1, which in turn demanded additional fine tuning. On average the end-of-run reasons associated with HLT failures were less than 2%. Compared to the beginning of data taking during Run 1 [57], the fraction of runs failing due to HLT issues was reduced by roughly a factor of 2. The absolute rate, which was above 100 failures per year in Run 1 decreased considerably. In total, only 18 physics runs failed due to an HLT issue in 2016. The causes were: GPU driver problems causing reboots solved by driver update (2 runs), GPU stuck due to driver problems (4 runs), malfunctioning GPU (2 runs), malfunctioning CPU (2 runs), unexpected node reboot (1 run), uncorrectable machine check exception (4 runs), and network communication problems (3 runs).

A significant fraction of failures are due to GPU driver problems, which are still not fully resolved by the vendor. A workaround was implemented that outsources the GPU reconstruction to a different operating system process. If the GPU or the driver get stuck, the HLT chain continues normal operation and skips the track reconstruction for the few events scheduled for that process. This reduces the statistics for online QA and calibration only negligibly.

Despite several failures that happened during the debugging of a reoccurring hardware problem at the beginning of July,

the HLT had a low failure rate in 2017. In the future, additional preemptive measures will be deployed for network, hard disk, and machine check failures, to reduce the failure rate even further.

For a better estimate of how much data taking time is lost due to HLT failures, the total time between the occurrence of the problem and the moment in which data are recorded is again calculated. This includes possible unsuccessful measures to continue data taking without a full restart as well as stop and startup time. This is a more realistic measurement of the dead time, compared to an estimate based on the time between end-of-run and start of the next data taking run. However, this is an upper bound, as it also includes time for unrelated actions performed in between. In this metric, the 18 failures in 2016 interrupted the data taking for 11 233 s in total out of 1409 h of data taking with HLT in its full configuration. This means that the HLT failures amounted to less than 0.22% of the available data taking time.

5. Outlook

The ALICE HLT has been operational since November 2009 with the first pp collisions at LHC at $\sqrt{s} = 0.9$ TeV and has since then processed all subsequent data. Operated with a combination of a fast FPGA hardware cluster finder and GPU tracker the ALICE HLT pioneered the use of hardware accelerator technologies in real-time computing at the LHC.

During the LHC Run 3 ALICE will collect 100 times more data with respect to what was recorded during Run 1 and Run 2. The increase in statistics will be made possible by a tenfold increase of the LHC luminosity as well as the change of the detector readout mode from triggered to continuous, allowing the readout of the full Pb–Pb interaction rate of up to 50 kHz. The data stream has to be compressed by a factor of 20 in order to be transported to the storage element for permanent storage. Achieving this compression ratio requires a paradigm shift in processing: all data will be reconstructed and calibrated online synchronous to the data taking. In addition to compression schemes already discussed in Section 3.5, parts of the data, e.g. clusters positively identified to be disposable, will be discarded making the overall compression scheme lossy. The quality of online reconstruction and calibration will therefore be paramount.

Concepts and technologies which are developed as part of the HLT (described in this paper and in Sections 2.6, 3.3, and 3.5) are being studied, prototyped and tested already now in a production environment, also being adapted and further developed in the software framework of O^2.

Acknowledgments

The ALICE Collaboration would like to thank all its engineers and technicians for their invaluable contributions to the construction of the experiment and the CERN accelerator teams for the outstanding performance of the LHC complex. The ALICE Collaboration gratefully acknowledges the resources and support provided by all Grid centres and the Worldwide LHC Computing Grid (WLCG) collaboration. The ALICE Collaboration acknowledges the following funding agencies for their support in building and running the ALICE detector: A. I. Alikhanyan National Science Laboratory (Yerevan Physics Institute) Foundation (ANSL), State Committee of Science and World Federation of Scientists (WFS), Armenia; Austrian Academy of Sciences and Nationalstiftung für Forschung, Technologie und Entwicklung, Austria; Ministry of Communications and High Technologies, National Nuclear Research Center, Azerbaijan; Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Universidade Federal do Rio Grande do Sul (UFRGS), Financiadora de Estudos e Projetos (Finep) and Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Brazil; Ministry of Science & Technology of China (MSTC), National Natural Science Foundation of China (NSFC) and Ministry of Education of China (MOEC), China; Croatian Science Foundation and Ministry of Science and Education, Croatia; Centro de Aplicaciones Tecnológicas y Desarrollo Nuclear (CEADEN), Cubaenergía, Cuba; Ministry of Education, Youth and Sports of the Czech Republic, Czech Republic; The Danish Council for Independent Research — Natural Sciences, the Carlsberg Foundation and Danish National Research Foundation (DNRF), Denmark; Helsinki Institute of Physics (HIP), Finland; Commissariat à l'Energie Atomique (CEA) and Institut National de Physique Nucléaire et de Physique des Particules (IN2P3) and Centre National de la Recherche Scientifique (CNRS), France; Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) and GSI Helmholtzzentrum für Schwerionenforschung GmbH, Germany; General Secretariat for Research and Technology, Ministry of Education, Research and Religions, Greece; National Research, Development and Innovation Office, Hungary; Department of Atomic Energy Government of India (DAE), Department of Science and Technology, Government of India (DST), University Grants Commission, Government of India (UGC) and Council of Scientific and Industrial Research (CSIR) , India; Indonesian Institute of Science, Indonesia; Centro Fermi — Museo Storico della Fisica e Centro Studi e Ricerche Enrico Fermi and Istituto Nazionale di Fisica Nucleare (INFN), Italy; Institute for Innovative Science and Technology, Nagasaki Institute of Applied Science (IIST), Japan Society for the Promotion of Science (JSPS) KAKENHI and Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan; Consejo Nacional de Ciencia (CONACYT) y Tecnología, through Fondo de Cooperación Internacional en Ciencia y Tecnología (FONCICYT) and Dirección General de Asuntos del Personal Academico (DGAPA), Mexico; Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO), Netherlands; The Research Council of Norway, Norway; Commission on Science and Technology for Sustainable Development in the South (COMSATS), Pakistan; Pontificia Universidad Católica del Perú, Peru; Ministry of Science and Higher Education and National Science Centre, Poland; Korea Institute of Science and Technology Information and National Research Foundation of Korea (NRF), Republic of Korea; Ministry of Education and Scientific Research, Institute of Atomic Physics and Romanian National Agency for Science, Technology and Innovation, Romania; Joint Institute for Nuclear Research (JINR), Ministry of Education and Science of the Russian Federation, National Research Centre Kurchatov Institute, Russian Science Foundation and Russian Foundation for Basic Research, Russia; Ministry of Education, Science, Research and Sport of the Slovak Republic, Slovakia; National Research Foundation of South Africa, South Africa; Swedish Research Council (VR) and Knut & Alice Wallenberg Foundation (KAW), Sweden; European Organization for Nuclear Research, Switzerland; National Science and Technology Development Agency (NSDTA), Suranaree University of Technology (SUT) and Office of the Higher Education Commission under NRU project of Thailand, Thailand; Turkish Atomic Energy Agency (TAEK), Turkey; National Academy of Sciences of Ukraine, Ukraine; Science and Technology Facilities Council (STFC), United Kingdom; National Science Foundation of the United States of America (NSF) and United States Department of Energy, Office of Nuclear Physics (DOE NP), United States of America. We thank AMD and ASUS for their support in commissioning the compute farm.

Appendix. The ALICE collaboration

S. Acharya[140], F.T.-. Acosta[20], D. Adamová[93], S.P. Adhya[140], A. Adler[74], J. Adolfsson[80], M.M. Aggarwal[98], G. Aglieri Rinella[34], M. Agnello[31], Z. Ahammed[140], S. Ahmad[17], S.U. Ahn[76], S. Aiola[145], A. Akindinov[64], M. Al-Turany[104], S.N. Alam[140], D.S.D. Albuquerque[121], D. Aleksandrov[87], B. Alessandro[58], H.M. Alfanda[6], R. Alfaro Molina[72], B. Ali[17], Y. Ali[15], A. Alici[10,53,27], A. Alkin[2], J. Alme[22], T. Alt[69], L. Altenkamper[22], I. Altsybeev[111], M.N. Anaam[6], C. Andrei[47], D. Andreou[34], H.A. Andrews[108], A. Andronic[143,104], M. Angeletti[34], V. Anguelov[102], C. Anson[16], T. Antičić[105], F. Antinori[56], P. Antonioli[53], R. Anwar[125], N. Apadula[79], L. Aphecetche[113], H. Appelshäuser[69], S. Arcelli[27], R. Arnaldi[58], M. Arratia[79], I.C. Arsene[21], M. Arslandok[102], A. Augustinus[34], R. Averbeck[104], M.D. Azmi[17], M. Bach[39], A. Badalà[55], Y.W. Baek[40,60], S. Bagnasco[58], R. Bailhache[69], R. Bala[99], A. Baldisseri[136], M. Ball[42], R.C. Baral[85], R. Barbera[28], L. Barioglio[26], G.G. Barnaföldi[144], L.S. Barnby[92], V. Barret[133], P. Bartalini[6], K. Barth[34], E. Bartsch[69], N. Bastid[133], S. Basu[142], G. Batigne[113], B. Batyunya[75], P.C. Batzing[21], D. Bauri[48], J.L. Bazo Alba[109], I.G. Bearden[88], B. Becker[102], C. Bedda[63], N.K. Behera[60], I. Belikov[135], F. Bellini[34], H. Bello Martinez[44], R. Bellwied[125], L.G.E. Beltran[119], V. Belyaev[91], G. Bencedi[144], S. Beole[26], A. Bercuci[47], Y. Berdnikov[96], D. Berenyi[144], R.A. Bertens[129], D. Berzano[58], L. Betev[34], A. Bhasin[99], I.R. Bhat[99], H. Bhatt[48], B. Bhattacharjee[41], A. Bianchi[26], L. Bianchi[125,26], N. Bianchi[51], J. Bielčík[37], J. Bielčíková[93], A. Bilandzic[103,116], G. Biro[144], R. Biswas[3], S. Biswas[3], J.T. Blair[118], D. Blau[87], C. Blume[69], G. Boca[138], F. Bock[34], S. Boetteger[102], A. Bogdanov[91], L. Boldizsár[144], A. Bolozdynya[91], M. Bombara[38], G. Bonomi[139], M. Bonora[34], H. Borel[136], A. Borissov[143,102], M. Borri[127], E. Botta[26], C. Bourjau[88], L. Bratrud[69], P. Braun-Munzinger[104], M. Bregant[120], T. G. Breitener[102], T.A. Broker[69], M. Broz[37], E.J. Brucken[43], E. Bruna[58], G.E. Bruno[33], M.D. Buckland[127], D. Budnikov[106], H. Buesching[69], S. Bufalino[31], P. Buhler[112], P. Buncic[34], O. Busch[132,i], Z. Buthelezi[73], J.B. Butt[15], J.T. Buxton[95], D. Caffarri[89], H. Caines[145], A. Caliva[104], E. Calvo Villar[109], R.S. Camacho[44], P. Camerini[25], A.A. Capon[112], F. Carnesecchi[10,27], J. Castillo Castellanos[136], A.J. Castro[129], E.A.R. Casula[54], C. Ceballos Sanchez[52], P. Chakraborty[48], S. Chandra[140], B. Chang[126], W. Chang[6], S. Chapeland[34], M. Chartier[127], S. Chattopadhyay[140], S. Chattopadhyay[107], A. Chauvin[24], C. Cheshkov[134], B. Cheynis[134], V. Chibante Barroso[34], D.D. Chinellato[121], S. Cho[60], P. Chochula[34], T. Chowdhury[133], P. Christakoglou[89], C.H. Christensen[88], P. Christiansen[80], T. Chujo[132], C. Cicalo[54], L. Cifarelli[10,27], F. Cindolo[53], J. Cleymans[124], F. Colamaria[52], D. Colella[52], A. Collu[79], M. Colocci[27], M. Concas[58,ii], G. Conesa Balbastre[78], Z. Conesa del Valle[61], G. Contin[127], J.G. Contreras[37], T.M. Cormier[94], Y. Corrales Morales[26,58], P. Cortese[32], M.R. Cosentino[122], F. Costa[34], S. Costanza[138], J. Crkovská[61], P. Crochet[133], E. Cuautle[70], L. Cunqueiro[94], D. Dabrowski[141], T. Dahms[103,116], A. Dainese[56], F.P.A. Damas[113,136], S. Dani[66], M.C. Danisch[102], A. Danu[68], D. Das[107], I. Das[107], S. Das[3], A. Dash[85], S. Dash[48], A. Dashi[103], S. De[85,49], A. De Caro[30], G. de Cataldo[52], C. de Conti[120], J. de Cuveland[39], A. De Falco[24], D. De Gruttola[30,10], N. De Marco[58], S. De Pasquale[30], R.D. De Souza[121], H.F. Degenhardt[120], A. Deisting[104,102], A. Deloff[84], S. Delsanto[26], P. Dhankher[48], D. Di Bari[33], A. Di Mauro[34], R.A. Diaz[8], T. Dietel[124], P. Dillenseger[69], Y. Ding[6], R. Divià[34], O. Djuvsland[22], A. Dobrin[34], D. Domenicis Gimenez[120], B. Dönigus[69], O. Dordic[21], A.K. Dubey[140], A. Dubla[104], S. Dudi[98], A.K. Duggal[98], M. Dukhishyam[85], P. Dupieux[133], R.J. Ehlers[145], D. Elia[52], H. Engel[74], E. Epple[145], B. Erazmus[113], F. Erhardt[97], A. Erokhin[111], M.R. Ersdal[22], B. Espagnon[61], G. Eulisse[34], J. Eum[18], D. Evans[108], S. Evdokimov[90], L. Fabbietti[103,116], M. Faggin[29], J. Faivre[78], A. Fantoni[51], M. Fasel[94], L. Feldkamp[143], A. Feliciello[58], G. Feofilov[111], A. Fernández Téllez[44], A. Ferrero[136], A. Ferretti[26], A. Festanti[34], V.J.G. Feuillard[102], J. Figiel[117], S. Filchagin[106], D. Finogeev[62], F.M. Fionda[22], G. Fiorenza[52], F. Flor[125], M. Floris[34], S. Foertsch[73], P. Foka[104], S. Fokin[87], E. Fragiacomo[59], A. Francisco[113], U. Frankenfeld[104], G.G. Fronze[26], U. Fuchs[34], C. Furget[78], A. Furs[62], M. Fusco Girard[30], J.J. Gaardhøje[88], M. Gagliardi[26], A.M. Gago[109], K. Gajdosova[37,88], C.D. Galvan[119], P. Ganoti[83], C. Garabatos[104], E. Garcia-Solis[11], K. Garg[28], C. Gargiulo[34], K. Garner[143], P. Gasik[103,116], E.F. Gauger[118], M.B. Gay Ducati[71], M. Germain[113], J. Ghosh[107], P. Ghosh[140], S.K. Ghosh[3], P. Gianotti[51], P. Giubellino[104,58], P. Giubilato[29], P. Glässel[102], D.M. Goméz Coral[72], A. Gomez Ramirez[74], V. Gonzalez[104], P. González-Zamora[44], S. Gorbunov[39], L. Görlich[117], S. Gotovac[35], V. Grabski[72], L.K. Graczykowski[141], K.L. Graham[108], L. Greiner[79], A. Grelli[63], C. Grigoras[34], V. Grigoriev[91], A. Grigoryan[1], S. Grigoryan[75], J.M. Gronefeld[104], F. Grosa[31], J.F. Grosse-Oetringhaus[34], R. Grosso[104], R. Guernane[78], B. Guerzoni[27], M. Guittiere[113], K. Gulbrandsen[88], T. Gunji[131], A. Gupta[99], R. Gupta[99], I.B. Guzman[44], R. Haake[145,34], O. S. Haaland[22], M.K. Habib[104], C. Hadjidakis[61], H. Hamagaki[81], G. Hamar[144], M. Hamid[6], J.C. Hamon[135], R. Hannigan[118], M.R. Haque[63], A. Harlenderova[104], J.W. Harris[145], A. Harton[11], H. Hassan[78], D. Hatzifotiadou[53,10], P. Hauer[42], S. Hayashi[131], S.T. Heckel[69], E. Hellbär[69], H. Helstrup[36], A. Herghelegiu[47], E.G. Hernandez[44], G. Herrera Corral[9], F. Herrmann[143], K.F. Hetland[36], T.E. Hilden[43], H. Hillemanns[34], C. Hills[127], B. Hippolyte[135], B. Hohlweger[103], D. Horak[37], S. Hornung[104], R. Hosokawa[132], J. Hota[66], P. Hristov[34], C. Huang[61], C. Hughes[129], P. Huhn[69], T.J. Humanic[95], H. Hushnud[107], L.A. Husova[143], N. Hussain[41], S.A. Hussain[15], T. Hussain[17], D. Hutter[39], D.S. Hwang[19], J.P. Iddon[127], R. Ilkaev[106], M. Inaba[132], M. Ippolitov[87], M.S. Islam[107], M. Ivanov[104], V. Ivanov[96], V. Izucheev[90], B. Jacak[79], N. Jacazio[27], P.M. Jacobs[79], M.B. Jadhav[48], S. Jadlovska[115], J. Jadlovsky[115], S. Jaelani[63], C. Jahnke[120], M.J. Jakubowska[141], M.A. Janik[141], M. Jercic[97], O. Jevons[108], R.T. Jimenez Bustamante[104], M. Jin[125], P.G. Jones[108], A. Jusko[108], S. Kalcher[39], P. Kalinak[65], A. Kalweit[34], K. Kanaki[22], J.H. Kang[146], V. Kaplin[91], S. Kar[6], A. Karasu Uysal[77], O. Karavichev[62], T. Karavicheva[62], P. Karczmarczyk[34], E. Karpechev[62], U. Kebschull[74], R. Keidel[46], M. Keil[34], B. Ketzer[42], Z. Khabanova[89], A.M. Khan[6], S. Khan[17], S.A. Khan[140], A. Khanzadeev[96], Y. Kharlov[90], A. Khatun[17], A. Khuntia[49], M.M. Kielbowicz[117], B. Kileng[36], B. Kim[60], B. Kim[132], D. Kim[146], D.J. Kim[126], E.J. Kim[13], H. Kim[146], J.S. Kim[40], J. Kim[102], J. Kim[146], J. Kim[13], M. Kim[102,60], S. Kim[19], T. Kim[146], T. Kim[146], K. Kindra[98], S. Kirsch[39], I. Kisel[39], S. Kiselev[64], A. Kisiel[141], J.L. Klay[5], C. Klein[69], J. Klein[58], S. Klein[79], C. Klein-Bösing[143], S. Klewin[102], A. Kluge[34], M.L. Knichel[34], A.G. Knospe[125], C. Kobdaj[114], M. Kofarago[144], M.K. Köhler[102], T. Kollegger[104], A. Kondratyev[75], N. Kondratyeva[91], E. Kondratyuk[90], P.J. Konopka[34], M. Konyushikhin[142], L. Koska[115], O. Kovalenko[84], V. Kovalenko[111], M. Kowalski[117], I. Králik[65], A. Kravčáková[38], L. Kreis[104], M. Krivda[65,108], F. Krizek[93], M. Krüger[69], E. Kryshen[96], M. Krzewicki[39], A.M. Kubera[95], V. Kučera[60,93], C. Kuhn[135], P.G. Kuijer[89], L. Kumar[98], S. Kumar[48], S. Kundu[85], P. Kurashvili[84], A. Kurepin[62], A.B. Kurepin[62], S. Kushpil[93], J. Kvapil[108], M.J. Kweon[60], Y. Kwon[146], S.L. La Pointe[39], P. La Rocca[28], Y.S. Lai[79], R. Langoy[123], K. Lapidus[34,145], C. E. Lara Martinez[102], A. Lardeux[21], P. Larionov[51], E. Laudi[34], R. Lavicka[37], T. Lazareva[111], R. Lea[25], L. Leardini[102], S. Lee[146], F. Lehas[89], S. Lehner[112], J. Lehrbach[39], R.C. Lemmon[92], I. León Monzón[119], P. Lévai[144], X. Li[12], X.L. Li[6], J. Lien[123], R. Lietava[108], B. Lim[18], S. Lindal[21], V. Lindenstruth[39], S.W. Lindsay[127], C. Lippmann[104], M.A. Lisa[95], V. Litichevskyi[43], A. Liu[79], H.M. Ljunggren[80], W.J. Llope[142], D.F. Lodato[63], V. Loginov[91], C. Loizides[94], P. Loncar[35], X. Lopez[133], E. López Torres[8], P. Luettig[69], J.R. Luhder[143], M. Lunardon[29], G. Luparello[59], M. Lupi[34], A. Maevskaya[62], M. Mager[34], S.M. Mahmood[21], T. Mahmoud[42], A. Maire[135], R.D. Majka[145], M. Malaev[96], Q.W. Malik[21], L. Malinina[75,iii], D. Mal'Kevich[64], P. Malzacher[104], A. Mamonov[106], V. Manko[87], F. Manso[133], V. Manzari[52], Y. Mao[6], M. Marchisone[134],

J. Mareš[67], G.V. Margagliotti[25], A. Margotti[53], J. Margutti[63], A. Marín[104], C. Markert[118], M. Marquard[69], N.A. Martin[104,102], P. Martinengo[34], J.L. Martinez[125], M.I. Martínez[44], G. Martínez García[113], M. Martinez Pedreira[34], S. Masciocchi[104], M. Masera[26], A. Masoni[54], L. Massacrier[61], E. Masson[113], A. Mastroserio[52,137], A.M. Mathis[103,116], P.F.T. Matuoka[120], A. Matyja[129,117], C. Mayer[117], M. Mazzilli[33], M.A. Mazzoni[57], F. Meddi[23], Y. Melikyan[91], A. Menchaca-Rocha[72], E. Meninno[30], M. Meres[14], S. Mhlanga[124], Y. Miake[132], L. Micheletti[26], M.M. Mieskolainen[43], D.L. Mihaylov[103], K. Mikhaylov[64,75], A. Mischke[63,i], A.N. Mishra[70], D. Miśkowiec[104], J. Mitra[140], C.M. Mitu[68], N. Mohammadi[34], A.P. Mohanty[63], B. Mohanty[85], M. Mohisin Khan[17,iv], M.M. Mondal[66], C. Mordasini[103], D.A. Moreira De Godoy[143], L.A.P. Moreno[44], S. Moretto[29], A. Morreale[113], A. Morsch[34], T. Mrnjavac[34], V. Muccifora[51], E. Mudnic[35], D. Mühlheim[143], S. Muhuri[140], M. Mukherjee[3], J.D. Mulligan[145], M.G. Munhoz[120], K. Münning[42], R.H. Munzer[69], H. Murakami[131], S. Murray[73], L. Musa[34], J. Musinsky[65], C.J. Myers[125], J.W. Myrcha[141], B. Naik[48], R. Nair[84], B.K. Nandi[48], R. Nania[53,10], E. Nappi[52], M.U. Naru[15], A.F. Nassirpour[80], H. Natal da Luz[120], C. Nattrass[129], S.R. Navarro[44], K. Nayak[85], R. Nayak[48], T.K. Nayak[140,85], S. Nazarenko[106], R.A. Negrao De Oliveira[69], L. Nellen[70], S.V. Nesbo[36], G. Neskovic[39], F. Ng[125], B.S. Nielsen[88], S. Nikolaev[87], S. Nikulin[87], V. Nikulin[96], F. Noferini[10,53], P. Nomokonov[75], G. Nooren[63], J.C.C. Noris[44], J. Norman[78], A. Nyanin[87], J. Nystrand[22], M. Ogino[81], A. Ohlson[102], J. Oleniacz[141], A.C. Oliveira Da Silva[120], M.H. Oliver[145], J. Onderwaater[104], C. Oppedisano[58], R. Orava[43], A. Ortiz Velasquez[70], A. Oskarsson[80], J. Otwinowski[117], K. Oyama[81], Y. Pachmayer[102], V. Pacik[88], D. Pagano[139], G. Paić[70], P. Palni[6], J. Pan[142], A.K. Pandey[48], S. Panebianco[136], R. Panse[39], V. Papikyan[1], P. Pareek[49], J. Park[60], J.E. Parkkila[126], S. Parmar[98], A. Passfeld[143], S.P. Pathak[125], R.N. Patra[140], B. Paul[58], H. Pei[6], T. Peitzmann[63], X. Peng[6], L.G. Pereira[71], H. Pereira Da Costa[136], D. Peresunko[87], G.M. Perez[8], E. Perez Lezama[69], J. Peschek[39], V. Peskov[69], Y. Pestov[4], V. Petráček[37], M. Petrovici[47], R.P. Pezzi[71], S. Piano[59], M. Pikna[14], P. Pillot[113], L.O.D.L. Pimentel[88], O. Pinazza[53,34], L. Pinsky[125], S. Pisano[51], D.B. Piyarathna[125], M. Płoskoń[79], M. Planinic[97], F. Pliquett[69], J. Pluta[141], S. Pochybova[144], P.L.M. Podesta-Lerma[119], M.G. Poghosyan[94], B. Polichtchouk[90], N. Poljak[97], W. Poonsawat[114], A. Pop[47], H. Poppenborg[143], S. Porteboeuf-Houssais[133], V. Pozdniakov[75], S.K. Prasad[3], R. Preghenella[53], F. Prino[58], C.A. Pruneau[142], I. Pshenichnov[62], M. Puccio[26], V. Punin[106], K. Puranapanda[140], J. Putschke[142], R.E. Quishpe[125], S. Ragoni[108], S. Raha[3], S. Rajput[99], J. Rak[126], A. Rakotozafindrabe[136], L. Ramello[32], F. Rami[135], R. Raniwala[100], S. Raniwala[100], S.S. Räsänen[43], B.T. Rascanu[69], R. Rath[49], V. Ratza[42], I. Ravasenga[31], K.F. Read[129,94], K. Redlich[84,v], A. Rehman[22], P. Reichelt[69], F. Reidt[34], X. Ren[6], R. Renfordt[69], A. Reshetin[62], J.-P. Revol[10], K. Reygers[102], V. Riabov[96], T. Richert[88,80], M. Richter[21], P. Riedler[34], W. Riegler[34], F. Riggi[28], C. Ristea[68], S.P. Rode[49], M. Rodríguez Cahuantzi[44], K. Røed[21], R. Rogalev[90], E. Rogochaya[75], D. Rohr[34], D. Röhrich[22], P.S. Rokita[141], F. Ronchetti[51], E.D. Rosas[70], K. Roslon[141], P. Rosnet[133], A. Rossi[56,29], A. Rotondi[138], F. Roukoutakis[83], A. Roy[49], P. Roy[107], O.V. Rueda[80], R. Rui[25], B. Rumyantsev[75], A. Rustamov[86], E. Ryabinkin[87], Y. Ryabov[96], A. Rybicki[117], S. Saarinen[43], S. Sadhu[140], S. Sadovsky[90], K. Šafařík[34,37], S.K. Saha[140], B. Sahoo[48], P. Sahoo[49], R. Sahoo[49], S. Sahoo[66], P.K. Sahu[66], J. Saini[140], S. Sakai[132], S. Sambyal[99], V. Samsonov[91,96], A. Sandoval[72], A. Sarkar[73], D. Sarkar[140], N. Sarkar[140], P. Sarma[41], V.M. Sarti[103], M.H.P. Sas[63], E. Scapparone[53], B. Schaefer[94], J. Schambach[118], H.S. Scheid[69], C. Schiaua[47], R. Schicker[102], A. Schmah[102], C. Schmidt[104], H.R. Schmidt[101], M.O. Schmidt[102], M. Schmidt[101], N.V. Schmidt[94,69], A.R. Schmier[129], J. Schukraft[34,88], Y. Schutz[34,135], K. Schwarz[104], K. Schweda[104], G. Scioli[27], E. Scomparin[58], M. Šefčík[38], J.E. Seger[16], Y. Sekiguchi[131], D. Sekihata[45], I. Selyuzhenkov[104,91], S. Senyukov[135], E. Serradilla[72], P. Sett[48], A. Sevcenco[68], A. Shabanov[62], A. Shabetai[113], R. Shahoyan[34], W. Shaikh[107], A. Shangaraev[90], A. Sharma[98], A. Sharma[99], M. Sharma[99], N. Sharma[98], A.I. Sheikh[140], K. Shigaki[45], M. Shimomura[82], S. Shirinkin[64], Q. Shou[6,110], Y. Sibiriak[87], S. Siddhanta[54], T. Siemiarczuk[84], D. Silvermyr[80], G. Simatovic[89], G. Simonetti[103,34], R. Singh[85], R. Singh[99], V.K. Singh[140], V. Singhal[140], T. Sinha[107], B. Sitar[14], M. Sitta[32], T.B. Skaali[21], M. Slupecki[126], N. Smirnov[145], R.J.M. Snellings[63], T.W. Snellman[126], J. Sochan[115], C. Soncco[109], J. Song[60], A. Songmoolnak[114], F. Soramel[29], S. Sorensen[129], F. Sozzi[104], I. Sputowska[117], J. Stachel[102], I. Stan[68], P. Stankus[94], T. M. Steinbeck[39], E. Stenlund[80], D. Stocco[113], M.M. Storetvedt[36], P. Strmen[14], A.A.P. Suaide[120], T. Sugitate[45], C. Suire[61], M. Suleymanov[15], M. Suljic[34], R. Sultanov[64], M. Šumbera[93], S. Sumowidagdo[50], K. Suzuki[112], S. Swain[66], A. Szabo[14], I. Szarka[14], U. Tabassam[15], J. Takahashi[121], G.J. Tambave[22], N. Tanaka[132], M. Tarhini[113], M.G. Tarzila[47], A. Tauro[34], G. Tejeda Muñoz[44], A. Telesca[34], C. Terrevoli[29,125], J. M. Thaeder[39], D. Thakur[49], S. Thakur[140], D. Thomas[118], F. Thoresen[88], R. Tieulent[134], A. Tikhonov[62], A.R. Timmins[125], A. Toia[69], N. Topilskaya[62], M. Toppi[51], S.R. Torres[119], S. Tripathy[49], T. Tripathy[48], S. Trogolo[26], G. Trombetta[33], L. Tropp[38], V. Trubnikov[2], W.H. Trzaska[126], T.P. Trzcinski[141], B.A. Trzeciak[63], T. Tsuji[131], A. Tumkin[106], R. Turrisi[56], T.S. Tveter[21], K. Ullaland[22], E.N. Umaka[125], A. Uras[134], G.L. Usai[24], A. Utrobicic[97], M. Vala[38,115], L. Valencia Palomo[44], N. Valle[138], N. van der Kolk[63], L.V.R. van Doremalen[63], J.W. Van Hoorne[34], M. van Leeuwen[63], P. Vande Vyvre[34], D. Varga[144], A. Vargas[44], M. Vargyas[126], R. Varma[48], M. Vasileiou[83], A. Vasiliev[87], O. Vázquez Doce[116,103], V. Vechernin[111], A.M. Veen[63], E. Vercellin[26], S. Vergara Limón[44], L. Vermunt[63], R. Vernet[7], R. Vértesi[144], L. Vickovic[35], J. Viinikainen[126], Z. Vilakazi[130], O. Villalobos Baillie[108], A. Villatoro Tello[44], G. Vino[52], A. Vinogradov[87], T. Virgili[30], V. Vislavicius[88], A. Vodopyanov[75], B. Volkel[34], M.A. Völkl[101], K. Voloshin[64], S.A. Voloshin[142], G. Volpe[33], B. von Haller[34], I. Vorobyev[103,116], D. Voscek[115], J. Vrláková[38], B. Wagner[22], M. Wang[6], Y. Watanabe[132], M. Weber[112], S.G. Weber[104], A. Wegrzynek[34], D.F. Weiser[102], S.C. Wenzel[34], J.P. Wessels[143], U. Westerhoff[143], A.M. Whitehead[124], E. Widmann[112], J. Wiechula[69], J. Wikne[21], G. Wilk[84], J. Wilkinson[53], G.A. Willems[143,34], E. Willsher[108], B. Windelband[102], W.E. Witt[129], Y. Wu[128], R. Xu[6], S. Yalcin[77], K. Yamakawa[45], S. Yano[136], Z. Yin[6], H. Yokoyama[63], I.-K. Yoo[18], J.H. Yoon[60], S. Yuan[22], V. Yurchenko[2], V. Zaccolo[58,25], A. Zaman[15], C. Zampolli[34], H.J.C. Zanoli[120], N. Zardoshti[34,108], A. Zarochentsev[111], P. Závada[67], N. Zaviyalov[106], H. Zbroszczyk[141], M. Zhalov[96], X. Zhang[6], Y. Zhang[6], Z. Zhang[6,133], C. Zhao[21], V. Zherebchevskii[111], N. Zhigareva[64], D. Zhou[6], Y. Zhou[88], Z. Zhou[22], H. Zhu[6], J. Zhu[6], Y. Zhu[6], A. Zichichi[27,10], M.B. Zimmermann[34], G. Zinovjev[2], N. Zurlo[139],

Affiliation notes

[i] Deceased.

[ii] Dipartimento DET del Politecnico di Torino, Turin, Italy.

[iii] M.V. Lomonosov Moscow State University, D.V. Skobeltsyn Institute of Nuclear, Physics, Moscow, Russia.

[iv] Department of Applied Physics, Aligarh Muslim University, Aligarh, India.

[v] Institute of Theoretical Physics, University of Wroclaw, Poland.

Collaboration institutes

[1] A.I. Alikhanyan National Science Laboratory (Yerevan Physics Institute) Foundation, Yerevan, Armenia
[2] Bogolyubov Institute for Theoretical Physics, National Academy of Sciences of Ukraine, Kiev, Ukraine
[3] Bose Institute, Department of Physics and Centre for Astroparticle Physics and Space Science (CAPSS), Kolkata, India
[4] Budker Institute for Nuclear Physics, Novosibirsk, Russia
[5] California Polytechnic State University, San Luis Obispo, California, United States
[6] Central China Normal University, Wuhan, China
[7] Centre de Calcul de l'IN2P3, Villeurbanne, Lyon, France
[8] Centro de Aplicaciones Tecnológicas y Desarrollo Nuclear (CEADEN), Havana, Cuba
[9] Centro de Investigación y de Estudios Avanzados (CINVESTAV), Mexico City and Mérida, Mexico
[10] Centro Fermi — Museo Storico della Fisica e Centro Studi e Ricerche "Enrico Fermi', Rome, Italy
[11] Chicago State University, Chicago, Illinois, United States
[12] China Institute of Atomic Energy, Beijing, China
[13] Chonbuk National University, Jeonju, Republic of Korea
[14] Comenius University Bratislava, Faculty of Mathematics, Physics and Informatics, Bratislava, Slovakia
[15] COMSATS Institute of Information Technology (CIIT), Islamabad, Pakistan
[16] Creighton University, Omaha, Nebraska, United States
[17] Department of Physics, Aligarh Muslim University, Aligarh, India
[18] Department of Physics, Pusan National University, Pusan, Republic of Korea
[19] Department of Physics, Sejong University, Seoul, Republic of Korea
[20] Department of Physics, University of California, Berkeley, California, United States
[21] Department of Physics, University of Oslo, Oslo, Norway
[22] Department of Physics and Technology, University of Bergen, Bergen, Norway
[23] Dipartimento di Fisica dell'Università 'La Sapienza' and Sezione INFN, Rome, Italy
[24] Dipartimento di Fisica dell'Università and Sezione INFN, Cagliari, Italy
[25] Dipartimento di Fisica dell'Università and Sezione INFN, Trieste, Italy
[26] Dipartimento di Fisica dell'Università and Sezione INFN, Turin, Italy
[27] Dipartimento di Fisica e Astronomia dell'Università and Sezione INFN, Bologna, Italy
[28] Dipartimento di Fisica e Astronomia dell'Università and Sezione INFN, Catania, Italy
[29] Dipartimento di Fisica e Astronomia dell'Università and Sezione INFN, Padova, Italy
[30] Dipartimento di Fisica 'E.R. Caianiello' dell'Università and Gruppo Collegato INFN, Salerno, Italy
[31] Dipartimento DISAT del Politecnico and Sezione INFN, Turin, Italy
[32] Dipartimento di Scienze e Innovazione Tecnologica dell'Università del Piemonte Orientale and INFN Sezione di Torino, Alessandria, Italy
[33] Dipartimento Interateneo di Fisica 'M. Merlin' and Sezione INFN, Bari, Italy
[34] European Organization for Nuclear Research (CERN), Geneva, Switzerland
[35] Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Split, Croatia
[36] Faculty of Engineering and Science, Western Norway University of Applied Sciences, Bergen, Norway
[37] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Prague, Czech Republic
[38] Faculty of Science, P.J. Šafárik University, Košice, Slovakia
[39] Frankfurt Institute for Advanced Studies, Johann Wolfgang Goethe-Universität Frankfurt, Frankfurt, Germany
[40] Gangneung-Wonju National University, Gangneung, Republic of Korea
[41] Gauhati University, Department of Physics, Guwahati, India
[42] Helmholtz-Institut für Strahlen- und Kernphysik, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany
[43] Helsinki Institute of Physics (HIP), Helsinki, Finland
[44] High Energy Physics Group, Universidad Autónoma de Puebla, Puebla, Mexico
[45] Hiroshima University, Hiroshima, Japan
[46] Hochschule Worms, Zentrum für Technologietransfer und Telekommunikation (ZTT), Worms, Germany
[47] Horia Hulubei National Institute of Physics and Nuclear Engineering, Bucharest, Romania
[48] Indian Institute of Technology Bombay (IIT), Mumbai, India

[49] Indian Institute of Technology Indore, Indore, India
[50] Indonesian Institute of Sciences, Jakarta, Indonesia
[51] INFN, Laboratori Nazionali di Frascati, Frascati, Italy
[52] INFN, Sezione di Bari, Bari, Italy
[53] INFN, Sezione di Bologna, Bologna, Italy
[54] INFN, Sezione di Cagliari, Cagliari, Italy
[55] INFN, Sezione di Catania, Catania, Italy
[56] INFN, Sezione di Padova, Padova, Italy
[57] INFN, Sezione di Roma, Rome, Italy
[58] INFN, Sezione di Torino, Turin, Italy
[59] INFN, Sezione di Trieste, Trieste, Italy
[60] Inha University, Incheon, Republic of Korea
[61] Institut de Physique Nucléaire d'Orsay (IPNO), Institut National de Physique Nucléaire et de Physique des Particules (IN2P3/CNRS), Université de Paris-Sud, Université Paris-Saclay, Orsay, France
[62] Institute for Nuclear Research, Academy of Sciences, Moscow, Russia
[63] Institute for Subatomic Physics, Utrecht University/Nikhef, Utrecht, Netherlands
[64] Institute for Theoretical and Experimental Physics, Moscow, Russia
[65] Institute of Experimental Physics, Slovak Academy of Sciences, Košice, Slovakia
[66] Institute of Physics, Homi Bhabha National Institute, Bhubaneswar, India
[67] Institute of Physics of the Czech Academy of Sciences, Prague, Czech Republic
[68] Institute of Space Science (ISS), Bucharest, Romania
[69] Institut für Kernphysik, Johann Wolfgang Goethe-Universität Frankfurt, Frankfurt, Germany
[70] Instituto de Ciencias Nucleares, Universidad Nacional Autónoma de México, Mexico City, Mexico
[71] Instituto de Física, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brazil
[72] Instituto de Física, Universidad Nacional Autónoma de México, Mexico City, Mexico
[73] iThemba LABS, National Research Foundation, Somerset West, South Africa
[74] Johann-Wolfgang-Goethe Universität Frankfurt Institut für Informatik, Fachbereich Informatik und Mathematik, Frankfurt, Germany
[75] Joint Institute for Nuclear Research (JINR), Dubna, Russia
[76] Korea Institute of Science and Technology Information, Daejeon, Republic of Korea
[77] KTO Karatay University, Konya, Turkey
[78] Laboratoire de Physique Subatomique et de Cosmologie, Université Grenoble-Alpes, CNRS-IN2P3, Grenoble, France
[79] Lawrence Berkeley National Laboratory, Berkeley, California, United States
[80] Lund University Department of Physics, Division of Particle Physics, Lund, Sweden
[81] Nagasaki Institute of Applied Science, Nagasaki, Japan
[82] Nara Women's University (NWU), Nara, Japan
[83] National and Kapodistrian University of Athens, School of Science, Department of Physics , Athens, Greece
[84] National Centre for Nuclear Research, Warsaw, Poland
[85] National Institute of Science Education and Research, Homi Bhabha National Institute, Jatni, India
[86] National Nuclear Research Center, Baku, Azerbaijan
[87] National Research Centre Kurchatov Institute, Moscow, Russia
[88] Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark
[89] Nikhef, National institute for subatomic physics, Amsterdam, Netherlands
[90] NRC Kurchatov Institute IHEP, Protvino, Russia
[91] NRNU Moscow Engineering Physics Institute, Moscow, Russia
[92] Nuclear Physics Group, STFC Daresbury Laboratory, Daresbury, United Kingdom
[93] Nuclear Physics Institute of the Czech Academy of Sciences, Řež u Prahy, Czech Republic
[94] Oak Ridge National Laboratory, Oak Ridge, Tennessee, United States
[95] Ohio State University, Columbus, Ohio, United States
[96] Petersburg Nuclear Physics Institute, Gatchina, Russia
[97] Physics department, Faculty of science, University of Zagreb, Zagreb, Croatia

[98] Physics Department, Panjab University, Chandigarh, India
[99] Physics Department, University of Jammu, Jammu, India
[100] Physics Department, University of Rajasthan, Jaipur, India
[101] Physikalisches Institut, Eberhard-Karls-Universität Tübingen, Tübingen, Germany
[102] Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany
[103] Physik Department, Technische Universität München, Munich, Germany
[104] Research Division and ExtreMe Matter Institute EMMI, GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany
[105] Rudjer Bošković Institute, Zagreb, Croatia
[106] Russian Federal Nuclear Center (VNIIEF), Sarov, Russia
[107] Saha Institute of Nuclear Physics, Homi Bhabha National Institute, Kolkata, India
[108] School of Physics and Astronomy, University of Birmingham, Birmingham, United Kingdom
[109] Sección Física, Departamento de Ciencias, Pontificia Universidad Católica del Perú, Lima, Peru
[110] Shanghai Institute of Applied Physics, Shanghai, China
[111] St. Petersburg State University, St. Petersburg, Russia
[112] Stefan Meyer Institut für Subatomare Physik (SMI), Vienna, Austria
[113] SUBATECH, IMT Atlantique, Université de Nantes, CNRS-IN2P3, Nantes, France
[114] Suranaree University of Technology, Nakhon Ratchasima, Thailand
[115] Technical University of Košice, Košice, Slovakia
[116] Technische Universität München, Excellence Cluster 'Universe', Munich, Germany
[117] The Henryk Niewodniczanski Institute of Nuclear Physics, Polish Academy of Sciences, Cracow, Poland
[118] The University of Texas at Austin, Austin, Texas, United States
[119] Universidad Autónoma de Sinaloa, Culiacán, Mexico
[120] Universidade de São Paulo (USP), São Paulo, Brazil
[121] Universidade Estadual de Campinas (UNICAMP), Campinas, Brazil
[122] Universidade Federal do ABC, Santo Andre, Brazil
[123] University College of Southeast Norway, Tonsberg, Norway
[124] University of Cape Town, Cape Town, South Africa
[125] University of Houston, Houston, Texas, United States
[126] University of Jyväskylä, Jyväskylä, Finland
[127] University of Liverpool, Liverpool, United Kingdom
[128] University of Science and Technology of China, Hefei, China
[129] University of Tennessee, Knoxville, Tennessee, United States
[130] University of the Witwatersrand, Johannesburg, South Africa
[131] University of Tokyo, Tokyo, Japan
[132] University of Tsukuba, Tsukuba, Japan
[133] Université Clermont Auvergne, CNRS/IN2P3, LPC, Clermont-Ferrand, France
[134] Université de Lyon, Université Lyon 1, CNRS/IN2P3, IPN-Lyon, Villeurbanne, Lyon, France
[135] Université de Strasbourg, CNRS, IPHC UMR 7178, F-67000 Strasbourg, France, Strasbourg, France
[136] Université Paris-Saclay Centre d Études de Saclay (CEA), IRFU, Department de Physique Nucléaire (DPhN), Saclay, France
[137] Università degli Studi di Foggia, Foggia, Italy
[138] Università degli Studi di Pavia, Pavia, Italy
[139] Università di Brescia, Brescia, Italy
[140] Variable Energy Cyclotron Centre, Homi Bhabha National Institute, Kolkata, India
[141] Warsaw University of Technology, Warsaw, Poland
[142] Wayne State University, Detroit, Michigan, United States
[143] Westfälische Wilhelms-Universität Münster, Institut für Kernphysik, Münster, Germany
[144] Wigner Research Centre for Physics, Hungarian Academy of Sciences, Budapest, Hungary
[145] Yale University, New Haven, Connecticut, United States
[146] Yonsei University, Seoul, Republic of Korea

References

[1] ALICE Collaboration, K. Aamodt, et al., J. Instrum. 3 (08) (2008) S08002.
[2] L. Evans, P. Bryant, J. Instrum. 3 (2008) S08001, http://dx.doi.org/10.1088/1748-0221/3/08/S08001.
[3] ALICE Collaboration, J. Adam, et al., Phys. Rev. Lett. 116 (22) (2016) 222302, http://dx.doi.org/10.1103/PhysRevLett.116.222302, arXiv:1512.06104.
[4] ALICE Collaboration, J. Alme, et al., Nucl. Instrum. Methods Phys. Res. A 622 (1) (2010) 316–367, http://dx.doi.org/10.1016/j.nima.2010.04.042.
[5] ALICE Collaboration, G. Dellacasa, et al., ALICE Inner Tracking System (ITS): Technical Design Report, Technical Design Report ALICE, CERN-LHCC-99-12, CERN, Geneva, 1999.
[6] ALICE Collaboration, S. Acharya, et al., Nucl. Instrum. Meth. A881 (2018) 88–127, http://dx.doi.org/10.1016/j.nima.2017.09.028, arXiv:1709.02743.
[7] ALICE Collaboration, P. Cortese, et al., Technical Design Report ALICE, CERN-LHCC-2002-016, CERN, Geneva, 2002.
[8] ALICE Collaboration, G. Dellacasa, et al., Technical Design Report ALICE, CERN-LHCC-99-04, CERN, Geneva, 1999.
[9] ALICE Collaboration, P. Cortese, et al., Technical Design Report ALICE, CERN-LHCC-2008-014, CERN-ALICE-TDR-014, CERN, Geneva, 2008.
[10] ALICE Collaboration, ALICE technical design report of the dimuon forward spectrometer, 1999.
[11] ALICE Collaboration, P. Cortese, et al., Technical Design Report ALICE, CERN-LHCC-2004-025, CERN, Geneva, 2004.
[12] ALICE Collaboration, G. Dellacasa, et al., Technical Design Report ALICE, CERN-LHCC-99-05, CERN, Geneva, 1999.
[13] C. Adler, et al., Proceedings of the 5th International Workshop on Applied Parallel Computing, New Paradigms for HPC in Industry and Academia, in: PARA '00, Springer-Verlag, London, UK, UK, 2001, pp. 333–341.
[14] V. Lindenstruth, I. Kisel, Nucl. Instrum. Methods Phys. Res. A 535 (2004) 48–56, http://dx.doi.org/10.1016/j.nima.2004.07.267.
[15] V. Lindenstruth, IEEE Micro 26 (2) (2006) 48–57, http://dx.doi.org/10.1109/MM.2006.29.
[16] J. Alme, et al., J. Instrum. 8 (12) (2013) C12032, http://stacks.iop.org/1748-0221/8/i=12/a=C12032.
[17] T. Kollegger, The ALICE High Level Trigger: The 2011 run experience, in: 2012 18th IEEE-NPSS Real Time Conference, 2012, pp. 1–4, http://dx.doi.org/10.1109/RTC.2012.6418366.
[18] R.E. Panse, PhD thesis, University of Heidelberg, 2009.
[19] Intelligent Platform Management Interface, https://www.intel.com/content/www/us/en/servers/ipmi/ipmi-second-gen-interface-spec-v2-rev1-1.html.
[20] D. Rohr, S. Kalcher, M. Bach, A. Alaqeeli, H. Alzaid, D. Eschweiler, V. Lindenstruth, A. Sakhar, A. Alharthi, A. Almubarak, I. Alqwaiz, R. Bin Suliman, Proceedings of the 16th IEEE International Conference on High Performance Computing and Communications, HPCC 2014, Paris, France. IEEE, IEEE, 2014, pp. 42–45, http://dx.doi.org/10.1109/HPCC.2014.14.
[21] P. Buncic, M. Krzewicki, P. Vande Vyvre, Tech. Rep. CERN-LHCC-2015-006. ALICE-TDR-019, CERN, Geneva, 2015, https://cds.cern.ch/record/2011297.
[22] T. Alt, PhD thesis, Goethe-University Frankfurt, 2017.
[23] A. Borga, et al., J. Instrum. 10 (02) (2015) C02022, http://stacks.iop.org/1748-0221/10/i=02/a=C02022.
[24] D. Eschweiler, V. Lindenstruth, Proceedings of the 16th Real-Time Linux Workshop, Open Source Automation Development Lab (OSADL), Duesseldorf, Germany, 2014.
[25] H. Engel, T. Alt, T. Breitner, A.G. Ramirez, T. Kollegger, M. Krzewicki, J. Lehrbach, D. Rohr, U. Kebschull, J. Instrum. 11 (01) (2016) C01041, http://stacks.iop.org/1748-0221/11/i=01/a=C01041.
[26] Foreman lifecycle management tool, https://theforeman.org. (Accessed: 2018-12-03).
[27] Puppet software configuration management tool, https://puppet.com/. (Accessed: 2018-12-03).
[28] Zabbix LLC, Institution, http://www.zabbix.com/. (Accessed: 2018-12-03).
[29] ALICE Collaboration, I. Bird, et al., Tech. Rep. CERN-LHCC-2014-014, LCG-TDR-002, CERN, Geneva, 2014, https://cds.cern.ch/record/1695401?ln=en.
[30] T. Rosado, J. Bernardino, An Overview of Openstack Architecture, in: Proceedings of the 18th International Database Engineering, IDEAS '14, 2014, pp. 366–367, http://doi.acm.org/10.1145/2628194.2628195.
[31] D. Merkel, Linux J. 2014 (239) (2014) http://dl.acm.org/citation.cfm?id=2600239.2600241.
[32] R. Brun, F. Rademakers, New Computing Techniques in Physics Research V. Proceedings, 5th International Workshop, AIHENP '96, Lausanne, Switzerland, September 2-6, 1996, Nucl. Instrum. Meth. A389 (1997) 81–86, http://dx.doi.org/10.1016/S0168-9002(97)00048-X.
[33] M. Richter, T. Alt, S. Bablok, C. Cheshkov, P.T. Hille, V. Lindenstruth, G. Ovrebekk, M. Ploskon, S. Popescu, D. Rohrich, T.M. Steinbeck, J.M. Thader, IEEE Trans. Nucl. Sci. 55 (1) (2007) 133–138, http://dx.doi.org/10.1109/TNS.2007.913469.
[34] T.M. Steinbeck, V. Lindenstruth, M.W. Schulz, IEEE Trans. Nucl. Sci. 49 (2) (2002) 455–459, http://dx.doi.org/10.1109/TNS.2002.1003773.
[35] T.M. Steinbeck, V. Lindenstruth, D. Röhrich, A.S. Vestbo, A. Wiebalck, in: J. Fagerholm, J. Haataja, J. Järvinen, M. Lyly, P. Råback, V. Savolainen (Eds.), Applied Parallel Computing: Advanced Scientific Computing 6th International Conference, PARA 2002 Espoo, Finland, June 15-18, 2002 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 454–464, http://dx.doi.org/10.1007/3-540-48051-X_45.
[36] M. Krzewicki, V. Lindenstruth, ALICE Collaboration, J. Phys. Conf. Ser. 898 (3) (2017) 032056, http://stacks.iop.org/1742-6596/898/i=3/a=032056.
[37] B. Becker, S. Chattopadhyay, C. Cicalo, J. Cleymans, G. de Vaux, R.W. Fearick, V. Lindenstruth, M. Richter, D. Rohrich, F. Staley, T.M. Steinbeck, A. Szostak, H. Tilsner, R. Weis, Z.Z. Vilakazi, IEEE Trans. Nucl. Sci. 55 (2) (2008) 703–709, http://dx.doi.org/10.1109/TNS.2008.918521.
[38] G.E. Moore, Electronics 38 (8) (1965) 114–117, http://dx.doi.org/10.1109/jproc.1998.658762.
[39] G. Grastveit, H. Helstrup, V. Lindenstruth, C. Loizides, D. Roehrich, B. Skaali, T. Steinbeck, R. Stock, H. Tilsner, K. Ullaland, A. Vestbo, T. Vik, FPGA Co-processor for the ALICE High Level Trigger, 2003, ArXiv Physics e-prints, arXiv:physics/0306017.
[40] S. Gorbunov, U. Kebschull, I. Kisel, V. Lindenstruth, W.F.J. Müller, Comput. Phys. Comm. 178 (2008) 374–383.
[41] I. Kisel, Nucl. Instrum. Methods Phys. Res. A 566 (1) (2006) 85–88, http://dx.doi.org/10.1016/j.nima.2006.05.040, Proceedings of the 1st Workshop on Tracking in High Multiplicity Environments1st Workshop on Tracking in High Multiplicity Environments.
[42] R.E. Kalman, J. Basic Eng. 82 (1960) 35–45, http://dx.doi.org/10.1109/ICASSP.1982.1171734.
[43] D. Rohr, S. Gorbunov, V. Lindenstruth, ALICE Collaboration, J. Phys. Conf. Ser. 898 (3) (2017) 032030, http://stacks.iop.org/1742-6596/898/i=3/a=032030.
[44] D. Rohr, S. Gorbunov, A. Szostak, M. Kretz, T. Kollegger, T. Breitner, T. Alt, J. Phys. Conference Series, Proceedings of 19th International Conference on Computing in High Energy and Nuclear Physics 396 (1) (2012) 012044, http://stacks.iop.org/1742-6596/396/i=1/a=012044.
[45] M. Kretz, V. Lindenstruth, Softw. Pract. Exper. 42 (11) (2012) 1409–1430, http://dx.doi.org/10.1002/spe.1149, http://dx.doi.org/10.1002/spe.1149.
[46] ISO/IEC JTC1 SC22 WG21 N4744 (2018) http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2018/n4742.html.
[47] ALICE Collaboration, D. Rohr, S. Gorbunov, M. Krzewicki, T. Breitner, M. Kretz, V. Lindenstruth, Proceedings, 21st International Conference on Computing in High Energy and Nuclear Physics (CHEP 2015): Okinawa, Japan, April 13-17, 2015, J. Phys. Conf. Ser. 664 (8) (2015) 082047, http://dx.doi.org/10.1088/1742-6596/664/8/082047, arXiv:1712.09416.
[48] H.H. Gutbrod, et al., (Eds.) FAIR - Baseline Technical Report, 2006, ISBN: 3-9811298-0-6.
[49] M. Gyulassy, X.-N. Wang, Comput. Phys. Comm. 83 (1994) 307, http://dx.doi.org/10.1016/0010-4655(94)90057-4, arXiv:nucl-th/9502021.
[50] M. Krzewicki, PhD thesis, Utrecht University, 2013.
[51] J. Berger, U. Frankenfeld, V. Lindenstruth, P. Plamper, D. Röhrich, E. Schäfer, M.W. Schulz, T.M. Steinbeck, R. Stock, K. Sulimma, A. Vestbø, A. Wiebalck, Nucl. Instrum. Methods Phys. Res. A 489 (13) (2002) 406–421, http://dx.doi.org/10.1016/S0168-9002(02)00792-1.
[52] D.A. Huffman, Proc. IRE 40 (9) (1952) 1098–1101, http://dx.doi.org/10.1109/JRPROC.1952.273898.
[53] D. Röhrich, A. Vestbø, Nucl. Instrum. Methods Phys. Res. A 566 (2) (2006) 668–674, http://dx.doi.org/10.1016/j.nima.2006.06.056.
[54] ALICE Collaboration, D. Rohr, Tracking performance in high multiplicities environment at ALICE, in: 5th Large Hadron Collider Physics Conference (LHCP 2017) Shanghai, China, May 15-20, 2017, arXiv:1709.00618.
[55] M. Krzewicki, D. Rohr, C. Zampolli, J. Wiechula, S. Gorbunov, A. Chauvin, I. Vorobyev, S. Weber, K. Schweda, R. Shahoyan, V. Lindenstruth, A. Collaboration, J. Phys. Conf. Ser. 898 (3) (2017) 032055, http://stacks.iop.org/1742-6596/898/i=3/a=032055.
[56] D. Rohr, M. Krzwicki, H. Engel, J. Lehrbach, V. Lindenstruth, J. Phys. Conf. Ser. 898 (3) (2017) 032031, http://stacks.iop.org/1742-6596/898/i=3/a=032031.
[57] A. Szostak, J. Phys. Conf. Ser. 396 (1) (2012) 012048, http://stacks.iop.org/1742-6596/396/i=1/a=012048.