Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

ALICE HLT Cluster operation during ALICE Run 2

To cite this article: J Lehrbach *et al* 2017 *J. Phys.: Conf. Ser.* **898** 082027

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 082027 doi:10.1088/1742-6596/898/8/082027

ALICE HLT Cluster operation during ALICE Run 2

J Lehrbach[1], M Krzewicki[1], D Rohr[1,3], H Engel[2], A Gomez Ramirez[2], V Lindenstruth[1,2]*and* D Berzano[3] for the ALICE collaboration

[1] Frankfurt Institute of Advanced Studies, Ruth-Moufang-Str. 1, 60438 Frankfurt am Main, Germany

[2] Institut für Informatik, Johann Wolfgang Goethe-Universität Frankfurt, Robert-Mayer-Str. 11-15, 60629 Frankfurt am Main, Germany

[3] European Organization for Nuclear Research (CERN), CH-1211 Geneva 23, Switzerland

E-mail: `johannes.lehrbach@cern.ch`

Abstract. ALICE (A Large Ion Collider Experiment) is one of the four major detectors located at the LHC at CERN, focusing on the study of heavy-ion collisions. The ALICE High Level Trigger (HLT) is a compute cluster which reconstructs the events and compresses the data in real-time. The data compression by the HLT is a vital part of data taking especially during the heavy-ion runs in order to be able to store the data which implies that reliability of the whole cluster is an important matter. To guarantee a consistent state among all compute nodes of the HLT cluster we have automatized the operation as much as possible. For automatic deployment of the nodes we use Foreman with locally mirrored repositories and for configuration management of the nodes we use Puppet. Important parameters like temperatures, network traffic, CPU load etc. of the nodes are monitored with Zabbix. During periods without beam the HLT cluster is used for tests and as one of the WLCG Grid sites to compute offline jobs in order to maximize the usage of our cluster. To prevent interference with normal HLT operations we separate the virtual machines running the Grid jobs from the normal HLT operation via virtual networks (VLANs). In this paper we give an overview of the ALICE HLT operation in 2016.

1. Introduction

ALICE (A Large Ion Collider Experiment, see Figure 1) [1] is one of the four large experiments at the Large Hadron Collider (LHC) at CERN (see Figure 2) focusing on the study of heavy-ion collisions. The ALICE High Level Trigger (HLT) is an essential part of the experiment data taking process especially for the higher data rates during heavy-ion collisions. The ALICE HLT is used as a real time data compression and online reconstruction facility in order to reduce the amount of data which has to be recorded. This means that the raw data of the time projection chamber (TPC) is discarded after it got processed by the HLT and only the compressed clusters are stored. With the current compression ratio of ~5.5 during 2016 pp and Pb-Pb data taking for the TPC as the main contributor to the total data volume, it is possible to store all the data the experiment delivers.

The ALICE High Level Trigger cluster consists of 180 worker nodes and 12 infrastructure servers providing different services for provisioning, monitoring, logging etc. Essential services like DHCP and DNS are provided redundantly by two different servers to prevent a failure of the whole cluster if one of these infrastructure node fails.

The cluster has its own private network with a gateway node providing login possibilities from the CERN General Purpose Network (GPN). Our cluster is also connected to the ALICE

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 082027 doi:10.1088/1742-6596/898/8/082027

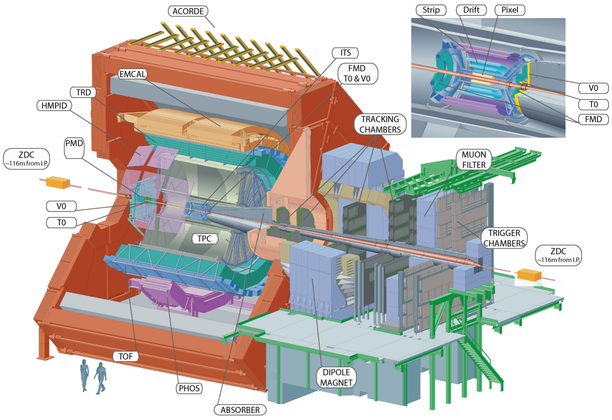

Figure 1. Schematic overview of ALICE.

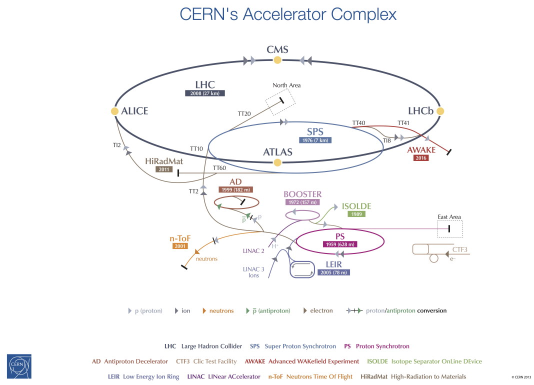

Figure 2. Schematic overview of the CERN accelerator complex.

Detector Control System (DCS) and Experiment Control System (ECS) network (see Figure 3).The ECS system controls the HLT during data taking.

All nodes have a gigabit Ethernet connection for provisioning, control, configuration, monitoring and also for the dedicated Intelligent Platform Management Interface (IPMI) network. Our data transport network for HLT operation uses Infiniband FDR with 56 Gbit/s in a fat tree topology to get the maximum bisection bandwidth between all nodes. For the data transport we use our own custom data transport framework base on the publisher subscriber model [2].

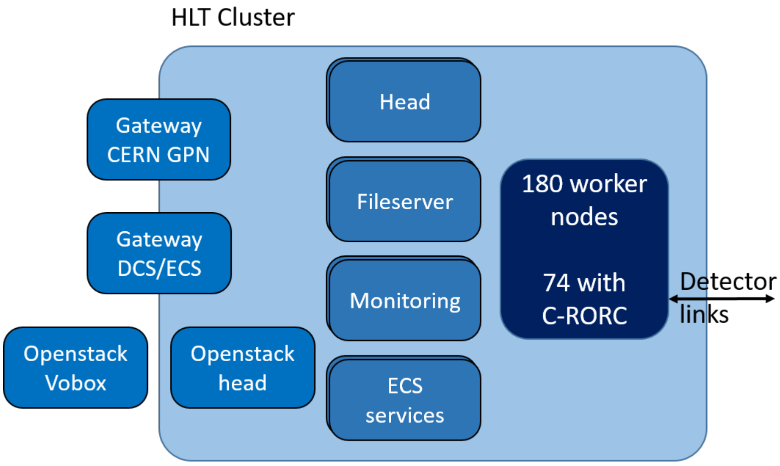

Figure 3. Schematic overview of the HLT production cluster.

All compute nodes have the same hardware, two 12 core CPUs with 128 GB RAM and one GPU. A subset of 74 of these nodes have an additional custom PCIe FPGA add-on card, the Common Read-Out Receiver Card (C-RORC) [3]. The C-RORC provides the main experimental data interface to the HLT via custom optical links, the Detector Data Links (DDL) with a link speed of 3.125 for the TPC and up to 5.3125 Gbit/s for other detectors. Up to 12 DDL links are connected to one C-RORC add-on card. The DDL links are used as the input of detector data into the HLT cluster. The FPGA of the C-RORC is also used as a first processing step for the TPC data performing cluster finding [3]. With the cluster finding offloaded to the FPGA we save significant CPU resources which would be necessary to perform this computation otherwise. This leads to a reduction of the necessary number of processing nodes.

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 082027 doi:10.1088/1742-6596/898/8/082027

The main part of the reconstruction is the TPC tracking which is an algorithm based on a cellular automaton and a Kalman filter. A considerable part of the tracking is done on the GPU [4]. Offloading the tracking to the GPU saves additional significant CPU resources leading to a reduction of the necessary compute nodes by ∼50%.

After processing the incoming data, the HLT ships its output via multiple DDL links with 5.3125 Gbit/s to Data Acquisition (DAQ). When the DAQ system receives the compressed data, the much bigger raw data can be discarded and only the compressed data is stored. This saves a large fraction of necessary tape storage space and itself enables to store all data with a maximum detector read-out rate of 6 kHz during pp and 3.75 kHz during Pb-Pb data taking. For Quality Assurance (QA) reasons 1% of the raw data is also kept.

2. Cluster administration

To ensure a consistent state of the whole cluster at any time we automatize as much as possible and reduce manual changes to a necessary minimum. The basic configuration of the operation system is managed via Foreman [5]. All servers are provisioned via Ethernet and booted via the Preboot eXecution Environment (PXE). The complete cluster can easily be rebuild in around 3 hours with only changing one flag and rebooting all servers. Installing all additional packages and configuring all servers after a fresh installation takes ∼30 minutes in addition. The complete cluster rebuild was done successfully beginning of 2016 in order to change the operating system from the deprecated Fedora 20 to the more up to date CERN CentOs 7.

On top of Foreman we use Puppet [6] to configure our servers. In this regard we use the same tool-set as CERN IT. Puppet is nicely integrated into Foreman so we can easily assign Puppet roles to each server. Servers with the same configuration are organized via host groups which makes it even easier to modify a certain group of servers according to different needs. For testing purposes and the development of new features, we have different environments in place which can be easily switched back and forth via Foreman. Each environment is matched to a git branch of the Puppet configuration repository which makes it quite easy to save everything in one git repository. Commits to our local git repository on our head node trigger an automatic import of new or modified Puppet modules to Foreman via a git hook. This way we can ensure that always the latest version is deployed on our cluster. The git repository is also synchronized to an independent Gitlab repository to guarantee a regular backup.

To ensure a consistent configuration of our servers with full version control of all installed packages we only use locally mirrored package repositories. This way we avoid unwanted updates which might break something or need a rebuild of our framework. In longer periods without data taking we usually update our repositories to profit from bug fixes and security updates. In order to test the full setup before deploying it to our production system we have build a smaller development cluster of 70 servers from older Run 1 servers which are organized in the same way as our production system. This way we can safely perform any test and ensure that our framework runs without any problems and avoid any downtime during data taking.

The C-RORC uses a custom firmware which is dependent on the detector which is connected to it. The firmware versions are stored on our local FTP server which also hosts the repositories and the deployed firmware version is controlled via Puppet. The firmware versions for all input and output links is also managed via Puppet. Our setup automatically configures all FPGAs with the correct firmware and regularly verifies the firmware version to exclude mismatches.

3. Performance and stability

The HLT cluster was upgraded in 2014 to cope with a higher readout rate of the TPC during ALICE Run 2. In the beginning of 2016 the Readout Control Unit (RCU) of the TPC was upgraded to the new RCU2 which supports a faster link speed of the DDL and also brings additional benefits for the read-out [7]. The possible data input rate for the HLT roughly doubled due to this upgrade. In order to cope with the higher data and event rates we deployed several framework improvements in 2016 [2], [8]. With data replay tests we confirmed that the HLT can cope with a total event rate of up to 6 kHz during pp collisions with 2.4 kHz TPC event replay rate, which exceeds the maximum event rate the TPC is able to deliver. For Pb-Pb we successfully tested event rates up to 3.75 kHz with 950 Hz TPC contribution which corresponds to the maximum possible TPC input capacity with the tested event sizes. We verified with these tests that the HLT can cope with all possible trigger scenarios which are planned for Run 2 also with the increased input capacity of the RCU2 [2].

After the upgrade of the TPC read-out there was a revalidation of the HLT TPC FPGA cluster finder to make sure that discarding the raw data and only storing compressed clusters does not affect the physics performance. After the validation was successful the raw TPC data was discarded in favour of the HLT compressed data in every physics run.

There was a total of 1070 hours of physics data taking after the successful validation from May to September 2016, which all included the HLT. During this time 665 distinct physics runs with HLT contribution were started. 13 out of the 665 physics runs were ended by HLT which corresponds to ∼2% end of run reasons caused by HLT. The mean time between failures was slightly above 82 hours of running. 5 runs were ended due to Hardware defects like a machine exception which trigged a reboot of a server, 5 runs were ended by software errors which were fixed shortly after they occurred the first time. The remaining 3 runs ended by HLT were due to operator requests in order to change the current configuration. The stability of the HLT during data taking was improved significantly in 2016.

The compression the HLT provides for the TPC data was improved this year as well from ∼4.3 to ∼5.5 leading to a ∼20% saving of tape space after the changes were deployed. Figure 4 shows the projection of the data to be stored to tape with the old and the improved compression scheme. The improvements of the compression are also a first step for the Run 3 developments where even higher compression factors are needed in order to store the expected data rate.

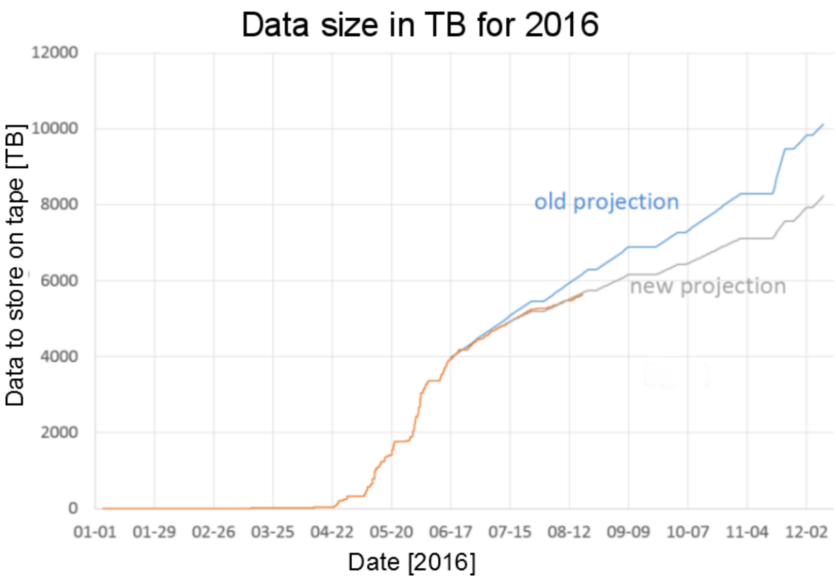

Figure 4. Projection of the ALICE data written to storage in 2016 with the old and the improved compression scheme.

In addition we improved the configuration and start-up procedure for the HLT in order to reduce the time to start data taking during stable beam conditions. The HLT was the slowest system to configure in 2015 and we managed to improve this time significantly so we are now in the shadow of the detector configuration and no longer prolong the start-up time of data taking [2].

4. Monitoring

From the operation point of view it is important to know at all time in which state the cluster currently is and if everything is still OK. This concerns the running state of every server as well as the entire system health including the network. In order to guarantee good running conditions at all times we use Zabbix to monitor the health of our servers via different metrics. Currently there are about 100 metrics per server monitored like temperature, CPU load, network traffic, free disk space etc. For several critical metrics there are email notifications in place to notify the HLT operations crew of possible problems. We also included a daily status report as email notification which summarizes all alarms and metrics which might indicate problems of our clusters. This includes error counters for the Infiniband network and the output of several node sanity checks if these are not passed properly. In many cases we could tackle possible problems before they actually interfered with data taking for example by excluding a server which shows a problematic behaviour.

Some critical metrics like CPU temperature are monitored in a redundant fashion via the operating system with the Zabbix daemon and additionally via IPMI requests to provide these values even if a server is not responding any more. In addition there are automatic shutdown procedures in place which get triggered e.g. in case of critical temperature values in order to prevent damage to the servers.

During the year we also improved the monitoring of our framework and added several new logging information and error outputs like too many events in the processing chain. There are several filters in place to scan for possible problems and immediately report these to the central experiment logging system. This improved the available information for the ALICE shifter controlling the data taking process in case of any HLT related problems. In addition there are also triggers to send email notifications to the HLT operations crew in case of problems during data taking. These logging and notification improvements are also available for detector code running inside the HLT. We are using this new features to send email notifications to detector teams in case of detector errors or corruption of the incoming detector data streams are found by our framework.

5. Operating as a Worldwide LHC Computing Grid (WLCG) site

In order to utilize our clusters as much as possible we can run as a WLCG Grid site. The WLCG setup was developed in cooperation with the ALICE Offline group. In 2016 we used OpenStack Virtual Machines (VMs) to compute ALICE Monte Carlo simulations during technical stops and longer LHC machine development phases. In order to avoid any interference with our configuration for normal operation we restricted the network access of the VMs as much as possible. For network separation we use VLAN tagging configured at switch level to avoid any Grid communication in our HLT internal networks (see Figure 5). The OpenStack VMs have an own virtual network interface on top of the physical interface. We deployed IP table rules for the virtual network interface on all hosts to allow only communication between the VM gateway and the VMs. All other network traffic originating from the virtual interface to our networks is discarded. In order to have full control of the Grid operation the OpenStack services are disabled by default and have to be started manually via management scripts. Due to the limited I/O possibilities and the restricted storage capabilities our cluster is running exclusively centrally-managed Monte Carlo productions with high CPU utilisation and reduced I/O load so it is not used for all Grid jobs.

Figure 6 shows the amount of compute time we were able to contribute to the ALICE Grid jobs until September 2016. Our development cluster is donating its compute resources towards the Grid operation as long as it's not used for development. The old Run 1 servers have less resources available so fewer VMs can be spawned due to a lack of available CPU cores and RAM limitations. The production cluster was only active during technical stops but contributed a

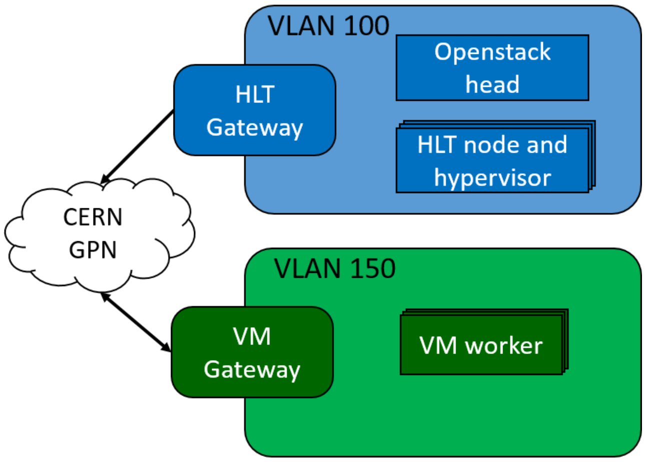

Figure 5. Schematic overview of the HLT WLCG network setup.

larger fraction to the overall compute resources since there are more nodes available with more up-to-date hardware which are able to compute much faster. The start-up of the OpenStack VMs via our management scripts only takes 5 minutes for the whole cluster if the image which is started is cached locally on each server. The jobs on the other hand are rather long running jobs with a timeout of 24 hours or more. Since ALICE jobs do not provide any form of suspension, we can only run efficiently during longer data taking interruptions and technical stops. With the opportunistic usage of our clusters we could contribute ∼2.5% of the ALICE Monte Carlo simulation jobs. For the future we plan to extend the periods of running Grid jobs even further and are considering to dedicate a fraction of the production cluster even during proton-proton data taking when not all of the compute resources are needed.

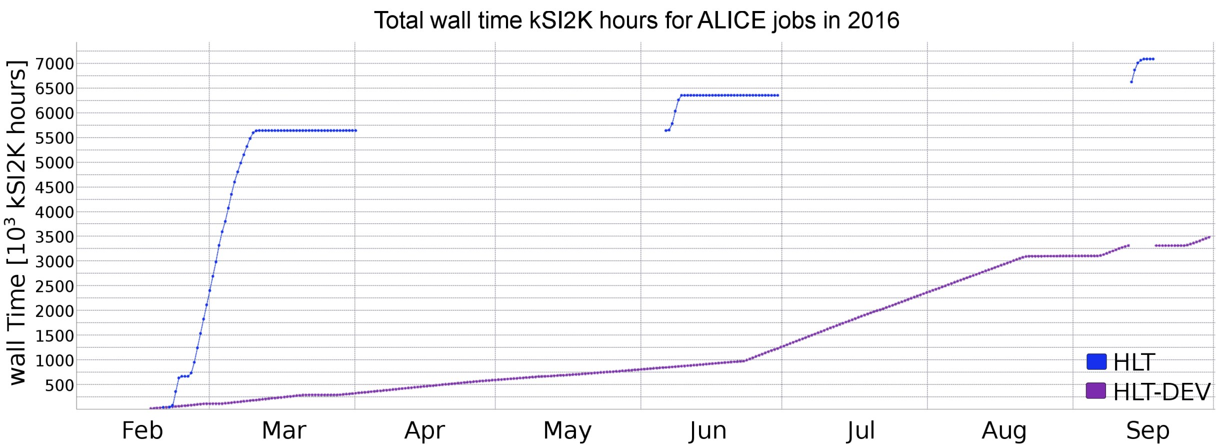

Figure 6. Summary of the opportunistic usage of our compute resources to compute ALICE Monte Carlo simulation jobs in 10^3 kSI2K hours

6. Conclusion and outlook

The ALICE HLT is an essential part of ALICE data taking enabling the experiment to store all data due to sufficient data compression. In 2016, the HLT was only causing 2% of the end of runs reasons during physics data taking until end of September. There were significant improvements in 2016 to reduce bottlenecks in order to increase the running efficiency of the HLT for example for the start and configuration time. With the framework improvements the

HLT is capable to cope with all planed ALICE data taking scenarios during Run 2 and imposes no limitation for any trigger scenario. During longer interruptions of data taking the HLT is used as a Grid site and is donating its compute resources to ALICE Monte Carlo jobs to support the collaboration with additional analysis computing power. This is already a considerable amount of compute time for simulations which we plan to increase even further during the extended year end technical stop of the LHC. In order to improve the opportunistic setup even further we are currently moving to a Docker container-based approach by running one job per container. This experimental approach is based on a tool called Plancton [9] and first tests are showing that it is providing the required level of isolation with more scheduling granularity and much less management efforts.

References

[1] ALICE collaboration, *The ALICE Experiment at the CERN Large Hadron Collider*, JINST, 3(08):S08002,2008
[2] Rohr D et al, *Improvements of the ALICE HLT data transport framework for LHC Run 2*, JPCS, 2017
[3] Engel H et al, *FPGA based data processing in the ALICE High Level Trigger in LHC Run 2*, JPCS, 2017
[4] Rohr D et al, *GPU-accelerated track reconstruction in the ALICE High Level Trigger*, JPCS, 2017
[5] *theForeman.org*
[6] *Puppet.com*
[7] Alme J et al, *RCU2 - The ALICE TPC readout electronics consolidation for Run 2*, JINST, 8:C12032, 2013
[8] Krzewicki M et al, *Alice HLT Run2 performance overview*, JPCS, 2017
[9] Concas M et al, *Plancton an opportunistic computing project based on Docker containers*, JPCS, 2017