


# Efficient Distributed Computations with DIRAC

Viktor Gergel<sup>1</sup>, Vladimir Korenkov<sup>2,3</sup>, Andrei Tsaregorodtsev<sup>3,4</sup>,  
and Alexey Svistunov<sup>1</sup>

<sup>1</sup> Lobachevsky State University of Nizhni Novgorod, Nizhni Novgorod, Russia  
gergel@unn.ru, alexey.svistunov@itmm.unn.ru

<sup>2</sup> Joint Institute for Nuclear Research, Dubna, Russia  
korenkov@jinr.ru

<sup>3</sup> Plekhanov Russian University of Economics, Moscow, Russia

<sup>4</sup> CPPM, Aix Marseille Université, CNRS/IN2P3, Marseille, France  
atsareg@cppm.in2p3.fr

**Abstract.** High Energy Physics (HEP) experiments at the LHC collider at CERN were among the first scientific communities with very high computing requirements. Nowadays, researchers in other scientific domains are in need of similar computational power and storage capacity. Solution for the HEP experiments was found in the form of computational grid - distributed computing infrastructure integrating large number of computing centers based on commodity hardware. These infrastructures are very well suited for High Throughput applications used for analysis of large volumes of data with trivial parallelization in multiple independent execution threads. More advanced applications in HEP and other scientific domains can exploit complex parallelization techniques using multiple interacting execution threads. A growing number of High Performance Computing (HPC) centers, or supercomputers, support this mode of operation. One of the software toolkits developed for building distributed computing systems is the DIRAC Interware. It allows seamless integration of computing and storage resources based on different technologies into a single coherent system. This product was very successful to solve problems of large HEP experiments and was upgraded in order to offer a general-purpose solution. The DIRAC Interware can help including also HPC centers into a common federation to achieve similar goals as for computational grids. However, integration of HPC centers imposes certain requirements on their internal organization and external connectivity presenting a complex co-design problem. A distributed infrastructure including supercomputers is planned for construction. It will be applied for inter-disciplinary large-scale problems of modern science and technology.

**Keywords:** Distributed computing · High-performance computations · Cloud services · Grid systems · Workflow management · Big data management

## 1 Introduction

The number of scientific domains with highly intensive computational applications is rapidly increasing. The High Energy Physics (HEP) experiments at the LHC collider,

CERN, have pioneered the new era of highly data intensive studies. However, other disciplines are quickly increasing their data volume requirements. Applications dealing with Exabyte-level data volumes are already on the horizon. New scientific communities need urgently tools to work with large datasets and massively parallel applications adapted to their specific tasks and suitable to the expertise level of their scientists. The scientific collaborations nowadays are often international with many groups coming from different laboratories and universities. As a result, the available computing and storage resources of a given collaboration are usually distributed as each group is coming up with its own contribution. Therefore, there is a strong necessity of building computing systems that cope with large volumes of distributed data and distributed computing resources that can be used for these data analysis.

The DIRAC Project was started to solve the data intensive analysis problem for one of the LHC experiments, LHCb, in 2003 [1, 2]. It was started as a Workload Management System (WMS) in order to operate multiple computing centers in Europe to produce modeling data for the experiment optimization. However, the need in an efficient Data Management System coping with many millions of files with distributed replicas and having a close coupling with the WMS was quickly understood. As a result, the DIRAC allows performing all the data analysis tasks of LHCb and other HEP experiments [4–6].

After multiple years of successful usage in the HEP domain, the DIRAC software was generalized to be suitable for other applications requiring large data volumes and computing power. It provides a development framework and many ready-to-use services to build distributed computing systems adapted to particular scientific communities. These tools serve to interconnect technologically heterogeneous computing and storage resources into a coherent system seen by the users as a single large computer with a friendly interface and consistent computational and storage subsystems. Therefore, we speak about DIRAC Interware – technology to aggregate multiple computing resources and services. This toolkit can be also used to integrate computing centers of the HPC type supporting massively parallel applications along with the traditional grid sites. This requires development of several new components and also a model of an HPC center, which is rich enough for a large number of applications that can run in such HPC federations.

In this article we will overview the DIRAC Interware, its base architecture and implementation. We will describe the base Workload Management and Data Management systems of DIRAC as well as computing and storage resources accessible with these services. We will discuss specific features of HPC centers and possible ways to integrate them into a comment distributed infrastructure. We will present also plans for integration of several HPC centers into a federation dedicated to actual scientific problems.

## 2 DIRAC Overview

DIRAC Project provides all the necessary components to create and maintain distributed computing systems. It is forming 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, the 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 [3]. 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.

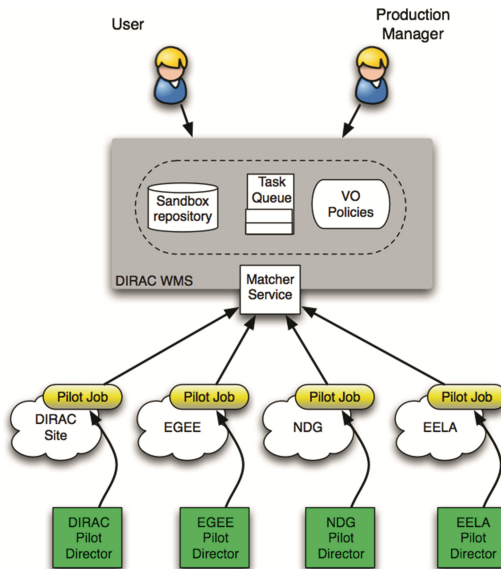


Fig. 1. WMS with pilot jobs

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. Examples of such grid infrastructures are WLCG and EGI grids.
- Open Science Grid (OSG) infrastructure based on the VDT (Virtual Data Toolkit) suite of middleware [7].
- Grids based on the ARC middleware which was developed in the framework of the Nordugrid project [8].
- 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 that will be used by DIRAC directors to submit pilot jobs to the local batch systems.
- 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 resource 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 allows connecting 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 [9]. An abstract interface is defined to describe access to a storage system and there are 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 storage system can be accessible via different interfaces with different access protocols. But for the users this stays logically a single service providing access to the same

physical storage space. Similar situation can happen also for the file catalog services. To simplify access to this kind of services, DIRAC defines aggregators that allow working with multiple services as if with a single one from the client perspective. All the plug-ins and aggregators are hidden behind the DataManager API which have methods to perform all the basic operations needing access to both storage and catalog services.

DIRAC is also providing a number of auxiliary and higher level services to support higher-level operations as well as to help administrators to run the system:

- Support for bulk asynchronous operations is provided by the Request Management System (RMS);
- Transformation System (TS) provides means to automate recurrent massive data operations driven by the data registration or file status change events;
- Staging service to manage bringing data on-line into a disk cache in the SEs with tertiary storage architecture. These operations are usually triggered automatically by the WMS before the jobs using these data as input can be submitted for execution to the worker nodes.
- FTS Manager service to submit and manage data transfer requests to an external File Transfer Service.
- 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, etc.

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.) [10].

If some DIRAC user community would need access to a storage system not yet supported by the DIRAC Interware, it will be easy to incorporate it by providing a new plug-in to the system.

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, for example NFS or Lustre. 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.

### 2.3 DIRAC Development Framework

All the DIRAC components are written in a well-defined software framework with a clear architecture and development conventions. A large part of the functionality is

implemented as plugins implementing predefined abstract interfaces. 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 be able to quickly react to problems.

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.

Users are provided with a number of different interfaces 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. DIRAC functionality is available also through a flexible and secure Web Portal which follows the user interface paradigm of a desktop computer.

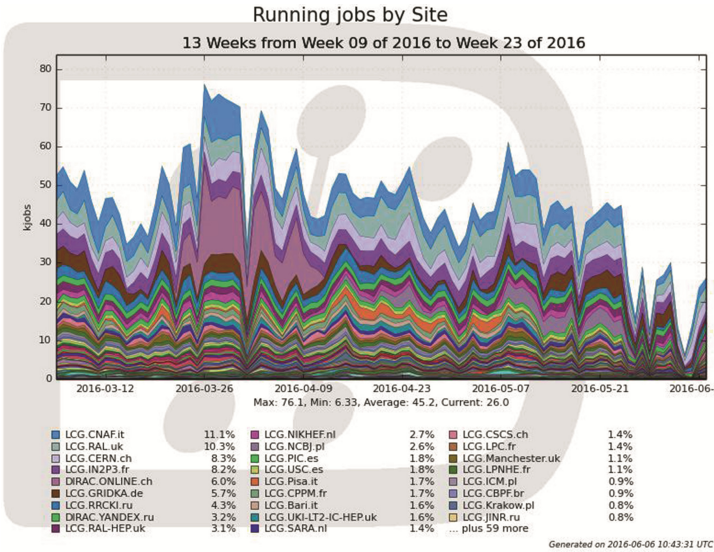
### 3 DIRAC Usage Examples

DIRAC based infrastructures are used by multiple scientific communities having to integrate heterogeneous resources at their disposal. Many of the common requirements are already satisfied by the core DIRAC components. However, each community can have its own specific workflows and data models. Therefore it is quite usual that large experiments are introducing new services implementing their particular management logic.

#### 3.1 Physics Applications

DIRAC was originally developed for the LHCb experiments at the LHC collider at CERN, Geneva. Among the High Energy Physics experiments, LHCb stays the most intensive user of the DIRAC Interware using it as the basis for its data production system [12]. Figure 2 illustrates the scale of the computing resources usage by LHCb.

The plot is produced by the DIRAC Accounting system and shows that on average the LHCb data production system is controlling about 50 thousands of simultaneous jobs running at more than 100 distributed computing centers. This is equivalent to running a virtual distributed computing center of up to 100 thousands CPU cores. The LHCb data volume reaches about 40 PBytes spread over more than 20 data centers in Europe and Russia. LHCb is using mostly resources provided by the WLCG computing



**Fig. 2.** Simultaneously running distributed LHCb jobs

grid infrastructure [13]. However, it also incorporates several large non-grid centers, such as Ohio Supercomputing Center in USA or Yandex computing farm in Russia. Those centers are incorporated seamlessly using the DIRAC Interware. LHCb is using all the DIRAC core services for managing workflows and data but it has also developed several specific ones, like for example, Bookkeeping service for storing all the data provenance information, or Production service for managing large numbers of tasks and files in an automated way. All the LHCb specific services are developed within the DIRAC Framework as extensions and thus reuse multiple core APIs.

Another example is the Belle II experiment at KEK, Tsukuba, Japan. This was the first experiment to start using DIRAC outside LHCb [5]. The initial requirement of the Belle Collaboration was the possibility to incorporate commercial cloud resources provided by the Amazon Company. The VMDIRAC subsystem was initiated as a DIRAC extension to manage computing resources coming from various cloud providers. Now it is making part of the DIRAC core services and other user communities can benefit from it.

The BES III experiment at IHEP, Beijing, China is one more HEP experiment using DIRAC for its production system [14]. In particular, IHEP developers contributed several modules to the DIRAC File Catalog service, for example, Dataset modules for managing large collections of files as a single entity. The DIRAC service installation for the BES III experiment in IHEP was recently upgraded to support multiple user communities, like the Juno experiment or the CEPC project [15].

### 3.2 Multi-domain DIRAC Services

The success of DIRAC for supporting large scientific user communities suggested the idea that DIRAC services can be also offered to smaller research groups without the need to install and maintain complicated software and hardware systems. Indeed many small groups, often without deep knowledge of the distributed computing matters, still need access to large computing infrastructures for their application. Therefore, DIRAC services were offered as part of several distributed computing infrastructure projects [16]. The first such service was provided by the France-Grilles National Grid Infrastructure (NGI) project in 2012. Now it serves about 20 different grid Virtual Organizations. For example, users from the international biomed Virtual Organization submit more than a half of their payloads through the FG-DIRAC service in France.

Since 2014, the DIRAC4EGI service is offered by the European Grid Infrastructure (EGI) Project. Several communities representing various scientific domains like life sciences, climatology and others use this service. The service is also intensively used for dissemination purposes, for example for tutorials on using distributed grid and cloud computing resources.

## 4 Federation of HPC Centers

Several examples of successful incorporation of HPC centers dedicated to massively parallel applications into a common distributed infrastructure including grid, cloud and stand-alone centers showed that it is possible to create a dedicated system to federate multiple HPC sites based on the DIRAC Interware technology. This will offer a full potential of these centers to large scientific communities that require more and more this kind of resources for their applications.

It is important to mention that combining grid computing centers together with the HPC centers can be very useful for communities with very complex workflows where some steps can be executed on a cheaper grid computing elements and others on HPC ones. Such optimization can reduce the time and the cost of the overall workflow execution.

### 4.1 Open Distributed Supercomputer Infrastructure Project

A project for construction of an Open Distributed Supercomputer Infrastructure (ODSI) will have to carry out several tasks. First of all, the concept of the ODSI must be formulated, which includes several aspects:

- Develop a model of an HPC center that will be as much in common for all the involved sites as possible. For each site, this model will be described in the system configuration with the site-specific parameters. As a result this will allow to present all the HPC centers as logical resources for the users that can be used in a transparent interchangeable manner;



- Develop efficient algorithms for managing large numbers of tasks executed in heterogeneous computing environment including HPC centers, which optimize the usage of computing and storage resources and minimize the overall execution time;
- Develop the necessary new DIRAC components to support the HPC specific workflows and reuse as much as possible the already existing tools. This will allow seamless migration for the DIRAC users to the new type of resources;
- Formulate common policies of usage of the HPC centers by large distributed user communities and implement tools to support those policies.

Building the ODSI infrastructure will need going through a number of prototypes involving an increasing number of HPC centers first on the national and then on the international levels. Several research laboratories and universities in Russia (JINR, Dubna, University of Nizhni Novgorod, and others) are planning to undertake such project. As a result it will create the infrastructure for solving a number of inter-disciplinary large-scale problems of the modern science and technology, which are already selected as the project pilot applications [17–19].

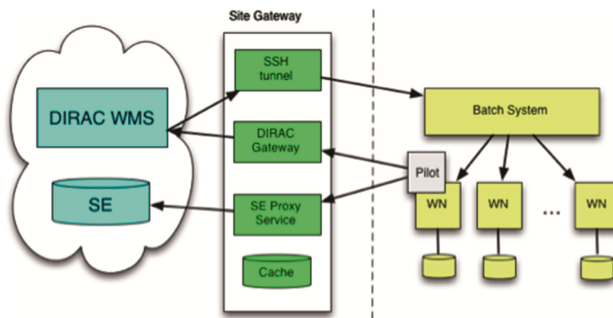
## 4.2 Co-design of a Federated HPC Supercomputer

In order to be included into the ODSI infrastructure an HPC center must follow several design requirements to ensure homogeneous access and security rules. Integration of traditional computing centers is relatively simple, especially those that participate in grid infrastructures. The HPC centers are in most cases designed and deployed without plans for eventual participation in any federation project. Therefore, their organization has little in common, which makes their integration difficult. The pilot job based WMS offers opportunities that can be very helpful in such projects because it does not require running complicated services on sites.

**Interaction with the DIRAC Central Services.** WMS with pilot jobs assumes outbound connectivity from the worker nodes. This is necessary to let pilots interact with the central services to report their status and request user payloads. If a computing center allows worker node outbound connectivity, then its connection to a DIRAC infrastructure is similar to traditional centers and requires minimal effort from the site administrators. However, a majority of HPC centers forbid such outbound connectivity for various reasons. In this case, DIRAC proposes a special service – Gateway – that can run on a HPC site gatekeeper host and serve as a proxy to pass messages from pilot jobs to the central services. Using this service requires a minor change in the pilot configuration on such sites while fully preserving the overall architecture and logic. The HPC center in this case must provide the gatekeeper host with appropriate dual external/local network connectivity. The host throughput capacity should be sufficient to support the possibly rather intensive traffic of data being produced or analyzed in the center. The security requirements to this host are very strict, as its certificate will be trusted by the DIRAC services as representing users whose jobs are running in the center.

Another problem of running jobs in the HPC centers with limited worker node connectivity is exporting the resulting data. If the data cannot be sent out directly from the worker nodes, this can be achieved by means of the Storage Element Proxy service.

This service allows access to any Storage Element from the machines not having the necessary software for corresponding plug-ins or other limitations. In this case, the client is accessing the Storage Element Proxy service with the DIRAC native DIPS protocol and the service transmits the access request to the destination SE with the suitable protocol. The client credentials are checked and used to access the destination service by delegation. In the case of running user jobs in computing centers where worker nodes do not have access to the WAN and therefore can not upload the resulting data directly, running the Storage Element Proxy service in the Gateway host of the computing center can help to export data from the worker nodes without a need to use some intermediate buffer storage and transfer data asynchronously by some additional agent or a *cron* job. Putting this all together, Fig. 3 illustrates the general scheme of connecting an HPC center to a DIRAC-based infrastructure.



**Fig. 3.** Pilot job interaction with DIRAC central services in case of no outbound connectivity in the worker nodes

**Multiple CPU Slot Reservation.** Applications running at HPC centers usually use multiple processors or even multiple worker nodes together. The reservation of multi-host computing slots is a complicated task and can be done by means of the local batch system scheduler, for example SLURM, OAR, or others. The pilot based WMS can exploit the tools offered by the target batch system but it can also offer other interesting opportunities here.

Computing slots reserved by the pilots can be orchestrated by a central DIRAC MPI service [11]. This service keeps track of all the groups of pilots that can work together to run parallel applications. These groups are combining pilots that are running on worker nodes on the same high performance local network, which allows exchanges using some variation of the MPI protocol. Accumulation of such pilot groups that can eventually constitute an MPI ring is a rather complicated and time-consuming process. The computing slots that are freed by previously running jobs are blocked by the workload management system in order to satisfy requirements of the jobs in its waiting queue and accumulate the necessary capacity. While the multi-processor slot is being accumulated, the constituent processors stay idle decreasing the overall efficiency. Therefore, the accumulated group of slots is a very valuable asset that should be used as much efficiently as possible. With the pilot jobs coordinated by the DIRAC MPI service such

multi-worker reservations can be reused for multiple user payloads without the need to redo the multi-slot reservations. As a result, this can increase dramatically the efficiency of the usage of the HPC resources.

In a batch system, the computing slot is reserved for a limited amount of time in order to ensure a fair sharing of resources among different tasks and users. However, worker nodes reserved by the DIRAC WMS can execute multiple jobs coming from different users and ensuring fair sharing on the meta-scheduler level. This mode of operation has many advantages. It puts less load on the local batch system scheduler and increases the efficiency of resources usage. However, administration of the batch system may require stopping the worker nodes from time to time to perform maintenance tasks, e.g. software or hardware upgrades. In this case, the DIRAC pilots occupying the worker nodes should receive signals from the batch system ordering the node liberation. The signals should be well specified and the corresponding handlers should be included into the DIRAC pilots. The handlers will then ensure graceful finalization of the running user applications avoiding losses of the job results that can happen in case of abrupt killing of the batch jobs. The design of the batch system signals and of the pilot signal handlers requires a close cooperation between the HPC centers administrators and developers of the DIRAC software.

## 5 Conclusions

The DIRAC Interware provides a framework and a rich set of services to build distributed computing systems. Such systems are successfully used for a number of High Energy Physics and AstroPhysics experiments, but also for other applications in different scientific domains. The Workload Management System with pilot jobs proved to be very efficient to control user tasks in a High Throughput environment. However, it can be also applied for aggregation of the HPC computing resources. The pilot job scheduling paradigm can increase significantly the scheduling efficiency for parallel applications requiring multi-processor computing slots. Combining traditional, cloud and HPC computing centers in a single distributed infrastructure can allow execution of complex workflows needing different types of resources on different subsequent steps. As a result, this can increase the overall efficiency of the usage of otherwise heterogeneous computing resources.

Building an Open Distributed Supercomputer Infrastructure aggregating multiple HPC centers in Russia and abroad can bring the support for massively parallel applications to a new level. This will make the supercomputer resources elastic from the user perspective, which means that much more power can be provided momentarily for a given application when it is actually needed. On the other hand it will dramatically increase the usage efficiency of multiple HPC centers.

## References

1. Tsaregorodtsev, A., et al.: DIRAC3: the new generation of the LHCb grid software. *J. Phys. Conf. Ser.* **219**, 062029 (2010)

2. DIRAC Project. <http://diracgrid.org>
3. Casajus, A., Graciani, R., Tsaregorodtsev, A.: DIRAC pilot framework and the DIRAC Workload Management System. *J. Phys. Conf. Ser.* **219**, 062049 (2010)
4. BES III Collaboration. <http://bes.ihep.ac.cn/bes3>
5. Kuhr, T., Hara, T.: Computing at Belle II. In: Proceedings of the CHEP 2012 International Conference, New-York, May 2012
6. Arrabito, L., et al.: Application of the DIRAC framework in CTA: first evaluation. In: Proceedings of the CHEP 2012 International Conference, New-York, May 2012
7. OpenScience Grid. <https://www.opensciencegrid.org/>
8. ARC project. <http://www.nordugrid.org/arc/>
9. Smith, A., Tsaregorodtsev, A.: DIRAC: data production management. *J. Phys. Conf. Ser.* **119**, 062046 (2008)
10. Gfal2 Project. <https://dmc.web.cern.ch/projects-tags/gfal-2>
11. Tsaregorodtsev, A., Hamar, V.: MPI support in the DIRAC Pilot Job Workload Management System. *J. Phys. Conf. Ser.* **396**, 032109 (2012)
12. Stagni, F., Charpentier, P.: The LHCb DIRAC-based production and data management operations systems. *J. Phys.: Conf. Ser.* **368**, 012010 (2012)
13. WLCG Computing Grid Infrastructure. <http://wlcg.web.cern.ch>
14. 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)
15. Yan, T., Suo, B., et al.: Multi-VO support in IHEP's distributed computing environment. *J. Phys. Conf. Ser.* **664**, 062068 (2015)
16. Tsaregorodtsev, A.: DIRAC Distributed Computing Services. *J. Phys. Conf. Ser.* **513**, 03209 (2014)
17. Barkalov, K., Gergel, V.: Multilevel scheme of dimensionality reduction for parallel global search algorithms. In: OPT-i 2014. An International Conference on Engineering and Applied Sciences Optimization, Kos Island, Greece, 4–6 June 2014, pp. 2111–2124 (2014)
18. Gergel, V.P., Strongin, R.G.: Parallel computing for globally optimal decision making on cluster systems. *Future Gener. Computer Systems* **21**(5), 673–678 (2005)
19. Bastrakov, S., Meyerov, I., Gergel, V., et al.: High performance computing in biomedical applications. *Procedia Comput. Sci.* **18**, 10–19 (2013)