# Accessing distributed computing resources by scientific communities using DIRAC services

V.Korenkov I.Pelevanyuk P.Zrelov
Joint Institute for Nuclear Research, Dubna
Plekhanov Russian Economics University, Moscow
korenkov@jinr.ru pelevanyuk@jinr.ru zrelov@jinr.ru

A.Tsaregorodtsev
CPPM, Aix Marseille University, CNRS/IN2P3, Marseille
Plekhanov Russian Economics University, Moscow
atsareg@in2p3.fr

## Abstract

Scientific data intensive applications requiring simultaneous use of large amounts of computing resources are becoming quite common. This domain was pioneered by High Energy Physics (HEP) experiments at the LHC collider at CERN. However, researchers in other branches of science start to have similar requirements. The experience and software tools accumulated in the HEP experiments can be very valuable for these scientific communities. One of the software toolkits developed for building distributed computing systems is the DIRAC interware. It allows seamless integration into a single coherent system of computing and storage resources based on different technologies. This product was very successful to solve problems of large HEP experiments and was reworked in order to offer a general-purpose solution suitable for other scientific domains. Services based on the DIRAC interware are now proposed to users of several distributed computing infrastructures on the national and European levels. This significantly lowers the threshold to start working with large scale distributed computing systems for the new researchers.

## 1 Introduction

Large High Energy Physics experiments, especially those running at the LHC collider at CERN, have pioneered the era of very data intensive applications. The aggregated data volume of these experiments exceeds by today 100 PetaBytes, which includes both data acquired from the experimental setup as well as results of the detailed modeling of the detectors. Production and processing of these data required creation of a special distributed computing infrastructure - Worldwide LHC Computing Grid (WLCG). This is the first example of a large-scale grid system successfully used for a large scientific community. It includes more than 150 sites from more than 40 countries around the world. The sites altogether are providing unprecedented computing power and storage volumes. WLCG played a very important role in the success of the LHC experiments that achieved many spectacular scientific results like discovery of the Higgs boson, discovery of the pentaquark particle states, discovery of rare decays of B-mesons, and many others.

In order to create and operate the WLCG infrastructure, special software, so-called middleware, was developed to give uniform access to various sites providing computational and storage resources for the LHC experiments. Multiple services were deployed at the sites and centrally at CERN to ensure coherent work of the infrastructure, with comprehensive monitoring and accounting tools. All the communications between various services and clients are following strict security rules; users are grouped into virtual organizations with clear access rights to different services and with clear policies of usage of the common resources.

On top of the standard middleware that allowed building the common WLCG infrastructure, each LHC experiment, ATLAS, CMS, ALICE and LHCb, developed there own systems in order to manage their workflows and data and cover use cases not addressed by the middleware. Those systems have many similar solutions and design choices but are all developed independently, in different development environments and have different software architectures. This software is used to cope with large numbers of computational tasks and with large number of distributed file replicas by automation of recurrent tasks, automated data validation and recovery procedures. With time, the LHC experiments gained access also to other computing resources than WLCG. An important functionality provided by the experiments software layer is access to heterogeneous computing and storage resources provided by other grid systems, cloud systems and standalone large computing centers, which are not incorporated in any distributed computing network. Therefore, this kind of software is often called interware as it interconnects users and various computing resources

**Proceedings of the XVIII International Conference «Data Analytics and Management in Data Intensive Domains» (DAMDID/RCDL'2016), Ershovo, Russia, October 11 - 14, 2016**

and allows for interoperability of otherwise heterogeneous computing clusters.

Nowadays, other scientific domains are quickly developing data intensive applications requiring enormous computing power. The experience and software tools accumulated by the LHC experiments can be very useful for these communities and can save a lot of time and effort. One of the experiment interware systems, the DIRAC project of the LHCb experiment, was reorganized to provide a general-purpose toolkit to build distributed computing systems for scientific applications with high data requirements [1]. All the experiment specific parts were separated into a number of extensions, while the core software libraries are providing components for the most common tasks: intensive workload and data management using distributed heterogeneous computing and storage resources. This allowed offering the DIRAC software to other user communities and now it is used in multiple large experiments in high energy physics, astrophysics and other domains. However, for relatively small user groups with little expertise in distributed computing, running dedicated DIRAC services is a very difficult task. Therefore, several computing infrastructure projects are offering DIRAC services as part of their services portfolio. In particular, these services are provided by the European Grid Infrastructure (EGI) project. This allowed many relatively small user communities to have an easy access to a vast amount of resources, which they would never have otherwise.

Similar systems originating from other LHC experiments, like BigPanDa [14] or AliEn [15] were also offered to use by other scientific collaborations. However, their usage is more limited than the one of DIRAC. BigPanDa is providing mostly the workload management functionality for the users and is not supporting data management operations, whereas DIRAC is a complete solution for both types of tasks. AliEn provides support for both data and workload management. However, it is difficult to extend for specific workflows of other communities. The DIRAC architecture and development framework is conceived to have excellent potential for extension of its functionality. Therefore, completeness of its base functions together with modular extendable architecture makes DIRAC a unique all-in-one solution suitable for many scientific applications.

In this paper, we review the DIRAC Project giving details about its general architectures as well as about workload and data management capabilities in Section 2. Examples of the system usage are described in Section 3 followed by Conclusions.

## 2 DIRAC Overview

DIRAC Project provides all the necessary components to create and maintain distributed computing systems. It forms a layer on top of third party computing infrastructures, which isolates users from the direct access to the computing resources and provides them with an abstract interface hiding the complexity of dealing with multiple heterogeneous services. This pattern is applied to both computing and storage resources. In both cases, abstract interfaces are defined and implementations for all the common computing service and storage technologies are provided. Therefore, users see only logical computing and storage elements, which simplifies dramatically their usage. In this section, we will describe in more details the DIRAC systems for workload and data management.

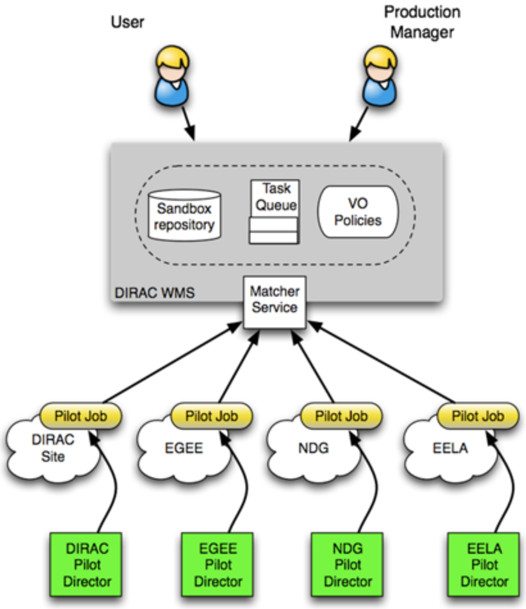

**Figure 1** WMS with pilot jobs

### 2.1 Workload Management

The DIRAC Workload Management System is based on the concept of pilot jobs [2]. In this scheduling architecture (Figure 1), the user tasks are submitted to the central Task Queue service. At the same time, the so-called pilot jobs are submitted to the computing resources by specialized components called Directors. Directors use the job scheduling mechanism suitable for their respective computing infrastructure: grid resource brokers or computing elements, batch system schedulers, cloud managers, etc. The pilot jobs start execution on the worker nodes, check the execution environment, collect the worker node characteristics and present them to the Matcher service. The Matcher service chooses the most appropriate user job waiting in the Task Queue and hands it over to the pilot for execution. Once the user task is executed and its outputs are delivered to the DIRAC central services, the pilot job can take another user task if the remaining time of the worker node reservation is sufficient.

There are many advantages of the pilot job concept. The pilots are not only increasing the visible efficiency of the user jobs but also help managing heterogeneous computing resources presenting them to the central services in a uniform coherent way. Large user communities can benefit also from the ability of applying the community policies that are not easy, if at all possible, with the standard grid middleware. Furthermore executing several user tasks in the same pilot largely reduces the stress on the batch systems no matter if they are accessed directly or via grid mechanisms, especially if users subdivide their payload in many short tasks trying to reduce the response time.

The pilot job based scheduling system allows easy aggregation of computing resources of different technologies. Currently the following resources are available for DIRAC users:

- Computing grid infrastructures based on the gLite/EMI grid middleware. The submission is possible both through the gLite Workload Management System and directly to the computing element services exposing the CREAM interface. WLCG and EGI grids are examples of such grid infrastructures.
- Open Science Grid (OSG) infrastructure based on the VDT (Virtual Data Toolkit) suite of middleware [3].
- Grids based on the ARC middleware, which was developed in the framework of the Nordugrid project [4].
- Standalone computing clusters with common batch system schedulers, for example, PBS/Torque, Grid Engine, Condor, SLURM, OAR, and others. Those clusters can be accessed by configuring an SSH tunnel to be used by DIRAC directors to submit pilot jobs to the local batch systems. No specific services are needed on such sites to include them into a distributed computing infrastructure.
- Sites providing resources via most widely used cloud managers, for example OpenStack, OpenNebula, Amazon and others. Both commercial and public clouds can be accessed through DIRAC.
- Volunteer resources provided with the help of BOINC software. There are several realizations of access to this kind of resources all based on the same pilot job framework.

As it was explained above, a new kind of computing resources can be integrated into the DIRAC Workload Management System by providing a corresponding Director using an appropriate job submission protocol. This is the plugin mechanism that enables easily new computing facilities as needed by the DIRAC users.

### 2.2 Data Management

The DIRAC Data Management System (DMS) is based on similar design principles as the WMS [5]. An abstract interface is defined to describe access to a storage system with multiple implementations for various storage access protocols. Similarly, there is a concept of a FileCatalog service, which provides information about the physical locations of file copies. As for storage services there are several implementations for different catalog service technologies all following the same abstract interface.

A particular storage system can be accessible via different interfaces with different access protocols. But for the users it stays logically a single service providing access to the same physical storage space. To simplify access to this kind of services, DIRAC aggregates plugins for different access protocols according to the storage service configuration description. When accessing the service, the most appropriate plugin is chosen automatically according to the client environment, security requirements, closeness to the service, etc. As a result, users are only seeing logical entities without the need to know the exact type and technology of the external services.

DIRAC provides plug-ins for a number of storage access protocols most commonly used in the distributed storage services:

- SRM, XRootd, RFIO, etc;
- gfal2 library based access protocols ( DCAP, HTTP-based protocols, S3, WebDAV, etc ) [6].

New plug-ins can be easily added to accommodate new storage technologies as needed by user communities.

In addition DIRAC provides its own implementation of a Storage Element service and the corresponding plug-in using the custom DIPS protocol. This is the protocol used to exchange data between the DIRAC components. The DIRAC StorageElement service allows exposing data stored on file servers with POSIX compliant file systems. This service helps to quickly incorporate data accumulated by scientific communities in any *ad hoc* way into any distributed system under the DIRAC interware control.

Similarly to Storage Elements, DIRAC provides access to file catalogs via client plug-ins. The only plug-in to an external catalog service is for the LCG File Catalog (LFC), which used to be a *de facto* standard catalog in the WLCG and other grid infrastructures. Other available catalog plug-ins are used to access the DIRAC File Catalog (DFC) service and other services that are written within the DIRAC framework and implement the same abstract File Catalog interface [7]. These plug-ins can be aggregated together so that all the connected catalogs are receiving the same messages on new data registration, file status changes, etc. The usefulness of aggregating several catalogs can be illustrated by an example of a user community that wants to migrate the contents of their LFC catalog to the DFC catalog. In this case, the catalog data can be present in both catalogs for the time of migration or for redundancy purpose.

The DIRAC File Catalog has the following main features:

- Standard file catalog functionality for storing file metadata, ACL information, checksums, etc.
- Complete Replica Catalog functionality to keep track of physical copies of the files.
- Additional file metadata to define ancestor-descendent relations between files often needed for applications with complex workflows.

- Efficient storage usage reports to allow implementation of community policies, user quotas, etc.
- Metadata Catalog functionality to define arbitrary user metadata on directory and file levels with efficient search engine.
- Support for dataset definition and operations.

The DFC implementation is optimized for efficient bulk queries where the information for large numbers of files is requested in case of massive data management operations. Altogether, the DFC provides logical name space for the data and, together with storage access plug-ins, makes data access as simple as in a distributed file system.

Storage Element and File Catalog services are used to perform all the basic operations with data. However, bulk data operations need special support so that they can be performed asynchronously without a need for a user to wait for the operation completion at the interactive prompt. DIRAC Request Management System (RMS) provides support for such asynchronous operations. Many data management tasks in large scientific communities are often repeated for different data sets. DIRAC provides support for automation of recurrent massive data operations driven by the data registration or file status change events. Other data related services include:

In addition to the main DMS software stack, DIRAC provides several more services helping to perform particular data management tasks:

- Staging service to manage bringing data on-line into a disk cache in the SEs with tertiary storage architecture;
- Data Logging service to log all the operations on a predefined subset of data mostly for debugging purposes;
- Data Integrity service to record failures of the data management operations in order to spot malfunctioning components and resolve issues;
- The general DIRAC Accounting service is used to store the historical data of all the data transfers, success rates of the transfer operations.

### 2.3 DIRAC development framework

All the DIRAC components are written in a well-defined software framework with a clear architecture and development conventions. Since large part of the functionality is implemented as plug-ins implementing predefined abstract interfaces, extending DIRAC software to cover new cases is simplified by the design of the system. There are several core services to orchestrate the work of the whole DIRAC distributed system, the most important ones are the following:

- Configuration service used for discovery of the DIRAC components and providing a single source of configuration information;
- Monitoring service to follow the system load and activities;
- Accounting service to keep track of the resources consumption by different communities, groups and individual users;
- System Logging service to accumulate error reports in one place to allow quick reaction to occuring problem.

Modular architecture and the use of core services allow developers to easily write new extensions concentrating on their specific functionality and avoiding recurrent tasks.

All the communications between distributed DIRAC components are secure following the standards introduced by computational grids, which is extremely important in the distributed computing environment. A number of interfaces are provided to users to interact with the system. This includes a rich set of command-line tools for Unix environment, Python language API to write one’s own scripts and applications, RESTful interface to help integration with third party applications. DIRAC functionality is available also through a flexible and secure Web Portal which follows the user interface paradigm of a desktop computer.

## 3 DIRAC Users

DIRAC Project was initiated by the LHCb experiment at CERN. LHCb stays the most active user of the DIRAC software. The experiment data production system ensures a constant flow of jobs of different kinds: reconstruction of events of proton-proton collisions in the LHC collider, modelling of the LHCb detector response to different kinds of events, final user analysis of the data [8]. Figure 2. illustrates the scale of computing resources usage by the LHCb experiment. As it can be seen, there are on average about 50 thousands jobs running simultaneously on more than 120 sites, with peak values going to up to 100 thousands jobs. This is equivalent to operating a virtual computing centre of about 100 thousands of processor cores. At the same time the total data volume of LHCb exceeds 10 PB distributed over more than twenty millions of files, many of those having 2 and more physical copies in about 20 distributed storage systems. Information about all these data is stored in the DIRAC File Catalog. LHCb has created a large number of extensions to the core DIRAC functionality in order to support its specific workflows. All these extensions are implemented in the DIRAC development framework and can be released, deployed and maintained using standard DIRAC tools.

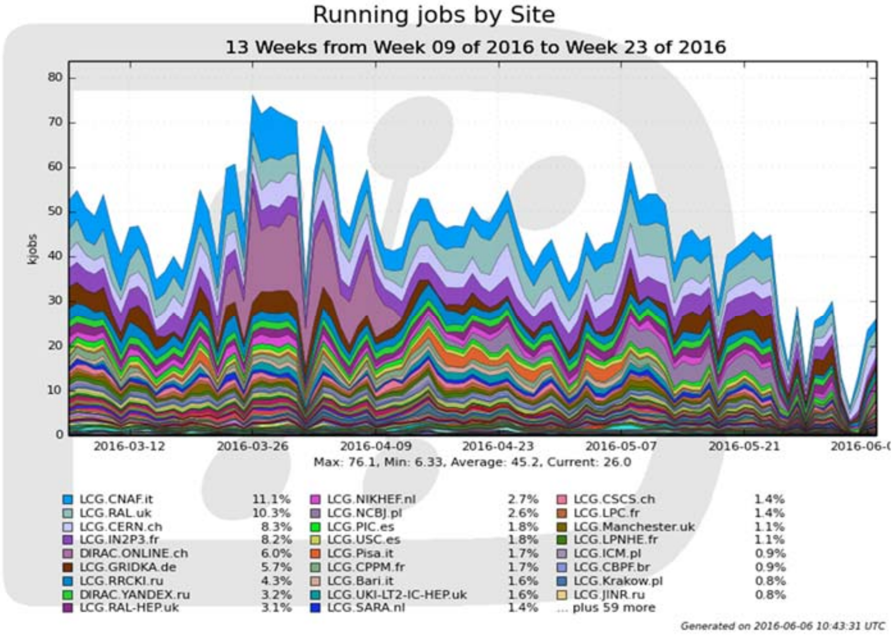

**Figure 2** Running jobs of the LHCb experiment

After the DIRAC system was successfully used within LHCb, several other experiments in High Energy Physics and other domains expressed interest in using this software for their data production systems, for example: BES III experiment at the BEPC collider in Beijing, China [9]; Belle II experiment at the KEK centre, Tsukuba, Japan [10]; the CTA astrophysics experiment being constructed now in Chile [11], and others. Open architecture of the DIRAC project was easy to adapt for the workflows of particular experiments. All of them developed several extensions to accommodate their specific requirements all relying on the use of the common core DIRAC services.

### 3.1 DIRAC as a service

Experience accumulated by running data intensive applications of the High Energy Physics experiments can be very valuable for researchers in other scientific domains, which have high computing requirements. However, if even the DIRAC client software is easy to install and use, running dedicated DIRAC services requires a high expertise level and is not easy especially for the research communities without long-term experience in large-scale computations. Therefore, several national computing infrastructure projects are offering now DIRAC services for their users. The first example of such service was created by the France-Grilles National Grid Initiative (NGI) project in France [12].

By 2011 in France, there were several DIRAC service installations used by different scientific or regional communities. There was also a DIRAC service maintained by the France-Grilles NGI as part of its training and dissemination program. This allowed several teams of experts in different universities to gain experience with installation and operation of DIRAC services. However, the combined maintenance effort for multiple DIRAC service instances was quite high. Therefore, it was proposed to integrate independent DIRAC installations into a single national service to optimize operational costs. The responsibilities of different partners of the project were distributed as follows. The France-Grilles NGI (FG) ensures the overall coordination of the project. The IN2P3 Computing Centre (CC/IN2P3) hosts the service providing the necessary hardware and manpower. The service is operated by a distributed team of experts from several laboratories and universities participating to the project.

From the start, the FG-DIRAC service was conceived for usage by multiple user communities. Now it is intensively used by researchers in the domains of life sciences, biomedicine, complex system analysis, and many others. It is very important that user support and assistance in porting applications to the distributed computing environments is the integral part of the service. This is especially needed for research domains where the computing expertise is historically low. Therefore, the France-Grilles NGI organizes multiple tutorials for interested users based on the FG-DIRAC platform. The tutorials not only demonstrate basic services capabilities but are also used to examine cases of particular applications and the necessary steps to start running them on distributed computing resources. The service has an active user community built around it and provides a forum where researchers are sharing their experience and helping the newcomers.

After the successful demonstration of DIRAC services provided by the French national computing research infrastructure, similar services were deployed in some other countries: Spain, UK, China, and some others. There are several ongoing evaluation projects testing the DIRAC functionality and usability for similar purposes. Since 2014, DIRAC services are provided by the European Grid Initiative (EGI) for the research communities in Europe and beyond [13].

A general-purpose DIRAC service was deployed in the Laboratory of Information Technologies in JINR, Dubna. This service is provided to users participating in international collaborations that already use DIRAC tools. It is also used for evaluation of DIRAC as a distributed computing platform for experiments that are now under preparation in JINR. The service is providing access to computing resources of the WLCG and EGI grid infrastructures. It has also several High Performance Computing (HPC) centers connected and offers a possibility to create complex workflows including massively parallel applications. The service is planned to become a central point for a federation of HPC centers in Russia and other countries. It will provide a framework for unified access to the HPC centers similar to existing grid infrastructures.

## 4 Conclusions

DIRAC interware is a versatile software suite for building distributed computing systems. It has gone a long way of development starting from a specific tool for a large-scale High Energy Physics experiment and is now available as a general-purpose product. Various computing and storage resources based on different technologies can be incorporated under the overall control by the DIRAC Workload and Data Management Systems. The open architecture of the DIRAC software allows easy connection of the new emerging types of resources as needed by the user communities. The system is designed for extensibility to support specific workflows and data requirements of particular applications. Completeness of its functionality as well as its modular design can ensure solution for a variety of distributed computing tasks and for a wide range of scientific communities in a single framework.

The number of DIRAC users is growing with the applications coming from various scientific domains. A number of multi-community DIRAC services are provided now by several national computing infrastructure projects are available to support small research communities not having dedicated systems for managing distributed computing resources. This helps many researchers without a deep special computing expertise level to get familiar with using distributed computing systems by following specialized tutorials and benefitting from assistance in porting their applications to this environment. Altogether, this makes large-scale data intensive computations more accessible improving the overall quality of their scientific results.

## References

[1] A. Tsaregorodtsev et al, DIRAC3 : The New Generation of the LHCb Grid Software, 2010 J. Phys.: Conf. Ser., 219 062029; DIRAC Project - http://diracgrid.org

[2] A.Casajus, R.Graciani, A.Tsaregorodtsev, DIRAC pilot framework and the DIRAC Workload Management System, 2010 J. Phys.: Conf. Ser. 219 062049

[3] OpenScience Grid - https://www.opensciencegrid.org/

[4] ARC project - http://www.nordugrid.org/arc/

[5] A.Smith, A.Tsaregorodtsev, DIRAC: data production management, 2008 J. Phys.: Conf.Ser. 119 062046

[6] Gfal2 Project - https://dmc.web.cern.ch/projects-tags/gfal-2

[7] S. Poss and A. Tsaregorodtsev, DIRAC File Replica and Metadata Catalog, 2012 J. Phys.: Conf.Ser. 396 032108

[8] F. Stagni F and Ph. Charpentier, The LHCb DIRAC-based production and data management operations systems, 2012 J. Phys.: Conf. Ser. 368 012010

[9] X.M. Zhang, I. Pelevanyuk, V. Korenkov et al, Design and Operation of the BES-III Distributed Computing System, 2015 Procedia Computer Science 66

[10] T.Kuhr, T.Hara, Computing at Belle II, 2015 J. Phys.: Conf. Ser. 396 032063

[11] L.Arrabito et al, Application of the DIRAC framework in CTA: first evaluation, 2015 J. Phys.: Conf. Ser. 396 032007

[12] France-Grilles DIRAC portal – http://dirac.france-grilles.in2p3.fr

[13] DIRAC4EGI service portal – http://dirac.egi.eu

[14] A.Klimentov et al, Next Generation Workload Management System For Big Data on Heterogeneous Distributed Computing, 2015 J. Phys.: Conf. Ser. 608 012040

[15] S. Bagnasco, L. Betev, P. Buncic et al, AliEn: ALICE environment on the GRID, 2008 J. Phys.: Conf. Ser. 119 062012