TAG Based Skimming In ATLAS

TAG Based Skimming In ATLAS

T Doherty[1], J Cranshaw[2], J Hrivnac[3], M Slater[4], M Nowak[5], D Quilty[1] and QZhang[2] for the ATLAS collaboration

1 University of Glasgow, Glasgow G12 8QQ, UK.
2 Argonne National Laboratory, 9700 S. Cass Avenue, Argonne IL 60439, USA.
3 LAL, Univ Paris-Sud, IN2P3/CNRS, Orsay, France.
4 University of Birmingham, Edgbaston, Birmingham, UK.
5 Brookhaven National Laboratory, Upton, NY 11973, USA.

E-Mail: thomas.doherty@glasgow.ac.uk

Abstract: The ATLAS detector at the LHC takes data at 200-500 Hz for several months per year accumulating billions of events for hundreds of physics analyses. TAGs are event-level metadata allowing a quick search for interesting events based on selection criteria defined by the user. They are stored in a file-based format as well as in relational databases. The overall TAG system architecture encompasses a range of interconnected services that provide functionality for the required use cases such as event selection, display, extraction and skimming. Skimming can be used to navigate to any of the pre-TAG data products. The services described in this paper address use cases that range in scale from selecting a handful of interesting events for an analysis specific study to creating physics working group samples on the ATLAS production system. This paper will focus on the workflow aspects involved in creating pre and post TAG data products from a TAG selection using the Grid in the context of the overall TAG system architecture. The emphasis will be on the range of demands that the implemented use cases place on these workflows and on the infrastructure. The tradeoffs of various workflow strategies will be discussed including scalability issues and other concerns that occur when integrating with data management and production systems.

1. Introduction

The ability to navigate to the subset of events relevant to a given physics analysis can improve the efficiency of the ATLAS Event Store. To enable this selection event-level metadata (TAGs) are collected. These event-level metadata contain key quantities about events and references to the events in upstream data products. TAGs are thus a means to decide which events are of interest to a specific analysis without retrieving and opening the files that contain them, and to efficiently retrieve exactly the events of interest, and no others [1]. The process of accessing these events and the demands the varying use cases make on the surrounding architecture is explored in this paper.

This paper first gives a brief description of the ATLAS data store itself in section 2, then of the TAG services architecture in section 3. The specifics of the Skimming service architecture are discussed in section 4. The primary use cases implemented are discussed in section 5, and the paper finishes with a conclusion.

2. ATLAS Event Data products

2.1. Event Data

ATLAS data are processed in stages which produce progressively more refined and filtered data products [2]. The data are divided into streams at the first stage, RAW, based on groups of triggers for different physics signatures, e.g.. Egamma, Jet, Muon, etc. These streams are allowed to overlap. Further data products called Event Summary Data (ESD) and Analysis Object Data (AOD) [3] are also produced as

Published under licence by IOP Publishing Ltd

managed productions. ESD are calibrated and clustered detector data (1 MB/event) The AOD are reconstructed physics objects such as electrons, photons, jets, etc., (200 kB/event). Beyond the online trigger selection none of these data products has any further filtering applied.

Production of these data products requires the immense computing and storage resources provided by the computing grids used by ATLAS: LHC Computing Grid (LCG), the Nordic DataGrid Facility (NDGF), and the Open Science Grid (OSG). The data is consequently grid-resident and must be accessed using grid tools.

ATLAS manages this grid-resident data using a system called Distributed Data Management [4] which can be accessed using the DQ2 tools [5] provided by ATLAS. This system deals exclusively with files which are grouped into chunks called datasets. Datasets are transferred between sites using a mechanism of subscriptions. Each participating site has a set of site services running, and when a site is subscribed to a particular dataset, the site services are responsible for transferring that dataset and keeping it up-to-date should new files be added. A set of central catalogues keeps track of the files in each dataset, the dataset locations, identifiers and subscriptions.
All of the primary data products such as RAW, ESD, and AOD are managed by this system.

The Production ANd Distributed Analysis (PanDA) [6] system is the workload management system for production and distributed analysis processing used within ATLAS. PanDA is designed to match the data-driven, dataset-based nature of the overall ATLAS computing organization and workflow. With data-driven meaning the production or analysis job is run where the data is stored. It has a simple client interface that allows easy integration with diverse front ends for job submission. It uses pilot [7] jobs for acquisition of processing resources.

2.2 Event Level Metadata (TAGs)

The ATLAS Computing Model [8] describes an Event Level Metadata system, or TAG Database [9]. The role of the TAG Database is to support seamless discovery, identification, selection and retrieval of ATLAS event data held in the multi-petabyte distributed ATLAS Event Store.

As events are selected by the online ATLAS Computing system, event data is written and stored in increasingly reduced formats. TAGs are constructed using physics objects in the AOD. At 1kB per event, an event tag is the most concise event data format used by ATLAS. Each event TAG contains attributes defined by the Physics Analysis Groups and TAG Database development group. The attributes are chosen on account of their potential to support selection of events and navigation within the system.

The content of the TAGs can be divided into five types of attribute.

1. *Event quantities* - attributes that apply to the whole event, such as run number, event number, luminosity and so on
2. *Physics objects* - electrons, muons, photons, taus, jets and their attributes
3. *Physics or Performance Group attributes* - space for each physics group to define its own attributes
4. *Trigger information* - for both low and high-level triggers.

5. *Pointers to event data* - navigational references to the data in the AOD, ESD and RAW files.

The fifth type of attribute, the pointers, takes the form of a token which contains information on the file containing the event data (in the form of a GUID) and the location of that event within that file, as well as other information for object retrieval. One can therefore ask for exact locations of events or simply which files contain those events. The following will discuss the workflows and architectural dependencies involved in retrieving the data products referenced in these pointers for various event selections.

TAGs are stored in both files and relational databases. As event metadata is created using AOD event data, TAGs are written to ROOT [10] files during processing on the Grid. These files are then used to populate a distributed set of relational databases with the full TAG data. TAGs use the LCG POOL Collections [11] data format which places a storage independence layer between the TAG tools and the data. This allows us to use TAG data stored in files on the grid or in databases transparently. This proves to be very useful for the skimming workflow architecture.

The distribution model for file-based TAGs follows that of the AOD data. The population of the relational databases is done by transferring these files back to CERN and running a process which loads the data into Oracle databases. Several sites have volunteered to participate in hosting these databases: CERN, DESY, RAL, TRIUMF, PIC, et. al.. Although currently a full copy of the TAG data is stored at CERN, the TAG service infrastructure does not require that the full copy of the data be available at a single site or server. A query engine called ELSSI has been developed to query the relational TAG data.

3. TAG Services Architecture

The overall Event Level Metadata system architecture encompasses a range of interconnected services that provide functionality for the main TAG use cases - which can be summarised as follows:

1. Data Discovery [12]
2. Query Automatization and Monitoring
3. Data Extraction (Skimming)

The third domain in this summary is the focus of this paper but it is important to understand how the TAG services integrate and work together.

3.1 Data Discovery

The Event Level Selection Service Interface (ELSSI) brings together all of the metadata needed to do event-level selections in a manner which allows users to query metadata about the full ATLAS event store. This tool brings together a variety of metadata services, including run and luminosity block level metadata.

Using a php-based web tool the user is stepped through the selection criteria. Starting with temporal cut (run range), streams, triggers and finally physics attribute cuts. Examples of activities supported by this tool are the following:

a) Explore trigger-stream correlations.
b) Check the effect of applying data quality selections

c) Count the events that meet the selection criteria.
d) Display any event-wise metadata for the events meeting the selection criteria in tabular or histogram (1-D plot) formats.
e) Find datasets and files for the selected events.
f) Skim the selected events using the Skimming Service.

3.2 Query Automatisation and Monitoring

In addition to interactive exploration of the Event Store using ELSSI, the TAG data is used for data monitoring as well. Also the result of the interactive exploration could be a query which needs to be run regularly. Individual tools and web services have been developed as needed to satisfy these needs.

3.3 Data Extraction (Skimming)

After going through the query creation step, the user can then ask to see the events in the Display tab or proceed to obtain the ROOT files using the Extract functionality. The extraction functionality is provided by a web service that produces a ROOT file with the complete TAG records for the selected events using POOL utilities wrapped in python. The user can retrieve the ROOT file on the AFS [13] file system or via a web link. The extraction output can be used as input to an ATLAS reconstruction framework job or to a manually run distributed analysis job using Ganga [14] or pAthena (the PanDA Athena client). In addition to the event references and their metadata, the extracted ROOT file also contains information about the luminosity domain and other relevant non-event metadata needed to support physics analyses such as cross section calculations.

4. Skimming Service Architecture

4.1 What is Skimming?

Skimming is the process of extracting a subset of events from the ATLAS data store into a desired format (RAW, ESD, AOD, TAG, D3PD). TAG-based skimming simply involves the use of TAG attributes when making a selection to extract and process an interesting subset of events. This can serve multiple use cases:

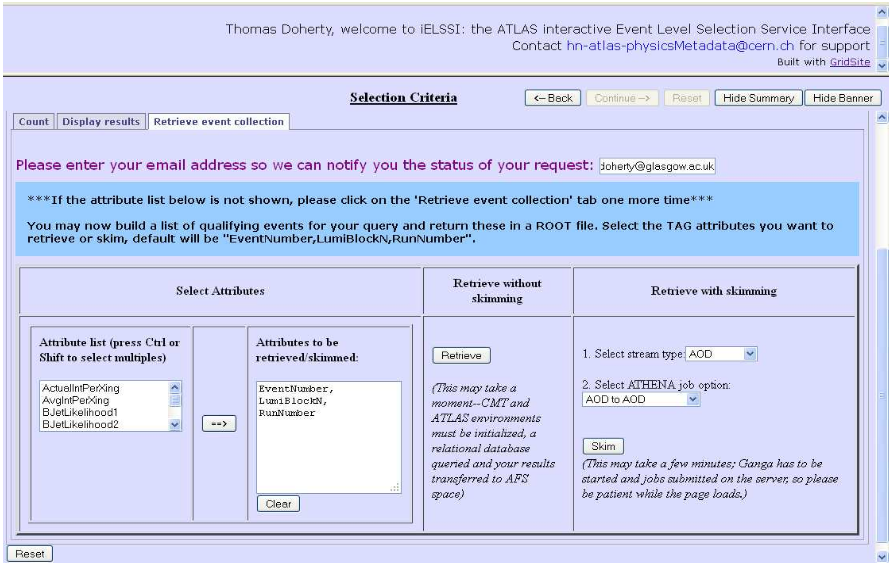
1. Monitoring.
2. Study of rare events or detector problems.
3. Packaging of data for export.
4. Creation of analysis-specific subsamples.

As an example of the possible scale involved in skimming, in a typical ATLAS Top Group physics analysis, there are approximately five hundred million events, split between the Egamma and Muons trigger streams, in the five inverse femtobarn of Data that were collected at 7 TeV centre-of-mass energy in 2011 and that pass the Good Run List requirement; of these events, approximately five thousand events remain after applying a set of 'pre-selection' cuts. By Good Run List we mean a subset of events recorded by ATLAS that are fit for analysis.

4.2 The Skimming Service Workflow

To illustrate the TAG skimming workflow we must explain how it is integrated with other tools used by ATLAS such as the DDM system, and how corresponding files of interest for each selected event can be retrieved. On the group production level a different approach is necessary and this is also explained in full.

First of all, the mechanism will be explained. As mentioned in section 3.1 a user can choose to skim the events they have just selected as illustrated in Figure 1.



Thomas Doherty, welcome to iELSSI: the ATLAS interactive Event Level Selection Service Interface
Contact hn-atlas-physicsMetadata@cern.ch for support
Built with GridSite

Selection Criteria

Please enter your email address so we can notify you the status of your request: doherty@glasgow.ac.uk

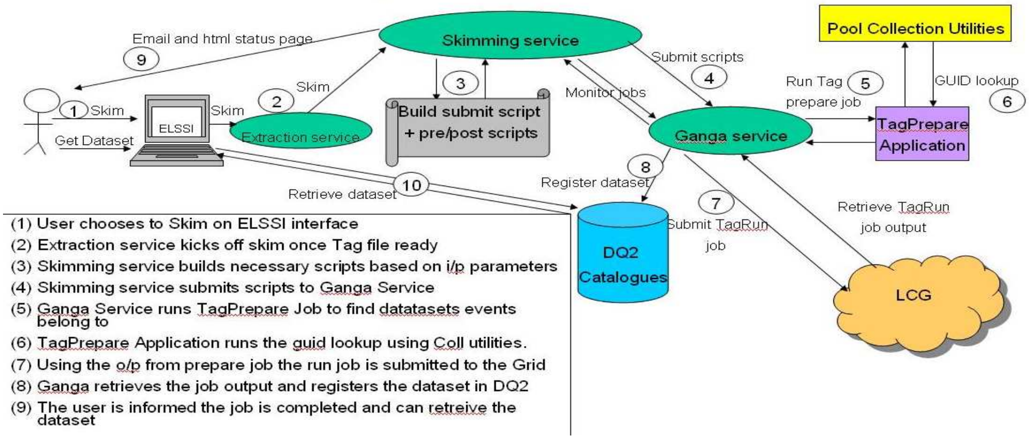
If the attribute list below is not shown, please click on the 'Retrieve event collection' tab one more time

You may now build a list of qualifying events for your query and return these in a ROOT file. Select the TAG attributes you want to retrieve or skim, default will be "EventNumber,LumiBlockN,RunNumber".

Select Attributes			Retrieve without skimming	Retrieve with skimming
Attribute list (press Ctrl or Shift to select multiples) ActualIntPerXing AvgIntPerXing BJetLikelihood1 BJetLikelihood2	==>	Attributes to be retrieved/skimmed: EventNumber, LumiBlockN, RunNumber Clear	Retrieve *(This may take a moment--CMT and ATLAS environments must be initialized, a relational database queried and your results transferred to AFS space)*	1. Select stream type: AOD 2. Select ATHENA job option: AOD to AOD Skim *(This may take a few minutes; Ganga has to be started and jobs submitted on the server, so please be patient while the page loads.)*

Figure 1: A choice to either extract (Retrieve) or Skim their selection on the iELSSI interface

A web service has also been developed for this purpose. The call between the extract Service and the Skim Service is made possible by using the underlying Athenaeum framework [15]. The skim mechanism works as in Figure 2.



(1) User chooses to Skim on ELSSI interface
(2) Extraction service kicks off skim once Tag file ready
(3) Skimming service builds necessary scripts based on i/p parameters
(4) Skimming service submits scripts to Ganga Service
(5) Ganga Service runs TagPrepare Job to find datatasets events belong to
(6) TagPrepare Application runs the guid lookup using Coll utilities.
(7) Using the o/p from prepare job the run job is submitted to the Grid
(8) Ganga retrieves the job output and registers the dataset in DQ2
(9) The user is informed the job is completed and can retreive the dataset

Figure 2 – the skim service mechanism

Recent work on the Athenaeum framework has allowed for all the input parameters needed for an extraction (such as event selection query) and skim (such as type of skim, Grid site details, splitting and output dataset name) to be configured all in one XML file. This flexible design allows for extra steps to be chained together – such as trigger decoding web service calls so that trigger selected by the user is decoded before it is added to the event selection query part of the XML configuration file. This also means that a front end client interface other than ELSSI can be introduced to allow the user to make either extract or skim calls. To this end an extract and skim command line interface has now been developed [16].

The extract service also parses the XML configuration file so that each of the relevant input parameters can be passed to the skim service call. Currently the Skimming service uses the Ganga job submission framework. Referring to Figure 2 – First a user makes the skim choice and the extract service successfully creates a TAG file. This file is used as input to the skim and extract makes a call to this service to initiate the skim. The skim service itself takes the input parameters and constructs Ganga iPython scripts based on the choices made. The ipython scripts are then made available to the Ganga service which stages two types of Job. The first job takes the extracted TAG file as input and uses the Pool Collection Utilities to perform a GUID lookup so that the GUID for each of the selected events is resolved to dataset and file names. Once the datasets and files are known then it is possible to use this information for the second job which is run on the Grid. This job finds a site that holds this data and then performs the copy necessary to make the skim for the selected events into an output dataset that can be retrieved. There are specific details for this Grid job and the mechanism involved that are discussed in section 5.

4.3 The Distributed Analysis System

As mentioned previously, PanDA makes the computing resources of the three grids available to physicists for their analysis, while hiding the complexity which is involved. Ganga, is a user interface for job definition and management on the grid, with a plugin architecture which allows it to run on various backends. There exists a PanDA backend for Ganga so that Ganga defined jobs can be submitted to the user analysis queues at sites available for PanDA. Each of the components of a job (application, backend, input, output, splitter and merger) can be implemented in various ways, as different plugins. ATLAS users, for example, can use the Athena() application, which gives access to Athena, the ATLAS analysis framework.

4.4 Data File Lookup using TAGs

The DDM system emphasises the use of datasets and does not work at the file level, the Tag Database has no knowledge of datasets and only contains references to data files using GUIDs. As was mentioned in section 2 the ATLAS POOL Collection system allows us to deal with this requirement. There is functionality available to split an extracted collection (tag file) along GUID boundaries – this allows us to then resolve to what datasets and its constituent files are needed that hold the subset of events that equate to the applied selection query cut using DQ2 functionality.

4.5 Scalability issues

Once the GUID lookup is performed the datasets and files needed for the skim are known. Using the DQ2 system it is possible to then find what suitable sites are available that holds this data. The initial design of the mechanism outlined on Figure 2 transported the TAG file with the Athena Grid job. Once scalability tests were run for this mechanism it was found that for extremely large skims of the order of millions of events – the extracted tag file was in the GB size range and this caused stability issues. The solution to this was to introduce the ‘double query’ method. This is where the extracted tag file is as before split and resolved to dataset names and files. Then rather than transporting the split extracted tag file with the Athena Grid job, once the site(s) is found that holds the data, the selection query is performed again on the equivalent TAG dataset at that site so that the skimmed files can be resolved again and copied into an output dataset. This results in a much more scalable and stable method for skimming but means that the TAG dataset must always be available at the site where its AOD, ESD or RAW equivalent is.

4.6 CVMFS

A requirement of the 'double query' mechanism for running Grid jobs as mentioned in the previous section is that a pair must exist between the dataset type being skimmed be it RAW, ESD or AOD and its TAG counterpart. If this pair does not exist at any site when a skim is being performed then the skim job will fail. In the short term the only way to rectify this is to subscribe the TAG dataset to the site to create the pair using the DATRI system. In the long term we intend to deploy all of the TAG datasets to CVMFS [17], a distributed file system introduced as a means of distributing smaller types of data to sites. This means the TAG datasets will be available to all sites that use CVMFS which is currently not all sites but the intention is that all sites will eventually use this file system.

5. Skimming use cases and workflows

As outlined in section 4 there are many different reasons for skimming within the ATLAS project. The initial primary requirement on the skimming service was to support every possible skim that is possible via the ELSSI interface. This type of skimming is suitable when a physicist is performing his/her own private analysis. It has become apparent however that physicists are normally associated with a physics group. Each physics group produce their own samples that each physicist within that group can trust and perform their analysis on. These samples are normally created using the ATLAS production system. It has also become apparent that physicists within a group are currently instructed to specify corrections to the object definitions used in the cuts they perform. These definitions are not currently stable and more than likely do not equate to the definition of the same object metadata stored in the TAG. It is assumed that these definitions will stabilise with time. Trigger definitions and therefore trigger cuts however are stable. This means that TAG skimming at a group level is currently more useful for data prep groups or physics groups that only apply trigger cuts. The SUSY physics group and Heavy Ion (HI) groups for example have provided the TAG developers use cases to support at group level.

The requirements of the ATLAS Production system (Prodsys) are completely different to those of the skimming service and private ELSSI based skimming. Prodsys does not use the tools created for analysis such as pAthena and Ganga. Prodsys does however use the basic building block of any TAG skim – the Reco transform. ATLAS Job Transforms are wrapper scripts around Athena which facilitate configuration, reproducibility and error reporting from Athena jobs. Job transforms can chain together single step transforms (e.g., RAW -> ESD) into multi-step ones, each consuming the output of the previous step in the chain. The ATLAS data products discussed in section 2 can be used as input to and be created from this transform (apart from RAW). Two further advantages of using this transform is that not only can it handle extracted TAG files as input – it can create data products such as D3PDs that are created from pre-TAG products such as AODs. Most physics groups have their group specific D3PD outputs. This in turn means that if they create this D3PD using the Reco transform it will be possible to use TAG based skimming in Prodsys to produce their group samples. It is currently necessary for Prodsys to chain the Reco transform to another transform called the stripTag transform to create a successful skim. This transform is used to strip away from the input TAG file any reference to files (AOD files for example in an AOD skim) that are not necessary for the skim and are not available at the site the skim is being performed. This transform will be deprecated once Athena performs the same functionality. For all types of output when using the Reco transform the type of output is defined by simply giving the output file name – be it an AOD, ESD or specific type of D3PD file.

Use-case specific details for each ATLAS data product will now be covered.

5.1 RAW skims

5.1.1 Private skimming

The TAG skimming service does support RAW skims. In this case the Reco transform is not used but the AtlCopyBSEvent.exe tool is. This tool performs a byte stream copy of the event. The workflow followed is as in Figure 2. The Tag Prepare step is instructed to run a GUID lookup on the RAW stream reference so that the RAW files can be resolved to datasets and files. The above tool is then used to copy each file into an output dataset. This use case was developed in liason with the Jet Trigger group and has been used for a private reprocessing campaign.

5.1.2 Prodsys group skimming

The Heavy Ion (HI) group currently use this type of skimming for their RAW reprocessing campaign. A transform has been created that wraps the AtlCopyBSEvent.exe tool so that this type of skim is possible in the prodsys environment. This TAG skimming use case is particularly useful for this group as it allows them to perform trigger cuts using TAGs without having to decompress and open each RAW file to perform a trigger cut.

5.2 AOD and ESD skims

5.2.1 Private skimming

This type of skim in the skimming service also follows the workflow outlined in figure two. The Ganga ipython submit script for the Grid job in this case can use the Reco transform to perform the skim once the output from the Tag prepare step is given. The Tag prepare step is instructed to run a GUID lookup on either an AOD or ESD stream reference depending on the type of skim. One current problem with ESD skimming is that after a small window of opportunity the ESD dataset is only available at CERN. Analysis jobs are not allowed to run at CERN sites and therefore ESD skims.

5.2.2 Prodsys group skimming

There has been no demand for these types of skimming so far at group level although work is ongoing to allow the Reco transform to create an ESD skim from RAW using a tag file as input.

5.3 D3PD skims

5.3.1 Private skimming

As with the ESD and AOD skims the skimming service can use the Reco transform to perform D3PD skims. As AODs are used as input to D3PD skims the AOD stream reference is used at the GUID lookup stage. At this point SUSY and SMWZ D3PD skims are supported in the service. Expanding on this is very simple if the D3PD skim is possible using the Reco transform.

5.3.2 Prodsys group skimming

This type of skim is possible by chaining the output of the striptag transform as input to the Reco transform. The SUSY group are currently testing and validating the use of TAG based skimming to create their D3PD samples at central production level. The cuts they are interested in using with TAGs are trigger based and therefore stable. Once this use case is validated it is hoped that it will pave the way for other groups to use the same workflow.

6. Conclusion

TAG based skimming works. The development and use of this type of skimming within ATLAS is growing in importance. The future direction of development will be on two parallel strands, based on the requirements of central group production skimming and private skimming. So far, in central production, the use of TAG based skimming has already proven to be advantageous for the SUSY and HI groups. For private skimming the ATLAS Jet trigger group have also shown that TAG based skimming can help streamline their workflow. All of these groups use the stable trigger cuts as their main selection criteria. Once definitions for physics objects stabilise, without the need for constantly changing corrections to be applied, the use of TAG based skimming as a method for cut based analysis will blossom.

References

[1] The ATLAS Collaboration 2005 ATLAS Computing, Technical Design Report

[2] Atlas Computing Group, “Computing Technical Design Report”, CERN LHCC-2005-022 ISBN 92-9083-250-9, 2005.

[3] Assamagan K, Barberis D et al. 2004 Final report of the ATLAS AOD/ESD Definition Task Force - ATLAS Notes Detectors and Experimental Techniques Software (CERN ATL-COM-SOFT-2004-008)

[4] M. Branco, D. Cameron and T. Wenaus. A Scalable Distributed Data Management System for ATLAS", J. Phys.: Conf. Ser. 119 052009, 2008.

[5] M. Lassing, et. al., “Managing ATLAS data on a petabyte-scale with DQ2”, J. Phys.: Conf. Ser. 119 062017, 2008.

[6] T. Maeno for the ATLAS collaboration, “PanDA: Distributed production and distributed analysis system for ATLAS”, J. Phys.: Conf. Ser. 119 062036, 2008.

[7] Nilsson P, et al, “Experience from a pilot based system for ATLAS” (2008) J. Phys.: Conf. Series 119 062038

[8] The ATLAS Computing Model. Technical Report CERN-LHCC-2004-037/G-085, CERN, January 2005.

[9] A Flexible, Distributed Event Level Metadata System for ATLAS D. Malon, J. Cranshaw, K. Karr, J. Hrivnac, A. Schaffer, CHEP Mumbai, India, February 2006

[10] ROOT: http://root.cern.ch.

[11] Deullmann D, The LCG POOL Project, General Overview and Project Structure Proceedings of CHEP 03, MOKT007

[12] Zhang Q, Engineering the ATLAS TAG Browser. J.Phys.: Conf. Ser. 331 042046

[13] CERN IT Service 2012 AFS - The Andrew File System (http://consult.cern.ch/service/afs/)

[14] J. Elmsheuser, et. al., “Reinforcing user data analysis with Ganga in the LHC era: Scalability, monitoring and user-support.”,J.Phys.: Conf. Ser. 331:072011, 2011.

[15] Hrivnac J, et al 2011 ATLAS Tags Web Service calls Athena via Athenaeum Framework J.Phys.: Conf. Ser. ATL-SOFT-PROC-2011-040

[16] Athenaeum Command Line Interface Implementation https://docs.google.com/drawings/d/1AJ6FQ17wRC99OyO8YCvhMawrjnRbczWI1_-lSb7K0xs/edit

[17] An alternative model to distribute VO software to WLCG sites based on CernVM-FS: a prototype at PIC Tier1 - 2011 J. Phys.: Conf. Ser. 331 062036