# Hybrid Distributed Computing Service Based on the DIRAC Interware

Victor Gergel[1], Vladimir Korenkov[2,3], Igor Pelevanyuk[2], Matvey Sapunov[4],
Andrei Tsaregorodtsev[3,5](✉), and Petr Zrelov[2,3]

[1] Lobachevsky State University, Nizhni Novgorod, Russia
gergel@unn.ru
[2] Joint Institute for Nuclear Research, Dubna, Russia
{korenkov,pelevanyuk,zrelov}@jinr.ru
[3] Plekhanov Russian Economics University, Moscow, Russia
[4] Aix-Marseille University, Marseille, France
matvey.sapunov@univ-amu.fr
[5] CPPM, Aix-Marseille University, CNRS/IN2P3, Marseille, France
atsareg@in2p3.fr

**Abstract.** Scientific data intensive applications requiring simultaneous use of large amounts of computing resources are becoming quite common. Properties of applications coming from different scientific domains as well as their requirements to the computing resources are varying largely. Many scientific communities have access to different types of computing resources. Often their workflows can benefit from a combination of High Throughput (HTC) and High Performance (HPC) computing centers, cloud or volunteer computing power. However, all these resources have different user interfaces and access mechanisms, which are making their combined usage difficult for the users. This problem is addressed by projects developing software for integration of various computing centers into a single coherent infrastructure, the so-called interware. One of such software toolkits is the DIRAC interware. This product was very successful to solve problems of large High Energy Physics experiments and was reworked 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. One of these services is deployed at Joint Institute for Nuclear Research, Dubna. It aims at integration of computing resources of several grid and supercomputer centers as well as cloud providers. An overview of the DIRAC interware and its use for creating and operating of a hybrid distributed computing system at JINR is presented in this article.

**Keywords:** Grid computing · Hybrid distributed · Computing systems · Supercomputers · DIRAC

## 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

L. Kalinichenko et al. (Eds.): DAMDID/RCDL 2016, CCIS 706, pp. 105–118, 2017.
DOI: 10.1007/978-3-319-57135-5_8

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.

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 their own systems 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 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 [2] or AliEn [3] 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.

This paper is an extension of the contribution to proceedings of the DAMDID’2016 Conference [15] providing a detailed description of the DIRAC service deployed in the Laboratory of Information Technologies in the Joint Institute for Nuclear Research (LIT/JINR), Dubna, Russia. The paper reviews the DIRAC Project giving details about its general architecture as well as about workload and data management capabilities in Sect. 2. Examples of the system usage are described in Sect. 3. Section 4 presents details of the DIRAC service deployment in LIT/JINR describing various components of the system as well as currently connected computing resources.

## 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.

### 2.1 Workload Management

The DIRAC Workload Management System is based on the concept of pilot jobs [4]. In this scheduling architecture (Fig. 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.

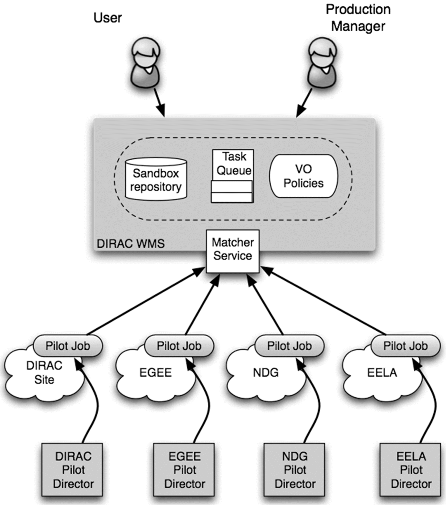

**Fig. 1.** WMS with pilot jobs

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 [5].
- Grids based on the ARC middleware, which was developed in the framework of the Nordugrid project [6].
- 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. Computing resources of some of those types are integrated within the JINR DIRAC service as described in Sect. 4.

## 2.2 Data Management

The DIRAC Data Management System (DMS) is based on similar design principles as the WMS [7]. 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.) [8].

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 [9]. This 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.

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 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 occurring 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, modeling of the LHCb detector response to different kinds of events, final user analysis of the data [10]. The DIRAC Workload Management System ensures on average about 50 thousand jobs running simultaneously on more than 120 sites, with peak values going to up to 100 thousand jobs. This is equivalent to operating a virtual computing center of about 100 thousands of processor cores. LHCb exploits traditional grid computing resources, several standalone computing farms and also resources from several public cloud providers. An important special ingredient to the LHCb computing system is the High Level Trigger (HLT) farm, which is used for a fast "on-line" filtering of events during the normal data taking periods. In the periods when the LHC collider is stopped for maintenance operations, the HLT farm is used for modeling of the LHCb detector providing the total power of up to 35 thousands CPU cores. No particular batch system is used to manage those CPUs, which are under full control of the DIRAC WMS. 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 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.

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 [11]; Belle II experiment at the KEK center, Tsukuba, Japan [12]; the CTA astrophysics experiment being constructed now in Chile [13], 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 [14].

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 [15].

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 [16].

## 4 DIRAC in Joint Institute for Nuclear Research

A general-purpose DIRAC service was deployed in the Laboratory of Information Technologies (LIT) in JINR, Dubna. The Joint Institute for Nuclear Research (JINR) - is an international research center for nuclear sciences, with 5500 staff members, 1200 researchers including 1000 Ph.D's from eighteen member states countries. JINR participates in many experiments in physics domain all over the world. The DIRAC in JINR 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 has 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.

### 4.1 Computing Needs of JINR Experiments

The need to provide limited computing power and data storage occasionally appears from time to time. It could be at foundation of a new experiment like Nuclotron-based Ion Collider facility (NICA) [17], grows of existing one, like Baikal Deep Underwater

Neutrino Telescope (Baikal) [18] or decision of scientific group to move from computations on local PC's to more powerful dedicated computing resource. In JINR, there are several ways to provide necessary computing resources.

The first option is to use the batch system of the local computing center for calculations and storage. This is usually a good option for small groups and projects but it has the following constraints: access allowed only for user members in JINR, there is no guarantee that batch system will be able to process certain amount of tasks by certain time and command line interface which is the only way to work with batch could be a deterrent to some users.

The second option is a creation of a new dedicated computing infrastructure. This option could lead to big paper work, creation of a prototype and further upgrading in case of approval. Apart from that the one who is building such infrastructure should take into account that dedicated computing infrastructures tend to be idle considerable amounts of time and require many man power to maintain it. Usually only big scientific groups or laboratories could use this option.

DIRAC software was adopted in JINR as a viable solution to provide computing power and storage for different users. From the beginning the DIRAC service in JINR was designed to serve multiple experiments or user groups to reduce overall maintenance costs. Another point to use one DIRAC installation for different groups, or in terms of DIRAC Virtual Organizations, is the fact that under one installation it is possible and recommended to delegate management of all resource sharing policies to DIRAC and by doing that both reduce amount of work of a resource administrator and decrease the resource idle time.

Another big advantage of using DIRAC is the possibility to manage both storage and computing resources, which provides "single point of management" when it comes to pledges distribution and users and groups administration. Freed man-power could be invested either in development for particular Virtual Organization or in the development for the particular DIRAC installation which by itself could lead to the benefit of the whole DIRAC community. Also, using DIRAC across several groups could simplify user training and give a unified access to very different computing resources which users would never get in different circumstances like clouds, batch systems, grid computing elements or even supercomputers.

Despite the fact that administration work is reduced, operating the DIRAC service cannot be performed by any particular user group alone and should be handled centrally. In JINR, the Laboratory of Information Technologies is taking responsibility for maintaining and administrating the DIRAC service. The Laboratory of Information Technologies by itself has some amount of computing and storage resources which could be used for evaluation DIRAC by possible users and small groups but in case of serious requirements could be easily extended with additional external computing or storage capacity.

DIRAC is an open source software with the core written in Python and web part written in JavaScript. It is possible to modify, expand and tune DIRAC for the specific purposes in JINR. Since JINR participates in the BES III experiment [19], in particular contributing to construction of its computing system, a good expertise level in DIRAC [9] was developed, which helped to launch the general-purpose DIRAC service.

### 4.2 DIRAC Service in JINR

The DIRAC service at JINR is set up to aggregate various computing facilities in Dubna and at the sites of its partners in Russia and abroad. It is open for connection of centers of different types and at the point of writing includes five computing elements which are described in more details in the following.

The service is hosted on 3 virtual servers provided in the JINR cloud infrastructure:

**dirac-conf.jinr.ru.** This server for configuration and security services. Those services are security sensitive and require restricted access; therefore they are placed on a dedicated host. The following services are deployed on this server:

- The master Configuration Service is the primary database for all the DIRAC configuration data including description of computing resources, users and groups, community policies. This is also the primary tool for discovery of all the DIRAC distributed components;
- Several permanently running agents to synchronize the DIRAC Configuration Service with third party information sources, for example, information indices of grid computing infrastructures;
- User credential repository serves user identity information for the components that need to perform operation on behalf of particular users;
- Security logging service keeps track of all the authenticated connections to any DIRAC service to provide information necessary to investigate eventual security incidents;

**dirac-services.jinr.ru.** All the services supporting workload and data management are hosted on this server:

- Job scheduling and monitoring services include all the components to process user payloads from the initial reception up to delivering of the final results to users;
- Data catalogs to keep track of user file physical locations and ensure easy data discovery with user defined metadata;
- Services to support data operations like massive data replication and removal, data integrity checking and others;
- Computing resources monitoring and accounting services providing detailed information on the history of resources consumption

**dirac-ui.jinr.ru.** This server is hosting the DIRAC Web Portal and the RESTful API service. For the safety reasons, those components are separated from the core DIRAC services to reduce potential risks. In particular, these services do not have direct access to databases keeping the state of the whole system.

### 4.3 Computing Resources

The described services allow users to execute their payloads on various computing facilities which could be local in JINR or remote. One of the goals of the project is to collect in one system computing resources with different properties to demonstrate feasibility and advantages of providing a single user interface for all of them. In the following we present a brief description of these facilities.

**DIRAC.JINR.ru** represents the HybriLIT supercomputer cluster at the JINR Computing Center [20]. This cluster allows execution of parallel applications using MPI exchanges between the threads. It also supports calculations using accelerator co-processors, like GPGPU and others;

**DIRAC.JINR-PBS.ru** is a small cluster of commodity computers used by several experiments and groups in JINR;

**DIRAC.JINR-SSH.ru** is a collection of UNIX hosts not integrated in any computing cluster. The hosts are simply described in the DIRAC Configuration System with their IP addresses and SSH connection details. In this case, the DIRAC workload Management System plays the role of the batch system for this ad hoc computing cluster.

**DIRAC.NNGU.ru** represents the Lobachevsky Supercomputer of the University of Nizhny Novgorod [21]. A peculiarity of access to this facility is that it is done through a VPN tunnel in addition to the SSH secure access to the batch system gateway. Only a small subset of the supercomputer is accessible to the JINR DIRAC service, however, it demonstrates the possibility of remote connection with yet another set of access protocols;

**DIRAC.AMU.fr** is the supercomputer center of the Aix-Marseille University, France [22]. It supports parallel MPI applications and applications requiring the use of GPGPU's. Access is allowed to the local queues on the best effort basis when no other waiting priority tasks are available, which is very often the case and makes this center a valuable "opportunistic" resource.

More resources are planned for integration with the JINR DIRAC service to allow better possibilities for the users. In particular, resources provided by the JINR Openstack cloud will be an important use case for the purpose of demonstration of unified access for the users (Table 1).

**Table 1.** The computing facilities connected to the DIRAC service in JINR

| Name | Access method | Batch system | Location | HPC support |
|---|---|---|---|---|
| DIRAC.JINR.ru | SSH | SLURM | JINR | yes |
| DIRAC.JINR-PBS.ru | SSH | PBS | JINR | no |
| DIRAC.JINR-SSH.ru | SSH | DIRAC | JINR + other | no |
| DIRAC.NNGU.ru | VPN + SSH | SLURM | Nizhny Novgorod, Russia | yes |
| DIRAC.AMU.fr | SSH | OAR | Aix-Marseille University, France | yes |

## 5 Conclusions

DIRAC interware [23] 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 DIRAC service in JINR is deployed with the goal to provide a platform for integration of heterogeneous computing resources with a simple and uniform interface for the users of experiments at JINR as well as for other user communities. It also plays the role of research and development facility to explore the possibility to integrate seemingly very different computing resources with different management systems and access methods. In particular, integration in one infrastructure of HTC and HPC computing resources helps to improve the efficiency of the resources usage and opens opportunities to create complex application workflows with different steps executed on different types of computers mostly suitable for the corresponding applications. Local and remote computing centers of different kinds are already connected to the service and demonstrated to work together with a single user interface. More resources, including clouds are planned for integration, which will provide more opportunities for the users of the service.

## References

1. Tsaregorodtsev, A., Brook, N., Casajus Ramo, A., et al.: DIRAC3: the new generation of the LHCb grid software. J. Phys: Conf. Ser. **219**, 6 (2010). doi:10.1088/1742-6596/219/6/062029
2. Klimentov, A., Buncic, P., De, K., et al.: Next generation workload management system for big data on heterogeneous distributed computing. J. Phys: Conf. Ser. **608**, 1 (2015). doi: 10.1088/1742-6596/608/1/012040
3. Bagnasco, S., Betev, L., Buncic, P., et al.: AliEn: ALICE environment on the grid. J. Phys: Conf. Ser. **119**, 6 (2008). doi:10.1088/1742-6596/119/6/062012
4. Casajus Ramo, A., Graciani, R., Tsaregorodtsev, A.: DIRAC pilot framework and the DIRAC workload management system. J. Phys: Conf. Ser. **219**, 6 (2010). doi: 10.1088/1742-6596/219/6/062049
5. OpenScience Grid. https://www.opensciencegrid.org/
6. ARC project. http://www.nordugrid.org/arc/
7. Smith, A., Tsaregorodtsev, A.: DIRAC: data production management. J. Phys: Conf. Ser. **119**, 6 (2008). doi:10.1088/1742-6596/119/6/062046
8. Gfal2 Project. https://dmc.web.cern.ch/projects-tags/gfal-2
9. Poss, S., Tsaregorodtsev, A.: DIRAC file replica and metadata catalog. J. Phys: Conf. Ser. **396**, 3 (2012). doi:10.1088/1742-6596/396/3/032108
10. Stagni, F., Charpentier, P.: The LHCb DIRAC-based production and data management operations systems. J. Phys: Conf. Ser. **368**, 1 (2012). doi:10.1088/1742-6596/368/1/012010
11. Zhang, X.M., Pelevanyuk, I., Korenkov, V., et al.: Design and operation of the BES-III distributed computing system. Procedia Comput. Sci. **66**, 619–624 (2015). doi:10.1016/j.procs.2015.11.070
12. Kuhr, T.: Computing at Belle II. J. Phys: Conf. Ser. **396**, 3 (2015). doi: 10.1088/1742-6596/396/3/032063

13. Arrabito, L., Barbier, C., Graciani Diaz, R., et al.: Application of the DIRAC framework in CTA: first evaluation. J. Phys: Conf. Ser. **396**, 3 (2015). doi: 10.1088/1742-6596/396/3/032007
14. France-Grilles DIRAC portal. http://dirac.france-grilles.in2p3.fr
15. Korenkov, V., Pelevanyuk, I., Tsaregorodtsev, A., Zrelov, P.: Accessing distributed computing resources by scientific communities using DIRAC services. In: The XVIII International Conference on Data Analytics & Management in Data Intensive Domains (DAMDID/RCDL 2016), Ershovo, Moscow Region, Russia, CEUR Workshop Proceedings, vol. 1752, pp. 110–115 (2016)
16. DIRAC4EGI service portal. http://dirac.egi.eu
17. Trubnikov, G., Agapov, N., Alexandrov, V., et al.: Project of the nuclotron-based ion collider facility (NICA) at JINR. In: EPAC 2008 Conference Proceedings, Genoa, Italy (2008)
18. Wischnewski, R.: The Baikal neutrino telescope – results and plans. Int. J. Modern Phys. A **20**(29), 6932–6936 (2005). doi:10.1142/S0217751X0503051X
19. Zweber, P.: Charm factories: present and future. AIP Conf. Proc. **1182**(1), 406–409 (2009)
20. Alexandrov, E., Belyakov, D., Matveyev, M., et al.: Research of acceleration of calculation in solving scientific problems on the heterogeneous cluster HybriLIT. RUDN J. Math. Inf. Sci. Phys. **4**, 20–27 (2015)
21. Lobachevsky supercomputer. https://www.top500.org/system/178472
22. Mesocentre of Aix-Marseille University. http://mesocentre.univ-amu.fr/en
23. DIRAC Project. http://diracgrid.org