ATLAS Grid Information System

Alexey Anisenkov¹, Alexei Klimentov², Roman Kuskov¹, Torre Wenaus²
¹ Budker Institute of Nuclear Physics, Novosibirsk, Russia
² Brookhaven National Laboratory, Upton, NY 11973, USA
Email: Alexey.Anisenkov@cern.ch

Abstract. ATLAS is a particle physics experiment at the Large Hadron Collider at CERN. The detector produces huge volumes of data, of the order of petabytes per year. To satisfy ATLAS requirements for petabyte scale production and distributed analysis processing, a grid technology is used to distribute, store and analyse these immense amounts of data. In this paper we present ATLAS Grid Information System (AGIS) designed to integrate configuration and status information about resources, services and topology of whole ATLAS Grid needed by ATLAS Distributed Computing applications.

1. Introduction
The ATLAS [1] experiment produces millions of gigabytes of data annually through simulation production and tens of petabytes of data per year from the detector itself. This data is distributed globally according to the ATLAS computing model [2]. The ATLAS computing model is based on a worldwide computing grid infrastructure that uses a set of hierarchical tiers. After a first pass reconstruction at CERN, raw data from the detector and reconstructed data products are replicated to 10 Tier-1 centres according to the datatype and the amount of data the centre has agreed to store. In addition to the data from the detector, simulated data is produced on Grid resources worldwide, and this data must also be replicated. Tier-1 and associated tiers are grouped into cloud. The ATLAS grid is composed by three independent sub-grids:
• Enabling Grid for E-sciencE (EGEE) [3]
• Nordic Data Grid Facility (NDGF) [4]
• Open Science Grid (OSG) [5]
These three sub-grids form the Worldwide LHC Computing Grid infrastructure (WLCG). Figure 1 shows a schematic view of ATLAS Grid topology.

This immense volume of data must be handled by the Computing with integrated Information systems. ATLAS Grid Information System represents a general-purpose storage of different parameters and configuration data needed by ATLAS Distributed Computing (ADC) [6,7] applications.

2. AGIS Requirements and Motivation
ADC applications require a lot of information about site resources, topology, and services originally stored in different sources. Sometimes configuration parameters and related static data hardcoded in application programs or spread out various configuration files. Each application in this way need to
implement communication logic to retrieve data from external sources that produces a lot of code duplications. To satisfy ADC requirements Central Information System should be designed and implemented to have a coherent approach to store and publish data. Such central place should be a database based system with good scalability and convenient data access interfaces allowed operations with data in terms of abstractions.

Such storage system helps to solve different information duplication issues and inconsistencies between data currently stored in many external databases, configuration files and even hard-coded in applications. It also helps to solve synchronization issues, to reduce code duplication and simplify application code logic. By introducing own caching mechanism AGIS helps to improve performance in data management.

Also, using AGIS as one data repository will improve system maintainability and allow adding new classes of information.

There are many different parameters that should be managed in that way. It is, for example, Sites availability information, general ATLAS topology, Sites downtime information, resources description, data access permissions and so forth. AGIS is designed to integrate configuration and status information about resources, services and topology of the whole ATLAS grid from various independent sources such as EGEE BDII, Panda site information DB [8], GOCDB, ATLAS DDM [9] configuration files (Tiers of ATLAS) and others.

3. AGIS Architecture Overview
AGIS implementation is based on ORACLE database backend. AGIS uses Django [10] as a high-level web application framework written in Python. To simplify data access Django built object-relational mapping technique is used. It allows manipulating of content data in terms of data models and actually uses relational database to store objects.

AGIS integrates middleware information sources by caching their information, removing the source as a direct dependency for the client, but without duplicating the source information system. To keep AGIS database up to date various content providers, also so-called as collectors, periodically check external sources and do content synchronization. In additional to data collected from external sources AGIS also stores own data managed within the database.

Access to all stored data exposed via user-friendly WEB interface and also via API and CLI tools provided by AGIS. AGIS server allows clients to get data in XML format, for instance all ATLAS topology can be obtained in such view.

The system architecture should be extendable as possible and allow simple adding of new forms of data. Figure 2 shows schematic view of AGIS architecture.

4. Theory and practice
In practice, it is not really trivial to change anything in already working system. If some component changes, all applications using it must be checked for consistency and fixed if need to correctly process the change.
One more fact is that it is really hard to integrate new Information System into working environment without enforcing significant changes to existing applications. When such applications were developed, there were no common high level design principles, no common software projects, so many developers created their own configurations specific for their project. For now, such applications have to migrate to another backend, and it brings headaches to everyone.

So, the first attempt in AGIS implementing was to mirror present structures of data, in order to make simple migration of existing ADC applications to AGIS backend. After this, all modification within AGIS could be made more transparently for external applications.

The development started from definition of all entities and relations between them. The first problem that was met is different meanings of Site concept by different external data storages and applications. Also there were no any agreements or conventions in sites naming rules. AGIS has adopted wide ADC definition and built hierarchical topology: Site is a part of a Regional Center which is a collection of Sites by some sign; the Regional Centers form ATLAS Cloud; the Site has different Processing Resources, Storage Elements, Computing Elements, production and analysis queues; the Storage Elements consist of many DDM endpoints.

The Figure.3 shows simplified diagram of AGIS entities implemented.

5. Stored Information
There are two types of information stored in AGIS:
- data retrieved from external sources (TiersOfATLAS, GOCDB, BDII, myOSG, PanDa databases, TaskRequest database, SAM, etc)
- data managed within AGIS (eg. site blacklisting)

In current implementation, AGIS stores inside all information concerning ATLAS topology: clouds, regional centers, sites specifics, such as geography, time zone, geo coordinates, etc. It also stores resources and services information – LFC, FTS, CE, SRM, Squid, Frontier services and others.

Incomplete list of Information stored in AGIS:
- ATLAS topology (clouds, centres, sites and sites specifics)
- Site’s Resources and Services information, status and its description
- Site information and configuration
- Data Replication sharing and pairing policy
- List of Activities and its properties (Functional Tests, re(Processing), Data distribution, etc.)
- Global Configuration Parameters needed by ADC applications
- User related information (privileges, roles, account info) for ADC applications (AGIS, Panda, DaTRI, etc)
- Downtime information about ATLAS sites, services
- Site Blacklisting data

![Simplified AGIS class diagram](image)

**Figure 3.** Simplified AGIS class diagram

Downtime information about ATLAS sites and services automatically publishes into AGIS Downtime Calendar. Downtime Calendar represents information about ATLAS sites downtimes from all three sub-grids (EGEE, NDGF and OSG).

Currently implemented data model allowed to separate data representation and data collecting into different modules. Google Calendar chosen as one of the best available solutions for data representation. An example snapshot of AGIS Downtime Calendar view is shown in Figure 4.

AGIS service periodically collects downtimes from external sources (OIM, GOCDB), updates database content and synchronizes changes on Google Calendar. ADC data subscriptions applications used in production asks AGIS through API to retrieve site availability status, to check if requested site in downtime and need to be excluded from Data Replication activity.

6. Project status and plans
For the present time AGIS project is in stage of extending its functionality and improving reliability. Sites Downtime calendar is fully implemented and being continuously improved with the help of users feedback.

Information stored in TiersOfATLAS file completely moved to AGIS database. There are also some improvements in ATLAS topology structure implemented. API and CLI interfaces allow access
to all information AGIS stored for external applications. There is also AGIS.info API interface implemented which is a replacement for DQ2.info interface to TiersOfATLAS contents. This interface should help in migration of old applications to new AGIS backend.

WEB client interface is under development now. First-shot implementation is ready and deployed on AGIS server, however in should be extended to all entities AGIS stored inside.

Current project work focuses on extending data structures stored in AGIS. A new concept called RegionalCenter is being imported as a replacement of Tier and Federation entities. It will allow more clear way to represent topology information and also will fix some inconsistencies presenting now.

New external data storages will be supported in the near future. So, more ADC applications could migrate to AGIS from their local storages. The most important services to support in the nearest future are Frontier and Squid ones.

![AGIS Downtime Calendar View](image)

**Figure 4. AGIS Downtime calendar view.**

### 7. Conclusion

AGIS is developed to provide a single portal and information cache for application developers and interactive users. The interactions with various information providers are hidden. The synchronization of AGIS content with the external sources is performed by agents which periodically communicate with sources via standard interface and update database content. Implemented python API helps end user to obtain data from AGIS database.

AGIS is currently in production for the ATLAS experiment. Released AGIS API used by ADC applications to retrieve data.

AGIS has already proved to be useful for ATLAS by replacing many configuration files and downtime information sources. It’s used by ADC monitoring and dataset subscriptions applications. The AGIS Downtime Calendar, represented ATLAS sites being in downtime, helps and is widely used by ADC shifters and experts during data distribution.
References

[1] The ATLAS experiment, http://atlas.ch.
[2] The ATLAS Computing Model, Adams, D; Barberis, D; Bee, C P; Hawking, R; Jarp, S; Jones, R; Malon, D; Poggio, L; Poulard, G; Quarrie, D et al. - ATLAS Collaboration, ATL-SOFT-2004-007; ATL-COM-SOFT-2004-009; CERN-ATL-COM-SOFT-2004-009; CERN-LHCC-2004-037-G-085. Geneva: CERN, 2005 - 37
[3] Enabling Grid for E-sciencE (EGEE), http://www.cern.ch/egee
[4] Open Science Grid (OSG) Web Page, http://www.opensciencegrid.org/
[5] Nordic Data Grid Facility (NDGF) Web Page, http://www.ndgf.org/
[6] ATLAS Distributed Computing, web page, https://twiki.cern.ch/twiki/bin/view/Atlas/AtlasDistributedComputing
[7] A. Anisyonkov, A. Klimentov, D. Krivashin, M. Titov. ATLAS Distributed Computing, XXII International Symposium on Nuclear Electronics & Computing (NEC 2009) (Institute for Nuclear Research and Nuclear Energy, Varna, Bulgaria, 2009), pp. 4. September 7-14, 2009; Proceedings of the Symposium; E10,11-2010-22, Dubna, 2010 - p. 45-49
[8] PanDa project, web page: https://twiki.cern.ch/twiki/bin/view/Atlas/Panda
[9] Klimentov A., ATLAS Distributed Data Management Operations. Experience and Projection, Journal of Physics: Conference Series
[10] Django project web site, http://www.djangoproject.com