Operational experience with the ALICE High Level Trigger

Artur Szostak\textsuperscript{1} for the ALICE collaboration

\textsuperscript{1} Department of Physics and Technology, University of Bergen, Norway
E-mail: artursz@iafrica.com

Abstract. The ALICE HLT is a dedicated real-time system for online event reconstruction and triggering. Its main goal is to reduce the raw data volume read from the detectors by an order of magnitude, to fit within the available data acquisition bandwidth. This is accomplished by a combination of data compression and triggering. When HLT is enabled, data is recorded only for events selected by HLT. The combination of both approaches allows for flexible data reduction strategies. Event reconstruction places a high computational load on HLT. Thus, a large dedicated computing cluster is required, comprising 248 machines, all interconnected with InfiniBand. Running a large system like HLT in production mode proves to be a challenge. During the 2010 pp and Pb-Pb data-taking period, many problems were experienced that led to a sub-optimal operational efficiency. Lessons were learned and certain crucial changes were made to the architecture and software in preparation for the 2011 Pb-Pb run, in which HLT had a vital role performing data compression for ALICE’s largest detector, the TPC. An overview of the status of the HLT and experience from the 2010/2011 production runs are presented. Emphasis is given to the overall performance, showing an improved efficiency and stability in 2011 compared to 2010, attributed to the significant improvements made to the system. Further opportunities for improvement are identified and discussed.

1. Introduction

Raw data volumes from the readout systems of the Large Hadron Collider (LHC) experiments are huge. During lead-lead (Pb-Pb) collisions, the rate from a large Ion Collider Experiment (ALICE) is particularly high, with peak rates up to 25 GB/s delivered by the front-end electronics. However, the Data Acquisition (DAQ) system only has a 4 GB/s bandwidth on the back end to mass storage. Therefore, such a high rate cannot possibly be recorded to mass storage by DAQ alone.

A dedicated real-time online software system, called the High Level Trigger (HLT) [1], is used to reduce the data volume by an order of magnitude to fit within the available DAQ bandwidth. Two important approaches are used to varying degrees to achieve this goal,

- data compression and
- triggering.

The first tries to reduce the data volume by storing only relevant physics parameters extracted by online event reconstruction. The second approach relies on the fact that not all of the raw data contains useful information. Many of the events read out do not contain interesting physics signal. The HLT can reduce the data rate by reconstructing the raw data and then applying a
trigger algorithm to the results. If specific signals are found in an event, then it is marked in the HLT decision. DAQ receives all of these decisions and will record only those events that have been marked by the HLT as interesting, discarding the raw data for the others.

A second but equally vital function of the HLT is online monitoring [2]. The HLT has access to all raw data and status information from the detectors during data-taking. Combined with online event reconstruction, the HLT becomes a powerful monitoring tool for ensuring data quality. Many problems can only be spotted easily when looking at the high level information, such as reconstructed tracks. In addition, online compression and triggering must be monitored live during data-taking to ensure stability of the system and quality of recorded data.

The HLT became vital to data-taking for ALICE during 2011, when it provided data reduction (advanced compression) for the Time Projection Chamber (TPC) detector [3]. TPC data reduction involves replacing the original raw data by online reconstructed clusters that can be reformatted and Huffman coded, thereby significantly reducing the data rate to mass storage. At the same time, the physics content for offline analysis is preserved.

The TPC is the largest detector in ALICE and the largest contributor to the total data volume. HLT data reduction has allowed ALICE to record 160 million events during the 2011 Pb-Pb period, which is 77% more than for the corresponding 2010 period. The data storage budget was however still kept to 0.8 PB, a size similar to 2010.

Maintaining a complex system like the HLT is a challenge. The biggest contributors to the complexity are the large number of interfaces to external and internal subsystems, and the sheer size of the HLT computing cluster. A very long commissioning period was required during 2010 to make the HLT functional. However, further upgrades and commissioning were required early in 2011 to bring the operational efficiency to a sufficiently high level so that it could effectively participate in production data-taking.

2. Architecture
The HLT system, as used for the Pb-Pb run, physically comprises a large computing cluster built mostly from commodity components. Custom hardware is used only where necessary to connect the HLT to external custom interfaces. The main production cluster has 205 individual machines used for computation. These are divided into,

- 117 Front End Processor (FEP) machines,
- 84 Compute Nodes (CN) and
- 4 portal machines.

The FEP machines contain the custom HLT Readout Receiver Cards (RORCs) that are connected with fibre optic cables to the detectors. The connection is called a Detector Data Link (DDL). Raw data is received from the detectors over these channels and fed into the HLT compute farm. On the back end there are additional FEPs that connect the HLT with fibres to the Data Acquisition (DAQ) system, allowing to ship the results back to the DAQ for recording to permanent storage. A similar card is used for output as for input, but programmed to send data rather than receive it. This keeps the varieties of custom hardware in the cluster to a minimum. The role of the FEP machines is to provide the physical hardware to receive raw data from detectors and send the computed results from HLT to DAQ. They also provide some computing power for the very first stage of event reconstruction and very last stage of encoding the output results.

CN machines are used to run the main part of the HLT application software. They provide large numbers of cores and memory for computationally intensive tasks. 64 of the CNs have an additional dedicated Graphics Processing Unit (GPU) installed. These machines are distinguished by the name CNGPU and are used for particular computational tasks that map well to such hardware, such as particle tracking.
Figure 1. Schematic diagram of the HLT’s architecture indicating connections from detectors, to DAQ, other external services and internal network connections.

The portal machines interface the main computing cluster with external services or provide master node and management functionality for the application software. External services include the Experiment Control System (ECS), Detector Control System (DCS) and the offline conditions database, which stores the online detector parameters/calibrations. These interfaces and the DDL connections can be seen in figure 1.

The internal architecture of the FEP nodes is based around the Tyan Thunder h2000M (S3992) mainboard, while CN machines use one of the SuperMicro H8DGT or X8DTT-IBX mainboards, depending on processor model. FEPs contain dual 4-core AMD Opteron 2378 processors providing 8 cores per machine. CNs are divided into two classes of machines. The 34 older CNs and all 4 portal machines contain dual 8-core Intel Xeon E5520 processors, thus 16 cores per machine. The newer CNs were purchased in a second HLT installation and commissioning cycle to provide as much computing power as possible for the Pb-Pb runs. They contain dual 12-core AMD Opteron 6172 processors giving a total of 24 cores per machine. Some relevant parameters of the processors used are given in table 1. In total, there are 2744 cores available in the production cluster.

In terms of memory, the FEP machines use 333 MHz DDR2. Each FEP has a total of 12 GB of main memory, providing about 1.5 GB per core. CNs based on Intel processors have a total of 24 GB of 1066 MHz DDR3 main memory, with 1.5 GB per core; those using AMDs have 64 GB of 1333 MHz DDR3 memory, with a higher ratio of 2.67 GB per core. Total distributed
Table 1. Specifications for processors used in FEP and CN machines.

| Parameter                  | Intel Xeon E5520 | AMD Opteron 2378 | AMD Opteron 6172 |
|----------------------------|------------------|------------------|------------------|
| Clock rate (GHz)           | 2.26             | 2.4              | 2.1              |
| L1 cache (kB/core)         | 64               | 128              | 128              |
| L2 cache (kB/core)         | 256              | 512              | 512              |
| Shared L3 cache (MB)       | 8                | 6                | 12               |
| Memory bandwidth (GB/s)    | 25.6             | 12.8             | 42.7             |
| I/O speed (GT/s)           | 2×5.86           | 3×2.0            | 4×6.4            |

memory sums up to about 5.29 TB over the whole cluster.

Every machine in the cluster is interconnected with an InfiniBand network for the main application data flow and Gigabit Ethernet for management or monitoring. Both networks are built and configured to have a flat topology, meaning that the point-to-point bandwidth between any two machines in the cluster is guaranteed to be the same for any pair of machines. A flat network topology makes configuration of the HLT slightly easier, due to simplified optimisation requirements. Congestion on a particular point-to-point link is minimised by distributing the application software and services over multiple machines, such that any pair of machines have no more processes communicating with each other than would saturate the link. Finding the right HLT layout configuration that minimises congestion and maximises throughput is an optimisation problem that must be solved during HLT commissioning.

Any cluster requires various core services. 20 dedicated infrastructure servers are used to support these, such as mass storage, network services (DHCP, DNS, NTP, etc.), backup and cluster monitoring. In addition, a small non critical cluster comprising 23 machines provides a development and testing environment for the application software. These development machines have the same internal architecture as the FEPs.

3. Software

The HLT application software is run on the computing cluster in a parallel manner to achieve the required levels of performance. The underlying software framework uses the publisher/subscriber paradigm to implement a data flow model for data processing [4]. The reconstruction and compression algorithms are broken up into a number of stages. Each stage is implemented by a HLT component that takes input data from a previous stage and passes its processed output data to the next stage. The very first stage receives its input from the detectors over DDLs, while the last stage sends its output over DDLs to DAQ. A single component maps onto a process that is run on a CPU core. There are usually multiple instances of a component for a particular processing stage running on multiple cores and machines. The HLT software framework distributes the input data over each stage’s duplicate instances to balance the processing and network load.

The following is a brief description of the processing component layout for the TPC, which is the dominant detector in terms of data volume and processing load. Other ALICE detectors are handled in a similar manner in HLT. The first stage of processing is cluster finding, which involves identifying the centroids of the charge clusters deposited on the TPC’s readout plane. 216 of these components are run in parallel on the FEPs; one component for each input DDL, which connects to one partition of the TPC. The cluster-finding component takes advantage of the large Field Programmable Gate Arrays (FPGAs) on the RORCs to dramatically speed up
the processing. The cluster finding is bound to the FEPs since the RORCs are only installed on them. The found clusters are sent and collected from all TPC partitions on the CNGPUs for further processing. Here a tracking component uses the GPUs to perform track following and reconstruction from the cluster points [5]. The results are particle track parameters that are sent to monitoring and triggering components running on other CNs. In parallel, a second compression component transforms the cluster parameters into a format that can be highly compressed with Huffman coding [6]. The clusters are then encoded into a compressed data block and sent to the output FEPs. The reconstructed events arriving at the triggering stage have their track parameters analysed to search for particular signals indicating interesting physics. The result is a HLT decision, which is encoded and also sent to the output FEPs. All of the compressed clusters, track parameters and trigger decisions from earlier stages are collected at the output FEPs for each event. There, a data-formatting component packages this data into a DDL-compatible data stream to finally be sent to DAQ. A data flow diagram of the processing is indicated in figure 2.

4. 2011 Upgrades
During the 2010 data-taking period, it was seen that the HLT system was running at a lower than expected efficiency. The main cause of this was the high number of problems seen by subsystems interacting with Kerberos [7]. HLT initially had the Andrew File System (AFS) deployed on the cluster to provide distributed mass storage. That choice was made since AFS can provide effective client-side caching performance, which was needed to distribute the application software and calibration data required for a given run. However, AFS is tightly coupled with Kerberos to provide security and authentication, with no way to separate the two.

A typical recurrent problem seen was intermittent security ticket timeouts followed by failures when renewing the tickets. This led to various services being unavailable at the start or during a run and subsequent failure of HLT. The general complexity of AFS has also proven to lead to difficulties when trying to keep the access times within a deterministic limit. Unexpectedly high access times could easily lead to run failure, due to HLT taking longer to startup than the timeout threshold.

During the Christmas 2010 shutdown period, the mass storage system was significantly reworked to tackle these particular problems. The existing mass storage servers were upgraded with newer, higher performance Areca ARC-1220 RAID controllers and configured for RAID 10. Storage servers for meta-data, which are particularly sensitive to I/O latency, were upgraded with OCZ-Vertex3 solid state disks. All other data storage servers use Seagate ST3500630AS disks. Additional servers were also added to further distribute the client load on the mass storage system. The AFS file system was replaced with the Fraunhofer File System (FhGFS) [8] to serve
the production application software. FhGFS is less complex and has fewer features than AFS; however, most of the AFS features were not necessary in the HLT cluster and proved to be just a burden. The overall effect was that the mass storage system performance was significantly improved with much less complexity. The FhGFS configuration uses 2 servers for the meta-data and 4 additional data storage servers. The management daemon for the file system runs in a virtual machine, which is hosted on one of 2 virtual machine servers configured for mutual automatic failover. The physical machines used for all of these servers are the same class as the 16 core Intel based CN nodes.

An additional step was to install most of the application software on the local FEP and CN disks, rather than have a network installation on the AFS partitions. The benefit was a significantly reduced load on the mass storage system. However, software distribution becomes more complex. This trade off was nevertheless acceptable, as the overall result was a much more stable system. More importantly, all of the above changes combined to give a significantly faster startup procedure for HLT. Configuration startup times were reduced from 120–140 s down to 50–60 s, with similar improvements for engage times [9].

Some further instability was related to the poor separation of the production parts of the HLT system from the support infrastructure for testing and development. All testing and development machines used to be on the same subnet as the production system, since they were also treated as spare capacity. However, a few instances of interference were spotted, where routine development work performed by users in the cluster could seriously degrade the performance of the mass storage system, thereby causing startup timeouts or run failures.

For the 2011 period, the network topology was changed significantly to insulate the production cluster from the support infrastructure. All development and test machines were moved to their own network. The infrastructure machines providing core services were also moved onto their own subnet, with firewalls preventing any uncontrolled traffic from spilling over between the networks. InfiniBand was installed as a second network only on the production machines to handle the main data load. A fat tree structure [10] was used, which makes the point-to-point communication topology flat from the point of view of the compute nodes. The existing Gigabit Ethernet network was reconfigured to also use a fat tree structure.

In addition, a completely separated and dedicated production mass storage system was set up that stores only the things relevant for online running. It can only be accessed by the production compute nodes. Such changes completely eliminated accidental interference, allowing routine development and testing to continue in safety. Architecturally, both mass storage systems are identical. This provides the added benefit of a failover solution in case of complete failure of the production mass storage system.

5. Error Monitoring

A smaller but significant fraction of operational efficiency loss in the HLT has been due to the difficulty of identifying the root source of run failures. Experience has shown that often the turnaround times can be significantly improved if the root cause of a previous run failure can be accurately identified quickly enough. The problem here was related more to the high-level interpretation of the gathered information, rather than just with the monitoring or information gathering. Many subtle problems can only be correctly identified by looking for correlations between different sources of information, such as log files, traces of operational parameters (event rates, occupancy of internal buffers, etc.) and subsystem states. This not only includes information from HLT, but also external systems that HLT interfaces with, such as the ECS, DCS and DAQ.

This kind of inefficiency has been mitigated over the last year by implementing specific error detection agents that automatically analyse and look for correlations in the monitored information. Summary diagnostics messages are reported back to the shift crew by the agents...
whenever an error condition occurs that can be correlated with a specific set of well defined states or log entries. The messages include the useful information about the source of the problem with suggested solutions to fix the problem. Although this approach proves very useful to reduce the turnaround time and improve operational efficiency, there is a slightly higher maintenance cost to update what the agents must look for as the run conditions evolve.

The actual mechanism the agents employ to identify errors is to check whether a particular set of conditions has occurred within a short time window. The time window should be small enough to be able to assume that there is a causal connection between the conditions. The types of conditions looked for include,

- particular state transitions of the various HLT component management state machines,
- specific messages in the log files of running processes or external subsystems,
- checking if monitored parameters like buffer capacity goes over or below a certain threshold and
- verifying the trending behaviour of certain monitored parameters.

A certain set of conditions is mapped to a unique error. When such conditions occur, the predefined error can thus be correctly identified and reported to the user. Some of the errors that would have previously just shown up as HLT timeouts or backpressure with no explanation for the root cause, but have now been correctly identified, include,

- running HLT with unsupported ALICE magnet current settings or unsupported set of detectors,
- incorrect HLT configuration selection for given running conditions,
- lost connections to nodes,
- external services not available and
- failure of the HLT startup procedure due to software problems.

6. Performance
To get an objective view of the operational behaviour and stability of the HLT, two key measurements are studied,

- operational efficiency and
- failure fraction.

Operational efficiency expressed as a percentage is defined as the amount of time the HLT was participating in physics runs divided by the total amount of data-taking time. The failure fraction as a percentage is the number of runs that failed due to a technical fault of HLT versus the total number of physics runs taken. Only faults due to HLT are taken into account. Thus, inefficiency due to other subsystems in ALICE is not included.

The efficiency for the period 2010–2011 is shown in figure 3 and the failure fractions in figure 4. Each year can be further divided into the proton-proton (pp) running period from March to October and the crucial Pb-Pb period during November and December. The integrated values for efficiency and failure fractions are summarised for these periods in table 2.

As can be seen, operational efficiency in 2010 was significantly lower than in 2011. 2010 was treated as the commissioning period for HLT. Efficiency rates were quite erratic due to extended periods of testing, commissioning and maintaining the system. However, it is clear that the significant improvements made to HLT have borne fruit. Efficiency was brought to a stable plateau of around 95% in 2011 after the significant changes made early in 2011 from the lessons learned in 2010.
Figure 3. HLT operational efficiency per month for the period 2010–2011. Circles indicate the efficiency and triangles indicate the total time spent data-taking for each month.

Figure 4. Number of HLT failures per month for the period 2010–2011. Circles indicate the fraction of failed runs and triangles indicate the total number of runs for each month.

Table 2. Integrated operational efficiencies and failure rates.

| Year | pp Period Efficiency % | Failures % | Pb-Pb Period Efficiency % | Failures % |
|------|------------------------|------------|---------------------------|------------|
| 2010 | 63.6                   | 6.2        | 80.5                      | 11.5       |
| 2011 | 81.8                   | 7.4        | 98.0                      | 11.3       |
The failure rates on the other hand show little overall change from the integrated values. Indicating that although many problems were fixed to allow HLT to participate in more runs and improve efficiency, secondary problems that were masked before or occurred less frequently are still present. One should also note that the monitoring methods were much improved in 2011, which means that failures due to HLT were more easily spotted and assigned correctly to the failing system. This introduces a slight bias to the results. Nevertheless, in the second part of the 2011 Pb-Pb period when HLT was under the heaviest stress, the failure rates dropped to almost zero, after many of the secondary problems were also eliminated. This can be seen in figure 4, by comparing the November versus December 2011 data points. The two outlying points in March and November 2011 are related to unforeseen problems discovered after significant changes to the data-taking conditions between pp and Pb-Pb run periods. The month of November was particularly affected by changes to operational procedures and conditions of the TPC. Some of the more common or particularly problematic failure modes include,

- startup failures due to internal communication problems in HLT components,
- stuck bits in RORC registers,
- FEP and CN hardware lockups,
- failures due to incorrect HLT configuration for given running conditions and 
- backpressure asserted by HLT due to software hangs.

Of further interest is the load placed on the HLT and the performance boundary conditions, such as the maximum event rates achievable and throughput in Pb-Pb. Two special runs were performed during the 2011 Pb-Pb period, where HLT was performing data reduction of TPC raw data: A minimum bias run, where the ALICE trigger was configured to record the maximum rate of all interaction (minimum bias) events possible; and a central run, where the trigger was configured to record only the 10% most central collision events, but also with the highest rate possible.

The event-processing rate in HLT for the minimum bias and central runs are shown in figure 5. The fitted average event rates and number of events in the HLT buffers are indicated in table 3. From these parameters, one can also extract the average latency per event for the full HLT compression and triggering pipeline from the ratio: events in chain over the event rate.

Over 85% of the raw data being processed by the HLT came from the TPC. To see how the data rate is handled by the HLT, the total data rates for the minimum bias and central runs are
Table 3. Performance parameters measured for HLT during the 2011 Pb-Pb period.

| Run Type       | Event Rate (Hz) | Number of Events in Buffers | Latency (s) | Data Rate (GB/s) |
|----------------|-----------------|-----------------------------|-------------|------------------|
| Minbias        | 633.4 ± 2.3     | 464.0 ± 1.1                 | 0.733 ± 0.003 | 8.0              |
| 10% Central    | 201.6 ± 2.8     | 330.0 ± 4.6                 | 1.64 ± 0.03  | 9.2              |

Figure 6. Input data rates for the minimum bias run (top) and central run (bottom). The input rates are indicated by the difference between Out (blue line) and In (green line) bandwidths on the FEP machines connected to the TPC.

Figure 7. Output data rates for the minimum bias run (top) and central run (bottom). The output rate from HLT is indicated by the In (green line) bandwidth to the FEP machines used for HLT output that are connected back to DAQ.

These rates were all achieved during real physics data-taking, with no back pressure being asserted by the HLT system. The maximum output rate of 2.3 GB/s to DAQ was not saturating the 4 GB/s bandwidth available. Similarly, the maximum input rates from the TPC of 9.2 GB/s were also well below the 16 GB/s input bandwidth available. In addition, during technical tests performed by replaying real events uploaded to the TPC front-end electronics, HLT could process
the maximum rate deliverable by the ALICE detector electronics, without contributing to the dead time. Thus, this demonstrates that HLT is capable of easily handling the full rate coming from the detectors. The readout system of the detectors themselves was the main limiting factor.

7. Lessons from 2011
During the 2011 Pb-Pb period, some important problems have been seen that require improvements in the future. A severe limitation is that technical runs cannot reproduce the conditions of real physics runs. Technical runs are used to commission the full ALICE readout, with all of the subsystems interacting with each other. The HLT is particularly negatively affected by this, because the computational loads are not reproduced by technical runs as seen in standalone HLT tests. Technical runs typically only generate noise in the raw data, since the detector readout electronics is triggered by a random trigger and there are no collisions to record. Standalone commissioning tests for the HLT do actually reproduce the expected load, because simulated and real recorded data is replayed through the whole HLT subsystem. However, a standalone test does not involve or reproduce all the interactions with external systems, where many of the more subtle problems associated with complex interacting subsystems can arise. The only way to detect these before an actual physics run is to instead have technical runs that fully reproduce the behaviour of detectors and raw data payloads as found during actual physics runs.

To alleviate this problem, some special technical runs were made with the TPC front-end electronics specially configured to store a single heavy ion event and replay it on every received hardware trigger. Such a procedure proved to help significantly during commissioning. However, since the TPC was the only detector capable of this and only one event could be replayed at a time, this still proved to have many drawbacks. It is expected that with more flexible technical runs capable of reproducing real data-taking conditions the problems seen during the first part of 2011 Pb-Pb could have been eliminated in the commissioning phase.

Another aspect of the HLT that was not well handled was the automatic calibration procedure. During 2011, many of the online calibration parameters used for event reconstruction had to be updated manually. The existing scheme requires that pedestal and calibration information, generated during dedicated runs for this purpose, are sent to the offline conditions database. From there they are automatically fetched by the HLT to use in online reconstruction. The assumption is that these parameters and additional objects created by offline processes are always valid and up to date. However, this turns out not to be the case. Sometimes the objects are invalid and cause the online reconstruction to fail, or are not correct or delayed, leading to poor reconstruction quality.

These problems were mitigated by disabling the automatic fetching of calibration parameters from the offline conditions database and instead performing manual updates to the online parameters where required. Although this procedure works, it is itself error prone and time consuming. Future upgrades to the HLT will have to rework the scheme for handling online calibration data.

On the technical side, it was seen that the HLT cluster infrastructure needs more integration with the application software. Initially, the system was designed to have the cluster managed independently of the application software. The assumption being that the cluster should be provided as a service to the application software. Although this works in many cases, experience has shown that for easier management and monitoring of the system, both the infrastructure services and the HLT application software should be more aware of each other. For example, all subsystems should share a common resource and inventory database, allowing a unified view of the current state of the system as a whole. Care should nevertheless be taken to keep the components loosely coupled and easily replaceable, by having well defined interfaces and communication protocols.
8. Conclusion

The HLT has proven to be a valuable tool to ALICE for data reduction that can easily handle all data rates seen during 2010–2011. However, it is a complex system that interfaces with many other subsystems in ALICE. 2010 was used as a commissioning period to get the key features of HLT running. To be able to use the HLT for production data reduction in ALICE, it was necessary to fix several important problems that significantly reduced operational efficiency and stability. The major restructuring work early in 2011 has shown to be effective in significantly improving the operational efficiency as measured by the higher run participation of HLT.

Although the current system is in a much better shape, there are still some lingering problems that need to be addressed. These include better handling of online calibration for reconstruction and improved testing/commissioning procedures for ALICE as a whole, in order to spot tricky problems and bugs only seen when multiple subsystems interact with each other. These remaining problems can be eliminated with future upgrades to the HLT and subsystems.

References

[1] ALICE Collaboration 2004 Technical Design Report of the Trigger, Data Acquisition, High-Level Trigger and Control System CERN/LHCC 2003-062
[2] Erdal H et al. 2012 Monitoring the data quality of the real-time event reconstruction in the ALICE High Level Trigger JoP: Conference Series, CHEP Proceedings
[3] ALICE Collaboration 1999 Technical Design Report of the Time Projection Chamber CERN/LHCC 2000-001
[4] Steinbeck T et al. 2004 New experiences with the ALICE High Level Trigger Data Transport Framework, CHEP Proceedings
[5] Rohr D et al. 2012 ALICE HLT TPC Tracking of Pb-Pb Events on GPUs JoP: Conference Series, CHEP Proceedings
[6] Richter M et al. 2012 Data compression in ALICE by on-line track reconstruction and space point analysis JoP: Conference Series, CHEP Proceedings
[7] Neuman B and Ts’o T 1994 Kerberos: An Authentication Service for Computer Networks IEEE Communications 32 (9) p 33–38
[8] Fraunhofer File System 2012 online http://www.fhdfs.com/
[9] Ram D et al. 2012 Flexible event reconstruction software chains with the ALICE High-Level Trigger JoP: Conference Series, CHEP Proceedings
[10] Al-Fares M, Loukissas A and Vahdat A 2008 A scalable, commodity data center network architecture SIGCOMM Comput. Commun. Rev. 38 (4) p 63–74