EPJ Web of Conferences **214**, 08027 (2019) https://doi.org/10.1051/epjconf/201921408027
CHEP 2018

The ALICE Analysis Facility Prototype at GSI

Kilian Schwarz[1,*], *Soeren* Fleischer[1], *Raffaele* Grosso[1], *Jan* Knedlik[1], *Thorsten* Kollegger[1], and *Paul* Kramp[1]

[1]GSI Helmholtzzentrum für Schwerionenforschung GmbH, Planckstr. 1, 64291 Darmstadt, Germany

Abstract. In LHC Run 3 the ALICE Computing Model will change. The Grid Tiers will to a large extend be specialised for a given role. 2/3 of the reconstruction and calibration will be done at the combined online and offline O2 compute facility, 1/3 will be done by the Tier1 centres in the Grid. Additionally all Tier1 centres, as already now, will take care of archiving one copy of the raw data on tape. The Tier2 centres will only do simulation. The AODs will will be collected on specialised Analysis Facilities which shall be capable of processing 10 PB of data within 24 hours. A prototype of such an Analysis Facility has been set up at GSI based on the experiences with the local ALICE Tier2 centre which has been in production since 2002. The main components are a general purpose HPC cluster with a mounted cluster file system enhanced by Grid components like an XRootD based Storage Element and an interface for being able to receive and run dedicated Grid jobs on the Analysis Facility prototype. The necessary I/O speed as well as easy local data access is facilitated by self developed XRootD PlugIns. Performance tests with real life ALICE analysis trains suggest that the target throughput rate can be achieved.

1 Introduction

An upgrade of the ALICE [1] detector is currently prepared for the Run 3 period of the Large Hadron Collider (LHC) [2] at CERN starting in 2020. The physics topics under study by ALICE during this period will require the inspection of all collisions at a rate of 50 kHz for minimum bias Pb-Pb and 200 kHz for pp and p-Pb collisions in order to extract physics signals embedded into a large background. The upgraded ALICE detector will produce more than 1 TByte/s of data. Both collision and data rate impose new challenges onto the detector readout and compute system. Therefore in LHC Run 3 the ALICE computing model will change with the idea to minimise data movement and to optimise processing efficiency.

In Run 2 reconstruction takes place at the Tier0 and Tier1 centres. The Tier1 centres additionally are responsible of keeping together another copy of the raw data (the first copy is at CERN) and storing them on tape as long term archive. Organised analysis jobs run on Tier1 and Tier2 centres while individual user analysis and Monte Carlo production is usually done on Tier2 centres. In LHC Run 3 however the Grid Tiers will to a large extend be specialised for a given role. Keeping a copy of the raw data and archiving them on tape will stay a responsibility of the Tier1 centres. But only 1/3 of the in total significantly increased reconstruction and calibration work will be done on the Tier1 centres. The challenge of dealing

---

[*] e-mail: k.schwarz@gsi.de

with 2/3 of that work will be met by a combined online and offline compute facility developed and managed by the ALICE O2 project [3]. The Tier2 centres and opportunistic resources as HPC centres and Cloud resources will take care of MonteCarlo production (simulation). The Analysis Object Data (AODs) will be collected on specialised Analysis Facilities which shall be capable of processing 10 PB of data within a time scale of one day (which corresponds to an analysis throughput of about 115 GB/s). A schematic view of the distribution tasks among the participating Tier centres can be seen in fig 1.

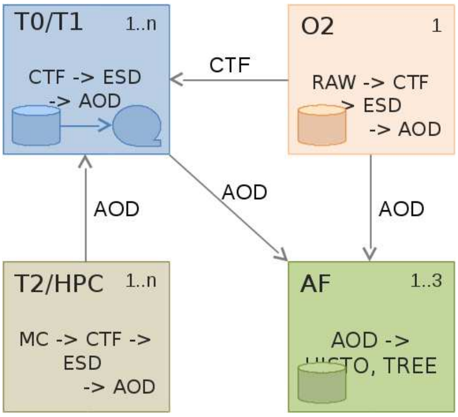

Figure 1. A schematic view of the distribution of tasks among the ALICE Tier centres in LHC run 3. The fig. has been taken from P. Buncic, ALICE T1T2 Workshop, Strasbourg, May 2017.

A prototype of such an Analysis Facility has been set up at the GSI Helmholtzzentrum für Schwerionenforschung (GSI) in Darmstadt, Germany. The design has been inspired by the experiences gained with the local ALICE Tier2 centre [4] which has been in production since 2002.

2 Setup of the ALICE Analysis Facility prototype

In the core of the ALICE Analysis Facility Prototype at GSI is the general purpose HPC cluster which provides 41000 logical cores compute capacity. The cluster is jointly used by users of the local GSI and FAIR (Facility for Antiproton and Ion Research) experiments, Theory groups, as well as local ALICE groups, the ALICE Tier2 centre and now the Analysis Facility prototype. The current production cluster, Kronos, uses the CPU models Intel Xeon E5-2660 v3, Intel Xeon E5-2680 v4, and AMD EPYC 7551. All worker nodes have a direct mount of the shared Cluster file system Lustre which provides a total storage capacity of currently 25 PB and is also commonly used by the user community mentioned above.

In order to provide global access to the data stored at the GSI Analysis Facility Prototype the core infrastructure is complemented by a Grid [5] Storage element which consists of a set of XRootD [6] daemons running on top of the Lustre file system. The main elements are an XRootD redirector in combination with an XRootD data server. The redirector of the storage element is using the split directive of XRootD and redirects external clients to the external interfaces of the XRootD data server machine and internal clients to the internal interface which is directly connected to the local Infiniband Cluster. For redundancy and high availability reasons it is planned to add lateron a second redirector as well as additonal data servers following the example of the setup for the ALICE Tier2 centre at GSI. Additionally two XRootD forward proxy servers are being provided which are used commonly with the Tier2 centre. They provide the possibility to jobs running inside the protected HPC environment to read input data from external data sources using the proxy interface.

Jobs can arrive at the ALICE Analysis Facility Prototype either directly via a local submit node for testing purposes or centrally managed from CERN via a so called "vobox". The AliEn [7] Grid services running on the vobox provide an interface to the global ALICE Grid infrastructure and especially to the central Grid services running at CERN. The jobs at the Analysis Facility Prototype run inside of Singularity [8] containers. This makes it possible to run ALICE Grid jobs smoothly in their standard Scientific Linux environment on top of the Debian based host system of the local HPC cluster.

A schematic view of the setup of the ALICE Analysis Facility Prototype at GSI is shown in fig. 2.

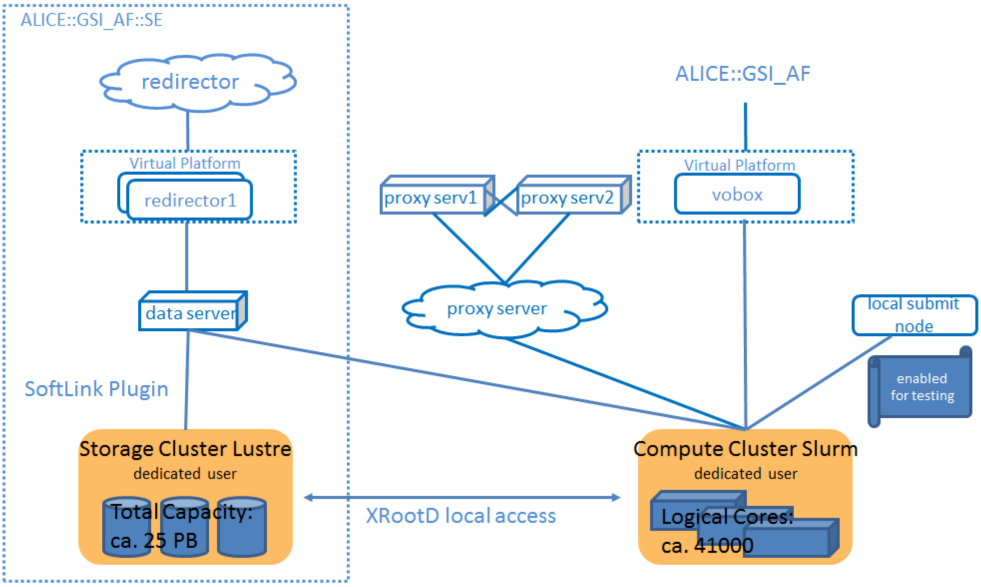

Figure 2. A schematic view of the setup of the ALICE Analysis Facility prototype at GSI. External clients would access the data stored on the GSI Lustre cluster via the XRootD services of the storage element.

As initial resources the ALICE Analysis Facility prototype at GSI will start with a disk space of 600 TB and job slots being taken from the ALICE Tier2 allocation. AOD files of an initial full data set of 2015 Pb Pb collisions (about 250 TB) have been transferred to the Analysis Facility prototype storage element for testing purposes.

3 XRootD PlugIns for the Analysis Facility Prototype

In order to optimise the I/O throughput of the analysis jobs, namely the access to data stored on the local Lustre cluster through XRootD data servers, several XRootD PlugIns have been developed at GSI. Two of these PlugIn based solutions are essential for the prototype setup presented here.

The first solution, the Symlink PlugIn, facilitates local data access without the need to interact with the global Grid infrastructure in order to be able to find and access the data stored on the Lustre file system at GSI. Data which are stored in Grid storage elements within the AliEn system, and also the storage element of the Analysis Facility prototype is enbedded in the ALICE environment, are registered in the file catalogue with a global unique identifier which is mapped to the human readable logical file name the user gave to the file. On a local storage system the file again is stored with an XRootD URL containing the hash based global unique identifier. So a local user who would like to analyse the data, even if the user has direct access to the Lustre Clustre to which XRootD at GSI stored the file, would have to ask the AliEn File catalogue first in order to get the human readable logical file name of the data the user wants to analyse. This can be circumvented by using the Symlink PlugIn which creates symbolic links in a different directory mapping the logical file names (LFNs) to the physical file names (GUIDs) on the storage system. The data can thus be accessed directly without the need to communicate with the external Grid services first. This is an advantage when doing efficient and fast local data analysis on the Analysis Facility prototype.

The symbolic links are maintained in the ALICE software package XrdAliceTokenAcc which is a standard XRootD Acc (access control , i.e. authorisation) library PlugIn. In this context it is checked whether or not the user/host is permitted access to a given path for the specified operation. The symbolic links are created during file access authorisation while checking for read or write access. An additional functionality for removing the links again has been added during file removal while checking for delete access.

The second solution is the RedirLocal PlugIn which works on the name space of the physical file names, the original XRootD URLs. It is an ofs.cmslib PlugIn which alters the redirector's request handling behaviour. Cmslib is a shared library that contains an implementation of the cluster management client interface that the Open File System (ofs) component of XRootD uses for handling file system specific operations (e.g., open, close, read, write, rename, etc). An XRootD redirector server may load this PlugIn in order to redirect clients to locally available files, if both client and redirection target are inside a private network, as this guarantees availability of the required file at the local shared file system. Cooperation with the XrootD core development team resulted in the integration of necessary client and server side changes into the XRootD base code of XRootD versions newer than 4.8.0. For redirections to become effective it is necessary that clients make use of the client API of XRootD version 4. Motivation for this PlugIn is that a single XRootD data server can only provide limited I/O bandwidth. Moreover in the GSI setup that XRootD data servers read and write via the local Lustre Cluster file system accessing files via XrootD data servers would double the network traffic inside the infiniband network. This is, especially with a limited number of XRootD servers, a bottleneck in CPU and bandwidth for the setup of the ALICE Analysis Facility prototype. Clients at GSI which can read files to be analysed directly from the local Lustre file system while circumventing XRootD data servers can make direct use of the aggregated bandwidth of all Lustre file servers envolved which sums up to 3.46 Tb/s for the current production system "Nyx". This is an important ingredient in being able to achieve the target rate of 10 PB analysis throughput within 24 hours. The advantages of using the RedirLocal PlugIn are displayed in fig 3.

An issue is still the complex interplay between local (Lustre) and XRootD file rights which is why the RedirLocal PlugIn is in the current setup of the ALICE Analysis Facility prototype only activated for read access.

In order to be able to make use of the RedirLocal XRootD PlugIn as described above for doing high performance data analysis at the GSI ALICE Analysis Facility prototype it is essential that a redirection to local files works also from within ROOT [9]. ROOT is a C++ based framework used by the ALICE experiment for data analysis and a typical ALICE analysis job would usually be a job executing a ROOT based analysis and thus data access would also usually be from inside ROOT. ROOT provides system-independent binary files

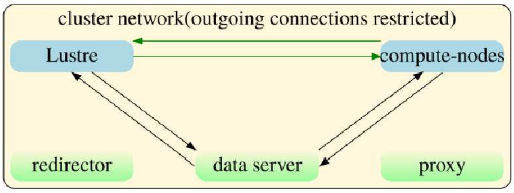

Figure 3. This figure shows how data access with and without the usage of the RedirLocal PlugIn works. Without the PlugIn a client in the GSI batch farm would be redirected to the local XrootD data server which in turn would read the file from the local Lustre Cluster thus doubling the network traffic. When using the RedirLocal PlugIn a client would be redirected directly to the file as it is stored on the Lustre file system. This way the client can make use of the full bandwidth of all file servers envolved in the GSI Lustre Cluster setup.

in which the user can store objects of any class having a dictionary. Information stored into a ROOT file can be organized in several subfolders, and the user can navigate it as he were browsing the file system of his operative system. User data are usually compressed. Event data are usually stored in Trees which are optimized for very high volumes and they are split in branches by default when saved into ROOT files. Standard ROOT Files are usually opened using the class TFile. If a user wants to open remote files from XRootD servers then the class TNetFile which inherits from TFile is being used. A TNetFile is like a normal TFile except that it reads and writes its data via an XRootD server. TXNetFile, inherting from TNetFile, again is an extension of TNetFile able to deal with new versions of XRootD servers. A new version of this class, TNetXNGFile which inherits directly from TFile, enables access to XRootD files using the new XRootD client for XRootD version 4 or larger. As mentioned above, due to new code components in the XRootD base code, for redirections to become effective it is necessary that clients make use of the client API of the new XRootD version. Therefore using the RedirLocal XRootD Plugin via a ROOT client works only when the class TNetXNGFile is being used and when ROOT has been compiled against an XRootD version larger than 4.8.0.

But the RedirLocal PlugIn does not need to work only with standalone ROOT clients, it also needs to work when ROOT analysis jobs arrive at the ALICE Analysis Facility prototype via Grid methods and file access takes place from within the Grid using Grid methods, especially when files are being accessed which are registered in the Grid File catalogue. The abstract base class defining an interface to common GRID services from inside ROOT is TGrid. A concrete implementation for the ALICE Grid environment AliEn is the class TAlien. In case a user wants to access files registered in AliEn TAlienFile is being used. A TAlienFile is like a normal TFile except that it reads and writes it's data via TXNetFile and gets authorization and the TXNetFile URL from an AliEn service. And this was already the problem. A patch had to be provided that makes TAlienFile to inherit from TNetXNGFile instead of using the older TXNetFile. This way also TAlienFile can make use of the new XRootD client and the RedirLocal XRootD PlugIn works also together with the AliEn Grid. Additionally a few smaller issues had to be added, for example it was necessary to add an LFN parameter to the constructor of TNetXNGFile in order to be able to pass the logical file name to TFile/TArchiveFile (as it is already the case in TXNetFile) and in case a file on the local file system (or Lustre) wants to be accessed the file server needed to be set to "localhost" which prevents that XRootD data servers are being queried,

Details of the XRootD PlugIns are described in the article "XRootD plug-in based solutions for site specific requirements" by J. Knedlik et. al. in this conference proceedings.

4 Performance tests

In order to investigate whether the target rate of an analysis throughput of 10 PB in 24 hours is achievable at the GSI ALICE Analysis Facility prototype a series of performance tests with ALICE analysis trains analysing data stored at the GSI Lustre clustre have been done. For this one has to understand how the Lustre File system works. Clients read and write files sending a request to the Metadata server (MDS). Using its metadata the MDS determines the location of a file and redirects the client to the Object Storage Server (OSS) which manages the Object Storage Targets (OST) storing the requested file objects. Each OST contains a number of binary objects representing the data for files in Lustre. Files in Lustre are composed of one or more OST objects, in addition to the metadata inode stored on the MDS. The allocation of objects to a file is referred to as the file layout (previously striping), and is determined when the file is created. The test Cluster used for the performance tests described in this article has a capacity of 8.2 PB and consists of 30 OSS in total where each OSS manages a number of 7 OSTs.

The tests were done in the way that local ALICE analysis trains with a simple I/O limited analysis task were submitted to the GSI batch farm with varying number of jobs. Analysis trains have been originally developed for the use in the Grid in order to optimise the concurrent analysis of big datasets by hundreds of users. They are optimised for high throughput and short turn-around times. The system combines several analysis tasks (also from different users) in so-called analysis trains which are then executed within the same batch jobs thereby reducing the number of times the data needs to be read from the storage systems. The Lustre partition used for the tests have been reserved for exclusive use. As analysis software ROOT version v5-34-30-alice and AliRoot (the ALICE analysis framework built on top of the ROOT framework) version v5-09-32 have been used. In order to achieve consistency only jobs which started within the first minute are being considered. Plots are generated when 98% of these jobs are finished.

The tests have been done in a threefolded way. First tests with a single OST have been done with an increasing number of jobs per train. Here only data from a single OST have been read and analysed. Then tests with a single OSS have been done with an increasing number of jobs per train. Here only data from a single OSS have been read and analysed. Finally a big test has been done using the full test cluster available. Here the data reading has been equally distributed among the 30 OSS in the test system. The last test was limited by the number of worker nodes which have access to the test storage cluster, therefore it was not possible to do this test with more than 2500 concurrent data reading analysis jobs. During the performance tests the jobs were reading directly from Lustre but the overhead introduced by XRootD and its PlugIns has been measured on a smaller scale and has been considered negligible.

The first test, reading from a single OST with an increasing number of concurrent jobs resulted in a maximum I/O rate of 440 MB/s which has been achieved with a number of 40 jobs per train. The second test, reading from a single OSS with an increasing number of concurrent jobs resulted in a maximum I/O rate of 2100 MB/s which has been achieved with a number of 200 jobs per train (see fig. 4). Both tests are in agreement with each other and show that with the current system an analysis throughput of about 11 MB/s per job can be achieved.

The large scaling test using the full test environment of a Lustre Cluster consisting of 30 OSS with a total storage capacity of 8.2 PB and a compute farm with 2500 cores for running jobs exclusively resulted in the fact that no scaling limit has been found. As can be seen in fig. 5 reading and analysing data with an increasing number of concurrent jobs with data reading equally distributed among the OSS shows a perfectly linear scaling behaviour

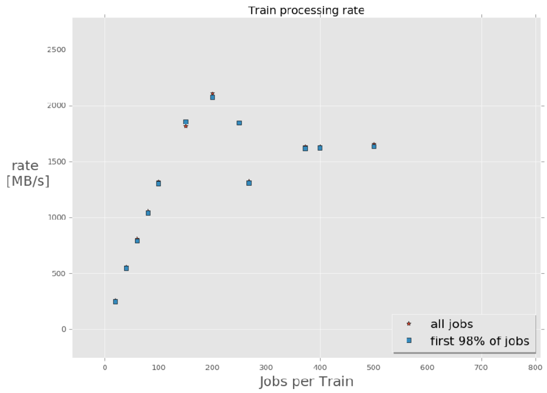

Figure 4. This figure displays the result of the performance test with a single OSS. Analysis trains with varying numbers of jobs were submitted to the GSI batch farm reading and analysing data from a single OSS. It can be seen that a maximum analysis throughput of about 2100 MB/s has been achieved with 200 concurrent analysis jobs.

with respect to analysis throughput in MB/s over number of concurrent analysis jobs. As in fig. 4 the red stars are the values measured using all jobs while the blue squares have been produced as soon as 98% of all jobs have been finished. As maximum I/O throughput a value of 32 GB/s has been measured, though, with a number of 2500 concurrent analysis jobs per train. This has been due to the fact that the maximum number of cores for running jobs in the test environment has been limited to 2500. This result of about 12.5 MB/s per analysis job corresponds well with the above measurements using a single OST and a single OSS. It shows that the desired target rate of an analysis throughput of 10 PB in 24 hours or 115 GB/s should be achievable by scaling the number of jobs and OSS accordingly. Extrapolating the measured numbers suggest that 9000 concurrent analysis jobs reading from a Lustre Cluster with 50 OSS should be sufficient in order to be able to achieve the desired numbers.

5 summary and conclusion

A working prototype of an ALICE Analysis Facility which will play an integral part in the computing model of ALICE in LHC Run 3 has been setup at GSI already now. Key solutions which are needed in order to achieve the desired analysis throughput have been implemented using XRootD PlugIns. Performance tests using standard ALICE analysis trains with the current test setup suggest that the target rate to be able to analyse 10 PB of data within a timescale of 24 hours should be achievable at the ALICE Analysis Facility prototype at GSI. Further improvements of the current setup including scaling to production size are on the way.

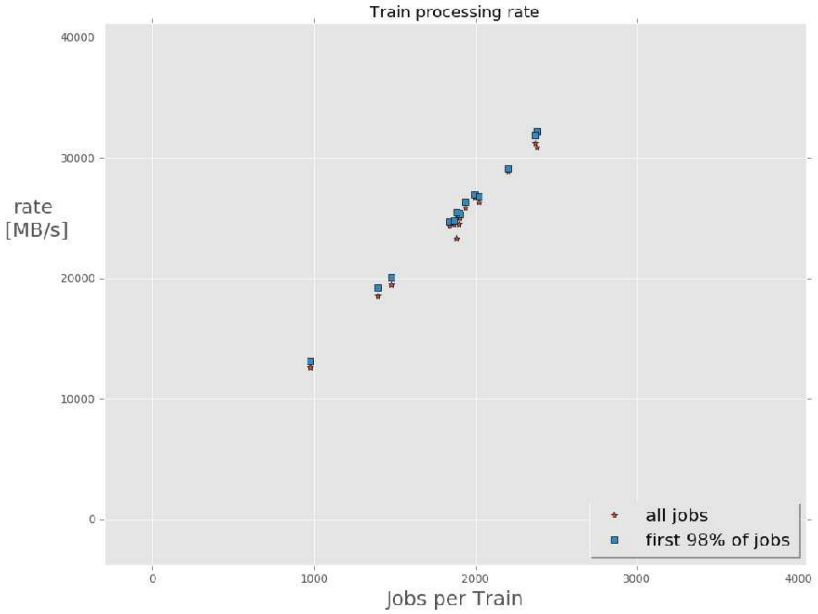

Figure 5. This figure displays the result of the performance test making use of the full test environment (Lustre file system consisting of 30 OSS providing a total storage capacity of 8.2 PB). Analysis trains with varying numbers of jobs were submitted to the GSI batch farm reading and analysing data from the test system with data reading equally distributed among the OSS. It can be seen that within the limits of the test environment (a maximum of 2500 concurrent jobs) the scaling behaviour is perfectly linear and no limitating factor can be seen.

References

[1] ALICE Collaboration, *The ALICE experiment at the CERN LHC*, JINST **3** S08002 (2008)
[2] L. Evans(ed.) and P. Bryant(ed.), *The CERN Large Hadron Collider: Accelerator and Experiments*, JINST **3** S08001 (2008)
[3] A. Ananya et al, J. Phys.: Conf. Ser. **513** 012037 (2014)
[4] K. Schwarz et. al., J. Phys.: Conf. Ser. **331** 052018 (2011)
[5] I. Foster, C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure* (Morgan Kaufmann, 2014)
[6] A. Dorigo etl. al., WSEAS Transactions on Computers 4(**4**):348-353 (2005)
[7] P. Saiz et. al., Nucl. Instrum. Meth. **A502** 437-440 (2003)
[8] G. Kurtzer, doi: 10.1371/journal.pone.0177459
[9] I. Antcheva et. al., https://doi.org/10.1016/j.cpc.2009.08.005