COMPUTING RESOURCE INFORMATION CATALOG: THE ATLAS GRID INFORMATION SYSTEM EVOLUTION FOR OTHER COMMUNITIES

A.V. Anisenkov[1,2]
on behalf of the ATLAS Collaboration and CRIC team

[1] *Budker Institute of Nuclear Physics, 11, ak. Lavrentieva prospect, Novosibirsk, 630090, Russia*

[2] *Novosibirsk State University, 2, Pirogova street, Novosibirsk, 630090, Russia*

E-mail: Alexey.Anisenkov@cern.ch

The Worldwide LHC Computing Grid infrastructure links about 200 participating computing centers affiliated with several partner projects. It is built by integrating heterogeneous computer and storage resources in diverse data centers all over the world and provides CPU and storage capacity to the LHC experiments to perform data processing and physics analysis at petabytes scale data operations. In order to be used by the experiments, these distributed resources should be well described, which implies easy service discovery and detailed description of service configuration. This contribution describes the evolution of the ATLAS Grid Information system (AGIS), a central information middleware system built for ATLAS Experiment, into a common Computing Resource Information Catalog (CRIC), the framework, which is aimed to describe the topology of the LHC Experiments computing models, providing unified description of resources and services used by experiments applications. The main goal of the information system is to provide consistent Distributed Computing model definition, collect and aggregate data coming from various data providers and generic information sources (like GOCDB, OIM, BDII), store and expose different parameters and configuration data which are needed by Distributed computing software components.

Keywords: information system, grid, distributed computing, grid computing facilities, grid middleware

1. Introduction

One important technical challenge in functioning of a modern experiment in high energy physics is the integration of Information and Computing resources into the single system to effectively store and analyze experimental data. For instance, the ATLAS experiment [1] at the Large Hadron Collider at CERN collects billions of events each data-taking year and produces an even larger amount of events, tens of petabytes data, through the simulation production according to several physics and detector models. All these data are stored and processed over the ATLAS Distributed Computing infrastructure (ADC) [2], heterogeneous distributed environment, including the Worldwide LHC Computing Grid (WLCG) [3], opportunistic Cloud and Supercomputers centers.

The variety of involved distributed computing infrastructures used by the Experiments at LHC requires a central information system to define the topology of computing resources and store different configuration data which are needed by many software components, experiment oriented middleware services and applications. The ATLAS Grid Information System (AGIS) [4] is such middleware framework designed for ATLAS experiment in order to mask the heterogeneity of computing environment and provide unified way to operate and configure ADC applications. AGIS collects, structures, and exposes the topology and many parameters that are needed to effectively make the data transfers, submit the jobs, properly configure high-level services, and monitor coherently the whole ADC.

Being in production during LHC Run-1 and in current Run-2 period as a central information grid middleware system for Distributed Computing in ATLAS, and providing agile, robust and flexible service, AGIS is continuously evolving to fulfill new user requests, enable enhanced operations and follow the extension of the ATLAS Computing model. AGIS is evolving not only within ATLAS, but also extends applications for other communities. The work started last year to upgrade AGIS and decouple the system into the ATLAS experiment specific part and the core, which is actually experiment independent. This new framework, which has been called CRIC for Computing Resources Information Catalogue, is focused to support other communities, in addition to ATLAS, the experiments at LHC, by providing a shared core part plus separate implementation of experiment specific plugins.

2. The Computing Resource Information Catalog

The Computing Resource Information Catalog will describe WLCG topology and will be the entry point to consume information about WLCG resources. Based on refactored architecture of AGIS system, it's comprised of a core module that it is integrated with experiment specific modules or plugins, as described further in this paper. The core module implements the description of computing model for physical resources provided by distributed computing environment, while experiment oriented plugins are focused to extend the model and structure the resources how they are used by given experiment. The core part consumes information from different information sources and it's flexible enough to add or remove information providers and even allow sites to enter directly information about their resources. Such flexibility allows to extend Experiment Computing model in a unified way and complement traditional grid resources provided by WLCG computing with other opportunistic resources like commercial Cloud platforms, HPC or supercomputer centers. Moreover, opportunistic sites do not need to be part of GOCDB [5] or OIM [6] anymore, nor run a BDII service to be able to be described in CRIC. This offers a major advantage to small sites who don't have enough effort to run extra information services. The Figure 1 illustrates relations between CRIC components and involved information sources.

For the moment, experiments like ATLAS and CMS are planning to be fully integrated with CRIC and to provide via CRIC interfaces all required configuration for their high-level computing frameworks like data management and workload management systems. Alice and LHCb for the time being are not interested in running dedicated experiment CRIC plugins but they are supposed to use lightweight CRIC instance as a single entry point and information provider for WLCG topology. Lightweight experiment CRIC instances for ALICE and LHCb will define set of sites and services which are used by these experiments. This is basic functionality which is also required for WLCG testing, monitoring and accounting systems. ALICE and LHCb lightweight CRICs will retrieve required information from the existing experiment topology systems like GOCDB, OIM, REBUS, and BDII.

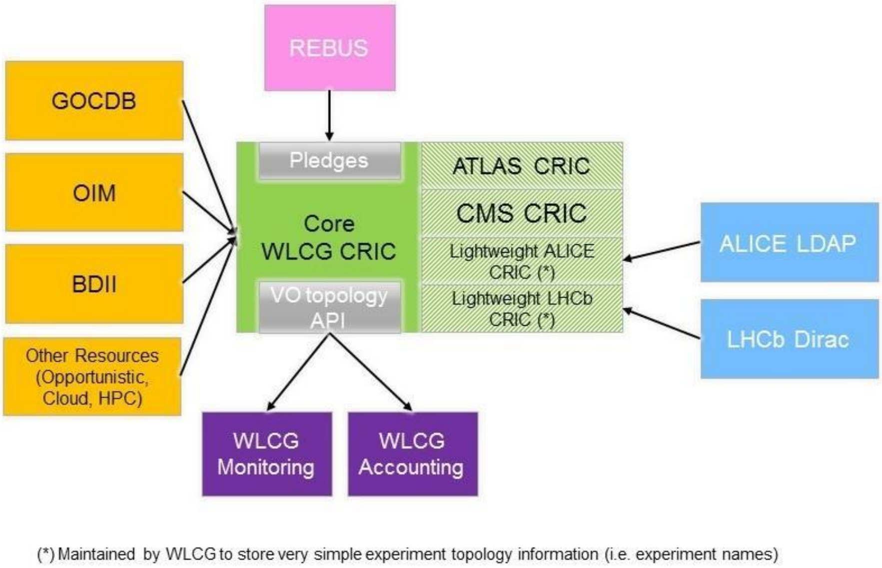

Figure 1. CRIC components and information sources

The COMPASS Experiment at CERN SPS is also evaluating the CRIC system as information middleware service. COMPASS distributed computing environment is similar to one used by ATLAS Computing – they rely on similar computing models and use same high-level frameworks for distributed data management and workload management systems. It makes possible to partial share implementations of CRIC modules for ATLAS and COMPASS plugins. Currently ongoing implementation of COMPASS CRIC plugins will help with further migration of AGIS to ATLAS CRIC.

3. CRIC Architecture

The system implements a client-server computing technology by providing a high-level user Web applications (WebUI) and REST style programming interfaces (API) to access, modify and explore data stored in database backend. In other word, CRIC is a database oriented system, which currently uses the Django [7], as a high-level web application framework written in Python. Thanks to the object relation mapping (ORM) technique which is built into the Django, CRIC does not depend on specific implementation of database backend. Access to the content of the database is applied in terms of high level models, thus avoiding any direct dependence on the relational database system used. This is important functional feature for CRIC, since in general Experiment CRICs require to support various database management system, for example ATLAS actively uses Oracle RBMS, while CMS and COMPASS prefer MySQL servers.

One of the key features of CRIC is that it makes clear distinction between physical computing resources *provided by* resource centers and the ones *used by* the experiment. By providing an abstraction layer from the physical resources, the system allows the experiment to define their own (real) organization of resources and complement the information schema with experiment specific structures. To automatize operations and validate information spread over different external sources available for Experiments, CRIC plays essential aggregator role by automatically collecting, caching and correcting data, for example the topology relations and static information about site specifics from various databases and external information sources (gLite BDII, GOCDB, MyOSG and REBUS). It integrates such data with other dynamic information of site resources and services, like site and service status, resource downtimes and blacklisting objects. As an example from ATLAS workflow, AGIS allows to automatically exclude from the production the resources being temporary blacklisted or in scheduled downtime for Distributed Data Management (Rucio) [8] and ATLAS Production and Distributed Analysis workload management (PanDA) [9] systems.

The concept of *provided by* and *used by* resources is implemented in decoupling the system into the shared core and experiment-related services. The purpose of the core is to describe physical resources hosted by the distributed computing sites, which are part of the WLCG infrastructure. Experiment CRIC encapsulates specific configuration for the resources used by the experiments for data storage, data distribution and data processing. It contains all necessary information for organization of the data management and workflow management activities and models the experiment specific concepts. Therefore, experiment CRIC serves the experiment data management and workload management systems as well as various operational tools, monitoring and accounting systems. It plays a key role in the information flow of the experiment offline computing.

Objects described in the experiment CRIC reference objects defining physical services contained in the core CRIC. Both (core and experiment) parts share common implementation framework. However, experiment CRICs describe concepts which are not necessary the same for various experiments and therefore, experiment CRICs represent experiment-specific plugins.

3. Current status

The first stage of CRIC refactoring code has been successfully applied. As the result we have integrated the Bootstrap [10] toolkit as Web2.0 frontend framework into the WebUI and deployed a WebUI prototype instance for the core CRIC service. Most of functionality related to REST API have been refactored and migrated from AGIS into CRIC core service already. There are also active ongoing developments of the implementation for CRIC CMS and CRIC COMPASS plugins.

Once several collaborations are considered to be taken on board, CRIC requires support of enhanced authorization and authentication methods to have customized and fine-grained access control management. One of the current developments, currently going into the final production is related to the implementation of authorization modules. CRIC interfaces provide several authentication methods which can be used by end-clients depending of their needs: starting from local password- or SSL certificate-based authentication and ending with unified single-sign-on authentication. Several types of permissions allow Experiments to effective implement own access policies. CRIC support 3 types of permissions:

- a global permission, which helps to restrict user actions, for example to allow clients to modify only part of information in WebUI forms;
- an instance specific permission, which should consider restrictions applied to particular object (for example to allow the modification of given Site);
- and finally, object based or model permission, that affects particular type of objects (for example to allow modification of all Sites).

The CRIC development team works in close collaboration with the CMS experts in order to understand the CMS data structures and properly implement CMS Computing model. For today, into the prototype instance of CRIC CMS has been already included various CMS specific collectors (fetching data from SiteDB, GlideIn configuration files, XML factory settings), the implementation of base objects for CMS Information model (like CMSSite, ComputeUnit, ComputeResources, GlideInEntry, etc). CMS CRIC part is being integrated into CRIC instance (getting from the box all core functionality). The first production prototype is in the process to be released soon.

Next step will be to work on an evolution of the storage description: the present CRIC system will benefit on the storage description which is already in production for the AGIS framework, used by ATLAS. But we have noticed that with the evolution of storage technologies, like the usage of Object Storages, the current implementation is stretched to its limits: today each different endpoint (HTTP, SRM, GsiFTP, Xroot, etc) is a different service, and the fact that in reality they are often just various protocol of the same storage is accomplished through manual initial configuration. We have already proposed, in agreement with WLCG, a possible storage description which will describe fully the storages and will allow the experiments to exploit the various interfaces to storages in an easy and flexible way.

4. Conclusion

The Computing Resources Information Catalog is the evolution of the ATLAS Grid Information System: it benefits of the many years experience gained with the ATLAS experiments, and it is designed in such a way to offer the possibility to be used by many collaboration, which could decide the level of involvement, in terms of deepness of resource description, that they prefer. One of the CRIC key features is the clear distinction of the description of the physical resources, resources "provided by", which are fetched from various sources and stored in the CRIC Core, and the logical resources, "used by" the computing framework such as the data management and the workflow management of the collaborations, and these logical resources are the CRIC Experiment specific plugin. This architecture allows the collaborations to fully describe their computing model, and to use CRIC as central entry point for topology description and configuration of their frameworks.

The experience acquired with having AGIS in production for ATLAS in the past years make us confident that CRIC will be reliable and flexible, and it might be very useful on the long term for many experiments.

The sociological challenge of integrating CRIC in running experiments like the WLCG ones is big, we hope that the flexibility, the features, the reliability of CRIC will such that other experiments could just at least try it out. The fact that also other non-WLCG experiments like COMPASS are also evaluating CRIC give us cause for good hope.

References

[1] *Aad G. et al.* The ATLAS Collaboration, "The ATLAS Experiment at the CERN Large Hadron Collider // Journal of Instrumentation. 2008. Vol. 3. S08003.

[2] *S. Campana et al.* 2015 ATLAS Distributed Computing in LHC Run2 J. Phys.: Conf.Ser. 664 032004.

[3] The Worldwide LHC Computing Grid (WLCG), http://wlcg.web.cern.ch/ (accessed 01.11.2017)

[4] *A. Anisenkov et al.* 2017 AGIS: Integration of new technologies used in ATLAS Distributed Computing J. Phys.: Conf.Ser. 898 092023

[5] Grid Configuration Database (GOCDB), https://wiki.egi.eu/wiki/GOCDB (accessed 01.11.2017)

[6] OSG Information Management System (OIM), Open Science Grid Information service (MyOSG), http://myosg.grid.iu.edu/about (accessed 01.11.2017)

[7] Django project, http://www.djangoproject.com (accessed 01.11.2017)

[8] *C. Serfon et al.* 2016 Rucio, the next-generation Data Management system in ATLAS Nuclear and Particle Physics Proceedings 273-275 969

[9] *Maeno T. on behalf of the ATLAS Collab*. Evolution of the ATLAS PanDA Workload Management System for Exascale Computational Science // J. Phys. Conf. Ser. 2014. V. 513. P. 032062; doi:10.1088/1742-6596/513/3/03206

[10] Bootstrap toolkit, http://getbootstrap.com/ (accessed 01.11.2017)