AGIS: Integration of new technologies used in ATLAS Distributed Computing

Alexey Anisenkov$^{1,2}$, Alessandro Di Girolamo$^3$, Maria Alandes Pradillo$^3$

$^1$ Budker Institute of Nuclear Physics, Novosibirsk, Russia
$^2$ Novosibirsk State University, Novosibirsk, Russia
$^3$ CERN, European Organization for Nuclear Research, Switzerland

E-mail: Alexey.Anisenkov@cern.ch

Abstract. The variety of the ATLAS Distributed Computing infrastructure requires a central information system to define the topology of computing resources and to store different parameters and configuration data which are needed by various ATLAS software components. The ATLAS Grid Information System (AGIS) is the system designed to integrate configuration and status information about resources, services and topology of the computing infrastructure used by ATLAS Distributed Computing applications and services. Being an intermediate middleware system between clients and external information sources (like central BDII, GOCDB, MyOSG), AGIS defines the relations between experiment specific used resources and physical distributed computing capabilities. Being in production during LHC Run1 AGIS became the central information system for Distributed Computing in ATLAS and it is continuously evolving to fulfill new user requests, enable enhanced operations and follow the extension of the ATLAS Computing model. The ATLAS Computing model and data structures used by Distributed Computing applications and services are continuously evolving and trend to fit newer requirements from ADC community. In this note, we describe the evolution and the recent developments of AGIS functionalities, related to integration of new technologies recently become widely used in ATLAS Computing, like flexible computing utilization of opportunistic Cloud and HPC resources, ObjectStore services integration for Distributed Data Management (Rucio) and ATLAS workload management (PanDA) systems, unified storage protocols declaration required for PandDA Pilot site movers and others. The improvements of information model and general updates are also shown, in particular we explain how other collaborations outside ATLAS could benefit the system as a computing resources information catalogue. AGIS is evolving towards a common information system, not coupled to a specific experiment.

1. Introduction
The ATLAS experiment [1] at the Large Hadron Collider has successfully collected billions of events during the LHC Run1 and the first 2 years of the LHC Run2 (2015 and 2016). All these data, plus tens of billions of events that have been simulated, are stored and processed over the ATLAS Distributed Computing infrastructure (ADC) [2]. ADC relies on approximately 130 computing centers, which pledge in 2016 about 140 PB of disk storage, more than 100 PB of tape storage, 150k cores (and up to
another 150k over-the-pledge or opportunistic resources), plus several other services like Squids, Frontier servers, and FTS servers [3]. The ATLAS Grid Information System (AGIS) is the framework which describes the ADC services and their topology, and exposes them in a unified way to the various ATLAS software components.

2. Main AGIS Concept
The variety of the ADC resources requires a central information system which collects, structures, and exposes the topology and many parameters that are needed to effectively make the data transfers, submit the jobs, properly configure the various services, and monitor coherently the whole ADC.

AGIS defines the topology of the ADC resources; it masks the heterogeneity of the computing infrastructures (EGI [4], OSG [5] and NorduGrid [6]) and provides, through a set of REST API, a consistent Computing Model definition for the ATLAS application services and developers. One of the key features of AGIS is that it makes clear distinction between physical computing resources provided by resource centers and the ones used by the experiment. Providing an abstraction layer from the physical resources, the system allows the experiment to define their own real organization of resources.

AGIS automatically collects and caches topology relations and static information about site specifics from various databases and external information sources like gLite BDII [7], GOCDB [8], MyOSG [9], REBUS [10]. It integrates such data with other dynamic information of site resources and services, like site and service status, resource downtimes and blacklisting objects for Distributed Data Management (DDM, Rucio) [11] or PanDA (ATLAS Production and Distributed Analysis workload management) [12] systems. Also, AGIS refreshes the information periodically and keeps it up to date and tracks possible changes.

In addition, to fit the ADC applications, additional data models and object relations were introduced. In the system, thanks to the flexible approach of resource definitions, which allows transparent declaration of virtual resources, we have been able to integrate also opportunistic Cloud resources and HPC, which have become widely used by ATLAS Computing.

One of the challenges that we tackled was indeed to enable the ATLAS Distributed Data Management and Workflow Management systems to use new type of resources like Object Stores.

3. Object Store Integration and Resource Unification
One of the recent AGIS updates includes the implementation of a new type of storage resource known as Object Storage [13], now becoming more and more used by various companies like Facebook and Amazon, and provided as default type of storage by several Cloud providers. The Object Storage is a storage implementation, where the data are managed as objects, including their metadata, as opposed to traditional file storage, where the data are managed through the file hierarchy or block storage, where the data are organized as blocks within sectors. This new Object Store architecture, which introduces new concepts like buckets, motivated us to review and deeply re-think the storage service implementation we have in AGIS.

In AGIS, for the storage service, we have one entry for each storage endpoint declared in the GOCDB or OIM. This is actually not enough to fully describe a storage, and actually it might be misleading, since different protocols used to access the same storage might have different endpoints and thus considered as completely different storages. In AGIS we have addressed this issue by declaring the Object storage as a service with attached ACL, quotas and protocols (at least one mandatory). In this way we are able to know which are the various protocols available to read specific files for given storage. Also, we have added the concept of activity and priority of protocols for that specific activity to allow the selection of the best protocols depending on the workflow, for instance for third-party copy the most effective protocols for a storage might be gridftp while for direct I/O data access performed from the Worked Nodes it could be Xrootd. For Object Store specifically we have protocols like S3 and WebDAV, the first used by the jobs downloading/uploading data from/to the OS, the second one for instance to read directly log files.
In the Figure 1 we show a simplified AGIS diagram.

The Object Store storage new definition is the first that we implemented in AGIS, while the other storages for now are still with the previous definition, which allowed the use of different protocols by ATLAS at the level of DDM Endpoint by configuring each of them for each storage: we will move to this new storage definition all the other storage in the near future.

![AGIS simplified diagram](image)

**Figure 1. AGIS simplified diagram**

### 4. Consolidation of Storage Protocols

An important step we performed in the 2016 was the consolidation of Storage Protocols between Panda, the ATLAS Workflow Management System, and Rucio, the ATLAS Distributed Data Management System. What we had before for Panda site configuration was a locally cached DB table, also known as SchedConfig, where for each Panda Resource (e.g. SiteX_SingleCore, SiteX_MultiCore, SiteX_SingleCore_HighMem) we had hardcoded the storage endpoints where to find the inputs and where to write the outputs. What we have now is actually the logical entities, and not anymore the full endpoint. In practice we have now completely coupled the DDM objects into Panda, when a job need to read from a certain DDM Endpoint it will, through the REST interface [14] AGIS provides, be able to explode the logical information into the actual full endpoint (including protocol). This heavily simplify operations in case we want to be able to e.g. change main protocols to be used, but also enable Panda to exploit in an easier way other workflows like accessing data over WAN: it is enough to know the logical DDM Endpoint from where to read the data and the job is able to find the best suitable protocols for WAN job read for that endpoint.
5. AGIS Evolution: a Generic Solution for Other Communities
AGIS is constantly evolving. Not only for ATLAS, but also for other communities. In the past year we have started a refactoring work to decouple the ATLAS experiment specific parts and the core, which is actually experiment independent. This new framework, which has been called CRIC for Computing Resources Information Catalogue, will be able to support other communities providing a shared core part plus experiment specific plugins. There is a paper dedicated to CRIC [15] in this same CHEP conference, where this framework is described more deeply.

References
[1] The ATLAS Collaboration 2008, The ATLAS Experiment at the CERN Large Hadron Collider JINST 3 S08003
[2] Simone Campana et al 2015 ATLAS Distributed Computing in LHC Run2 J. Phys.: Conf. Ser. 664 032004
[3] S Jézéquel and G Stewart 2012 ATLAS Distributed Computing Operations: Experience and improvements after 2 full years of data-taking J. Phys.: Conf. Ser. 396 032058
[4] European Grid Infrastructure (EGI), http://www.egi.eu/
[5] Open Science Grid (OSG) Web Page, http://www.opensciencegrid.org/
[6] NorduGrid project, http://www.nordugrid.org/
[7] Field L and Schulz M W 2004 Grid deployment experiences: the path to a production quality LDAP based grid information system. Proceedings of the International Conference on Computing in High Energy and Nuclear Physics (CHEP 2004)
[8] Gilles Mathieu et al 2010 GOCDB, a topology repository for a worldwide grid infrastructure J. Phys.: Conf. Ser. 219 062021
[9] Open Science Grid Information service (MyOSG), http://myosg.grid.iu.edu/about
[10] WLCG REsource, Balance, USage (REBUS), http://wlcg-rebus.cern.ch
[11] C. Serfon et al 2016 Rucio, the next-generation Data Management system in ATLAS Nuclear and Particle Physics Proceedings 273-275 969
[12] K De et al 2015 The future of PanDA in ATLAS distributed computing J. Phys.: Conf. Ser. 664 062035
[13] Object Storage architecture, https://en.wikipedia.org/wiki/Object_storage
[14] REST interfaces, http://en.wikipedia.org/wiki/Representational_state_transfer
[15] A. Anisenkov et al (2016) Consolidating WLCG topology and configuration in the Computing Resource Information Catalogue. Proceedings of the CHEP 2016 conference J. Phys.: Conf. Ser.