



Integration of the JINR Hybrid Computing Resources with the DIRAC Interware for Data Intensive Applications

Vladimir Korenkov^{1,3}, Igor Pelevanyuk^{1,3}(✉), and Andrei Tsaregorodtsev^{2,3}

¹ Joint Institute for Nuclear Research, Dubna, Russia
{korenkov, pelevanyuk}@jinr.ru

² CPPM, Aix-Marseille University, CNRS/IN2P3, Marseille, France
atsareg@in2p3.fr

³ Plekhanov Russian Economics University, Moscow, Russia

Abstract. Scientific data and computing-intensive applications become more and more widely used. Different computing solutions have different protocols and architectures, they should be chosen carefully in the design of computing projects of large scientific communities. In a modern world of diverse computing resources such as grids, clouds, and supercomputers the choice can be difficult. Therefore, software developed for integration of various computing and storage resources into a single infrastructure, the so-called interware, makes this choice easier. The DIRAC interware is one of these products. It proved to be an effective solution for many experiments in High Energy Physics and some other areas of science providing means for seamless access to distributed computing and storage resources. The DIRAC interware was deployed in the Joint Institute for Nuclear Research to serve the needs of different scientific groups by providing a single interface to a variety of computing resources: grid cluster, computing cloud, supercomputer Govorun, disk and tape storage systems. The DIRAC based solution was proposed for the currently operational Baryonic Matter at Nuclotron experiment as well as for the future experiment Multi-Purpose Detector at the Nuclotron-based Ion Collider fAcility. Both experiments have requirements making the use of heterogeneous computing resources necessary. A set of tests was introduced in order to demonstrate the performance of the JINR distributed computing system.

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

1 Introduction

Data intensive applications became now an essential means for getting insights of new scientific phenomena while analyzing huge data volumes collected by modern experimental setups. For example, the data recording to the tape system at CERN exceeded in total 10 Petabytes per month in 2018 for all the 4 LHC experiments. In 2021, with the start of the Run 3 phase of the LHC program, the experiments will resume data taking

with considerably increased rates. The needs for the computing and storage capacity of the LHCb experiment, for instance, will increase by an order of magnitude [1].

In a more distant future, with the start of the LHC Run 4 phase, the projected data storage needs of the experiments are estimated to exceed 10 Exabyte's. These are unprecedented volumes of data to be processed by distributed computing systems, which are being adapted now to cope with the new requirements.

Other scientific domains are quickly approaching the same collected data volumes: astronomy, brain research, genomics and proteomics, material science [3]. For example, the SKA large radio astronomy experiment [2] is planned to produce about 3 Petabytes of data daily when it will come into full operation in 2023.

The needs of the LHC experiments in data processing were satisfied by the infrastructure of the World LHC Computing Grid (WLCG). The infrastructure still delivers the major part of computing and storage capacity for these experiments. It is well suited for processing the LHC data ensuring massively parallel data treatment in a High Throughput Computing (HTC) paradigm. WLCG succeeded in putting together hundreds of computing centers of different sizes but with similar properties, typically providing clusters of commodity processors under control of one of the batch systems, e.g. LSF, Torque or HTCondor. However, new data analysis algorithms necessary for the upcoming data challenges require a new level of parallelism and new types of computing resources. These resources are provided, in particular, by supercomputers or High-Performance Computing (HPC) centers. The number of HPC centers is increasing and there is a clear need in setting up infrastructures allowing scientific communities to access multiple HPC centers in a uniform way as it is done in the grid systems.

Another trend in massive computing consists in provisioning resources via cloud interfaces. Both private and commercial clouds are available now to scientific communities. However, the diversity of interfaces and usage policies makes it difficult to use multiple clouds for applications of a particular community. Therefore, providing uniform access to resources of various cloud providers would increase flexibility and the total amount of available computing capacity for a given scientific collaboration.

Large scientific collaborations typically include multiple participating institutes and laboratories. Some of the participants have considerable computing and storage capacity that they can share with the rest of the collaboration. With the grid systems this can be achieved by installing complex software, the so-called grid middleware, and running standard services like Computing and Storage Elements. For managers of local computing resources who are usually not experts in the grid middleware, this represents a huge complication and often results in underused resources that would otherwise be beneficial for the large collaborations. Tools for easy incorporation of such resources can considerably increase efficiency of their usage.

The DIRAC Interware project provides a framework for building distributed computing systems using resources of all different types mentioned above and putting minimal requirements on the software and services that should be operated by the resources providers [4]. Developed originally for the LHCb experiment at LHC, CERN, the DIRAC Interware was generalized to suite requirements of a wide range of applications. It can be used to build independent distributed computing infrastructures as well as to provide services for existing projects. DIRAC is used by a number of High Energy Physics and

Astrophysics experiments but it is also providing services for several general-purpose grid infrastructures, for example, national grids in France [5] and Great Britain [6]. The EGI Workload Manager is the DIRAC service provided as part of the European Grid Infrastructure service catalog. It is one of the services of the European Open Science Cloud (EOSC) project inaugurated at the end of 2018 [7]. The EGI Workload Manager provides access to grid and cloud resources of the EGI infrastructure for more than 500 registered users.

This article is an extension of work originally presented in conference “Data Analytics and Management in Data Intensive Domains” (DAMDID/RCDL’2019) [4]. In this paper we give in Sect. 2 an overview of the DIRAC Interware, its capabilities and further development directions. In Sect. 3 computing facilities at the Joint Institute for Nuclear Research (JINR), Dubna are presented. We describe the DIRAC based infrastructure deployed at JINR putting together a number of local computing clusters as well as connecting cloud resources from JINR member institutions in Sect. 4. The results of storage and CPU performance evaluation of the integrated JINR computing infrastructure are presented in Sect. 5.

2 DIRAC Interware

The DIRAC Interware project provides a development framework and a large number of ready-to-use components to build distributed computing systems of arbitrary complexity. DIRAC services ensure integration of computing and storage resources of different types and provide all the necessary tools for managing user tasks and data in distributed environments [8]. Managing both workloads and data within the same framework increases the efficiency of data processing systems of large user communities while minimizing the effort for maintenance and operation of the complete infrastructure.

The DIRAC Workload Management System (WMS) architecture is based on the use of multiple autonomous agents which are deployed on computing resources using specific payload submission mechanisms in each case. This can be job submission to computing cluster batch systems, creation of appropriately contextualized virtual machines (VMs) in the cloud services or starting the agents as payloads on the BOINC volunteer computers. In all the cases, the agents inspect their running environment, evaluate the available computing capacity and present the description of the reserved resources to the DIRAC central Task Queue service. On the other hand, users are submitting their tasks to the central Task Queue where they are classified and prioritized according to the community policies. An efficient matching mechanism allows to quickly find the most suitable user task to each agent request taking into account the task requirements and priority. The agent-based WMS increases the efficiency of user payloads execution in a distributed system due to a preliminary validation of the execution environment reducing considerably execution failure rates. The agent-based WMS allows also to abstract away differences in access protocols of heterogeneous computing resources providers and aggregate them as a set of similar logical computing clusters, thus solving the problem of the resource’s heterogeneity [9].

Similarly to WMS, the DIRAC Data Management System (DMS) provides an abstract Storage Element interface together with its implementation for the most common modern storage access protocols (SRM, (GRID) FTP, HTTP, WebDAV, S3, etc).

Together with a central File Catalog service, which keeps traces of all the physical file replicas, the DIRAC DMS presents disparate distributed storage resources to the users as a single file system with a unique logical name space.

The DIRAC software is constantly evolving to follow changes in the technology and interfaces of available computing and storage resources. As a result, most existing HTC, HPC and cloud resources can be interconnected with the DIRAC Interware. In order to meet the needs of large scientific communities, the computing systems should fulfill several requirements. In particular, it should be easy to describe, execute and monitor complex workflows in a secure way respecting predefined policies of usage of common resources by multiple user communities.

2.1 Massive Operations

Usual workflows of large scientific collaborations consist in creation and execution of large numbers of similar computational and data management tasks. DIRAC is providing support for massive operations with its Transformation System. The system allows definition of Transformations – recipes to create certain operations triggered by the availability of data with required properties. Operations can be of any type: submission of jobs to computing resources, data replication or removal, etc. Each Transformation consumes some data and derives (“transforms”) new data, which, in turn, can form input for another Transformation. Therefore, Transformations can be chained creating data driven workflows of any complexity. Data production pipelines of large scientific communities based on DIRAC are using heavily the Transformation System defining many hundreds of different Transformations. Each large project develops its own system to manage large workflows each consisting of many Transformations. The DIRAC Production System is based on the experience of several community specific workflow management systems and provides a uniform way to create a set of Transformations interconnected via their input/output data filters. It helps production managers to monitor the execution of so created workflows, evaluate the overall progress of the workflow advancement and validate the results with an automated verification of all the elementary tasks.

2.2 Multi-community Services

Multiple relatively small scientific communities have access to common grid and cloud infrastructures but they cannot afford setting up and operating such complex services as DIRAC because of the lack of expertise and manpower. Therefore, services built with the DIRAC Interware can be provided by large infrastructure projects like national grids or JINR distributed cloud. DIRAC services can be configured to support multiple communities with comprehensive rules describing resources access rights for each user group. User communities can register in the DIRAC Configuration System additional resources, e.g. local university computing clusters, and make them available through the same interface as infrastructure resources. The user tasks monitoring and consumed resources accounting is provided per user and per community allowing to efficiently apply common quotas and policies of usage of the common infrastructure.

The security aspects are very important in distributed computing systems with multi-community access. In most of the currently existing multi-community grid infrastructures the security of all operations is based on the X509 PKI infrastructure. In this solution, each user has to, first, obtain a security certificate from one of recognized Certification Authorities (CA). The certificate should be then registered in a service holding a registry of all the users of a given Virtual Organization (VO). The user registry keeps the identity information together with associated rights of a given user. In order to access grid resources, users are generating proxy certificates which can be delegated to grid remote services in order to perform operations on the user's behalf.

The X509 standard based security is well supported in academia institutions but is not well suited for other researchers, for example, working in universities. On the other hand, there are well-established industry standards developed mostly for the web applications that allow identification of users as well as delegation of user rights to remote application servers. Therefore, grid projects started migration to the new security infrastructure based on the OAuth2/OIDC technology. With this technology, user's registration is done by local identity providers, for example, a university LDAP index. On the grid level a Single-Sign-On (SSO) solution is provided by a federation of multiple identity providers to ensure mutual recognition of user security tokens. In particular, the EGI infrastructure has come up with the Check-In SSO service as a federated user identity provider.

The DIRAC user management subsystem was recently updated in order to support this technology. Users can be identified and registered in DIRAC based on their SSO tokens containing also additional user metadata, which, in turn, defines user rights within the DIRAC framework. The DIRAC implementation of the new security framework is generic and can be easily configured to work with multiple identity providers and SSO systems.

2.3 DIRAC Software Evolution

The intensity of usage of the DIRAC services is increasing and the software must evolve to cope with the new requirements. This process is mostly driven by the needs of the LHCb experiment, which remains the main consumer and developer of the DIRAC software. As was mentioned above, the order of magnitude increase in the data acquisition rate of LHCb in 2021 dictates a revision of the technologies used in its data processing solutions.

Several new technologies were introduced recently into the DIRAC software stack. The use of Message Queue (MQ) services allows passing messages between distributed DIRAC components in an asynchronous way with the possibility of message buffering in case of system congestion.

The DIRAC service's states are kept in relational databases using MySQL servers. The MySQL databases have shown very stable operation over the years of usage. However, the increased amount of data to be stored in databases limits the efficiency of queries and new solutions are necessary. The so-called NoSQL databases have excellent scalability properties and can help in increasing the efficiency of the DIRAC components. The ElasticSearch NoSQL (ES) database solution was applied in several DIRAC subsystems relying on heavy queries on loosely structured data, e.g. activities monitoring or service logs inspection.

DIRAC from the moment of its inception used a custom client/service protocol to maximize the efficiency of queries. However, the current implementations of the secure HTTP protocol became very efficient with supporting software packages available, which follow all the evolutions of the protocol versions and security standards. Therefore, the next generation of the DIRAC client/server protocol will be based on the HTTP standard. This can help DIRAC acceptance by new communities with strong security requirements.

These and other additions and improvements in the DIRAC software aim at the overall increase of the system efficiency and scalability to meet requirements of multiple scientific communities relying on DIRAC services for their computing projects.

3 JINR Computing Resources

The Joint Institute for Nuclear Research is an international intergovernmental organization, a world-famous scientific center that is a unique example of the integration of fundamental theoretical and experimental research. It consists of seven laboratories: Laboratory of High Energy Physics, Laboratory of Nuclear Problems, Laboratory of Theoretical Physics, Laboratory of Neutron Physics, Laboratory of Nuclear Reactions, Laboratory of Information Technologies, Laboratory of Radiation Biology. Each laboratory is comparable with a large institute in the scale and scope of the research activity.

JINR has a powerful highly productive computing environment that is integrated into the world computer network through high-speed communication channels. The basis of the computer infrastructure of the Institute is the Multifunctional Information Computer Complex (MICC). It consists of several large components: grid cluster, computing cloud, supercomputer Govorun. Each component has its features, advantages, and disadvantages. Different access procedures, different configuration and connection with different storage systems make it difficult to use all the facilities together for one set of tasks.

3.1 Computing Resources

Grid Cluster. The JINR grid infrastructure is represented by a Tier1 center for the CMS experiment at the LHC and a Tier2 center.

After the recent upgrade, the data processing system at the JINR CMS Tier1 consists of 415 64-bit nodes: 2 x CPU, 6–16 cores/CPU that form altogether 9200 cores for batch processing [10]. The Torque 4.2.10/Maui 3.3.2 software (custom build) is used as a resource manager and a task scheduler. The computing resources of the Tier2 center consist of 4,128 cores. The Tier2 center at JINR provides data processing for all the four experiments at the LHC (Alice, ATLAS, CMS, LHCb) and in addition supports many virtual organizations (VO) that are not members of the LHC (BES, BIOMED, COMPASS, MPD, NOvA, STAR, ILC).

Grid cluster is an example of a n HTC type facility. It means that the primary task of this cluster is to run thousands of independent processes at the same time. Independent means that once a process has started and until it finishes, the process does not rely on any input that is being produced at the same moment by other processes.

Jobs may be sent to the grid using a CREAM Computing Element – a service installed in JINR specially for the grid jobs. Computing element works as an interface to the local batch farm. Its primary task is to authenticate the owner of the job and redirect it to the right queue. Users are required to have an X509 certificate and be a member of a Virtual Organization supported by the Computing Element.

Cloud Infrastructure. The JINR Cloud [11] is based on an open-source platform for managing heterogeneous distributed data center infrastructures - OpenNebula 5.4. The JINR cloud resources were increased up to 1564 CPU cores and 8.1 TB of RAM in total. Cloud infrastructure is used primarily for two purposes: to create personal virtual machines and to create virtual machines to serve as worker nodes for jobs. We are going to focus on the second case.

The biggest advantage of cloud resources as computing capacity is their flexibility. In the case of grid or batch resources, several jobs working on one worker node share between them: operating system, CPU cores, RAM, HDD/SSD storage, disk Input/Output capabilities, and network bandwidth. If a job needs more disk space or RAM it is not straightforward to submit the job to the grid without the help of administrators, who in most cases have to create a dedicated queue for this particular kind of job. In the case of clouds, it is much easier to provide a specific resource suiting the job requirements. It may be a virtual machine with a large disk, specific operating system, required number of CPU cores, RAM capacity and network.

When a job destined to the cloud enters the system the corresponding virtual machine is created by DIRAC using the OpenNebula API. During the contextualization process, the DIRAC Pilot is installed in the VM and configured to receive jobs for this cloud resource. Once the job is finished, the pilot attempts to get the next job. If there are no more jobs for the cloud, the pilot will request the VM shutdown. The pilot in the cloud environment is not limited by the time and may work for weeks. These features make cloud resources perfectly suitable for specific tasks with unusual requirements.

Govorun Supercomputer. The Supercomputer Govorun was put into production in March 2018 [12]. It is a heterogeneous platform exploiting several processors' technologies: a GPU part and a CPU part. The GPU part unites 5 servers DGX-1. Each server hosts 8 NVIDIA Tesla V100 processors. The CPU part is a highly dense liquid-cooled system. The processors inside are Intel® Xeon® Platinum 8268 (80 servers). The total performance of all the three parts is 1 PFlops for operations with single precision and 0.5 PFlops for double precision. SLURM 14.11.6 is used as the local workload manager.

The supercomputer is used for tasks, which require massive parallel computations. For example: to solve problems of lattice quantum chromodynamics for studying the properties of hadronic matter with high energy density and baryon charge and in the presence of strong electromagnetic fields, mathematical modeling of the antiproton-proton and antiproton-nucleus collisions with the use of different generators. It is also used for simulation of the collision dynamics of relativistic heavy ions for the future MPD experiment at the NICA collider.

Right now, the supercomputer utilizes its own authentication and authorization system. Every user of the supercomputer should be registered and allowed to send jobs.

Sometimes, a part of the supercomputer is free from parallel tasks and may be used as a standard batch system. Special user was created for the DIRAC service at JINR. All jobs sent to the Govorun are executed with this user identity. This frees actual users from additional registration procedures delegating user profile management and resource consumption accounting to DIRAC.

3.2 Storage Resources

EOS Storage on Disks. EOS [13] is a multi-protocol disk-only storage system developed at CERN since 2010 for the High Energy Physics experiments (including the LHC experiments). Having a highly-scalable hierarchical namespace, and with the data access possible via the xRootD protocol, it was initially used for the physics data storage. Today, EOS provides storage for both physics experiments and user use cases. For the user authentication, EOS supports Kerberos (for the local access) and X.509 certificates for the grid access. To ease the experiment's workflow integration, SRM as well as GridFTP access is provided. EOS supports the xRootD third-party copy mechanism from/to other xRootD enabled storage services.

The EOS was successfully integrated into the MICC structure. The NICA experiments already use EOS for data storage. At the moment there are ~200 TB of "raw" BM@N data and ~84 GB of simulated MPD data stored in the EOS instance. EOS is visible as a local file system on the MICC worker nodes. It allows users authorized by the Kerberos5 protocol to read and write data. A dedicated service was installed to allow usage of X509 certificates with VOMS extensions.

dCache Disk and Tape Storage. The core part of the dCache has been proven to efficiently combine heterogeneous disk storage systems of the order of several hundred TBs and present its data repository as a single filesystem tree. It takes care of the data integrity, hardware failures and provides that a minimal configured number of copies of each dataset resides within the system to ensure high data availability in the case of disk server maintenance or failure. Furthermore, dCache supports a large set of standard access protocols to the data repository and its namespace. It supports DCAP, SRM, GridFTP, and xRootD protocols [14].

dCache at JINR consists of two parts: disk storage and tape storage. The disk part operations are similar to EOS. The tape storage is accessed through dedicated disk buffer servers. When data is uploaded to the dCache tape part, it is first uploaded to the disk buffer. If the disk buffer is occupied above a certain threshold (which is 80% in our case), all the data is moved from disk to tape and removed from the disk buffer. While the data stays in the buffer, access to it is similar to the access to the dCache disk data. But once the data is moved to the tape and removed from the disk, it will be accessed with a certain delay. The time required to select the right tape and transfer data from the tape to the disk depends on the tape library task queue. Generally, the time varies from 20 s to up to several minutes.

Tape library should be used only for archive storage and preferably for big files. Otherwise, it may incur an unnecessary load on the tape system.

Ceph Storage. Software-defined storage (SDS) based on the Ceph technology is one of the key components of the JINR cloud infrastructure. It has been running in production mode since the end of 2017. It delivers object, block and file storage in one unified system. Currently, the total amount of raw disk space in the SDS is about 1 PB. Due to triple replication, effective disk space available for users is about 330 TB. Users of Ceph can attach part of the storage to a computer using the FUSE disk mounting mechanism. After that, it is possible to read and write data to the remote storage as if it is connected directly to the computer.

The Ceph storage was integrated into the DIRAC installation for tests. Since Ceph does not allow authentication by X509 certificates with VOMS extensions, a dedicated virtual machine was configured to host a DIRAC Storage Element – a standard service which works as a proxy to a file system. It checks certificates with VOMS extensions before allowing writing and reading to a dedicated directory. Right now, Ceph storage does not allow massive transfers since it relies on one server with Ceph attached by FUSE. The test demonstrated that the maximum speed of transfers is not exceeding 100 MB/s which is a consequence of 1 Gb network connection. The way to increase the performance of this storage is an improvement of the network speed up to 10 Gb/s and a possible creation of additional DIRAC Storage Elements which can share the load between them.

4 JINR DIRAC Installation

The DIRAC installation in JINR consists of 4 virtual machines. Three of them are placed on a dedicated server to avoid network and disk I/O interference with other virtual machines. The operating system on these virtual machines is CentOS 7. It appeared that some of the LCG software related to grid job submission is not compatible with CentOS. To cope with that, we created a new virtual machine with Scientific Linux 6 installed there. Flexibility of the DIRAC modular architecture allowed us to do that easily. The characteristics of the virtual machines hosting DIRAC services are presented in Table 1.

Table 1. Virtual machines hosting DIRAC services

	dirac-services	dirac-conf	dirac-web	dirac-sl6
OS	CentOS	CentOS	CentOS	Scientific Linux
Version	7.5	7.5	7.5	6.10
Cores	8	4	4	2
RAM	16 GB	8 GB	8 GB	2 GB

4.1 Use Cases in JINR

Up to now, we foresee two big possible use cases: Monte-Carlo generation for Multi-Purpose Detector (MPD) at NICA and data reconstruction for Baryonic Matter at Nuclotron(BM@N).

Raw data is received by the BM@N detector and uploaded to the EOS storage. There are two data taking runs available now: run 6 and run 7. The data sizes are respectively: 16 TB and 196 TB. All the data is split in files, 800 files for run 6 and around 2200 files for run 7. The main difficulty with these files is the fact that their sizes are very different: from several MBs to up to 800 GBs per one file. This makes data processing a tough task especially on resources with a small amount of local storage or a bad network connection. The data could be processed using the Govorun Supercomputer, but the EOS storage may be accessed only over the xRootD protocol. The data may require full reprocessing one day, if the reconstruction algorithms are changed.

So far, the best solution would be to process big files in the cloud, other files in the grid infrastructure and sometimes, when the supercomputer has free job slots, do some processing there. But without some central Workload Management system and Data Management system this is a difficult task. The data could be placed not only in EOS but also in dCache. This would allow data delivery to the worker nodes using grid protocols like SRM or xRootD. Once the X509 certificates start working for the EOS storage, it will also be included in the infrastructure and be accessible from everywhere.

The second use case is the Monte-Carlo generation for the MPD experiment. The Monte-Carlo generation could be performed almost on all the components of MICC at JINR. It is a CPU intensive task less demanding in terms of disk space and input/output rates. The file size could be tuned to be in a particular range for the convenience of the future storage and use. The use of a central distributed computing system may not be critical right now, but it will definitely be useful later, when the real data arrives. It would allow for design and testing of production workflows, and provide access to different organizations to participate in the analysis of the experiment data.

5 JINR Resource Performance Evaluation

While using hybrid computing resources it is not easy to predict the exact outcome of the job's execution. Too many variables are involved: network speed, disk I/O, different operating systems, different processors. Therefore, it is impossible to plan the load and data flows without additional characterization of the resources.

There are two ways to obtain this information. The first option is to collect details about all the components and estimate the approximate performance. It is a lot of work to collect and analyze this data as well as to keep this information up to date. Any major change or upgrade on one of the resources will make this information obsolete and would require repeated analysis. The results will have to be confirmed by real jobs sent to the system. The advantage of this approach is that it does not affect the resources and their performance itself. It may work on small infrastructures, but on big ones it would require too much administrative work related to data collection and analysis.

The second approach is to prepare a set of functional tests and run it from time to time to collect all important metrics and compare them with each other. This approach

may be dangerous since these tests may affect other jobs running on the same shared resources. The tests can interrupt execution of other jobs or data transfers. So, certain care should be taken to run those tests. The advantage of this approach is that it provides real metrics for the resource's performance.

For the purpose of performance evaluation of the JINR resources, the second approach has been chosen since it is accurate, consistent, easy to reproduce. It is possible to coordinate with administrators of local resources and reduce the chance of interruption of any work. Two main measurements were chosen as the most important ones: data download from Storage Elements and CPU performance.

5.1 Performance Test of Storage Resources

In the case of massive data processing, it is crucial to know the limitations of different components. The limitations may depend on the computing and storage resource usage patterns. In most cases, jobs need to download some amount of data before execution of the user application payload. In order to characterize Storage Elements, a synthetic test was proposed: run many jobs on one computing resource, make them start downloading all the same data at the same moment, measure how much time it takes to get the file.

Every test job had to go through the following steps:

1. Start execution on the worker node.
2. Check out the transfer start time.
3. Wait until the transfer starting time.
4. Start the transfer.
5. When the transfer is finished, report the duration of the operation.
6. Remove the downloaded file.

Two storage systems were chosen for the tests: EOS and dCache since only those systems were accessible for both reading and writing on all the computing resources at the moment of the tests. The test file size was chosen to be 3 GB as the most typical one. The amount of the test jobs in one test campaign depended on the number of free CPU cores in our infrastructure. We submitted 200 jobs during one test campaign. Not all of them could start at the same time, which means that during the test less than 200 jobs may download data. This is taken into account when we calculate the total transfer speed.

Several test campaigns were performed to evaluate the variance of the test, but all of them showed similar results. In Fig. 1 two representative examples are presented. To calculate the transfer speed, the following formula was used:

$$\text{Transfer speed} = \frac{\text{Total data transferred}}{\text{Longest transfer duration}}$$

This formula allows calculation of the worst transfer speed of one file during the test campaign. For EOS the calculated transfer speed was 990 GB/s with 200 jobs and for dCache it was estimated to be 1390 GB/s with 176 jobs. It should be mentioned that all the tests were performed on a working infrastructure, so some minor interference may

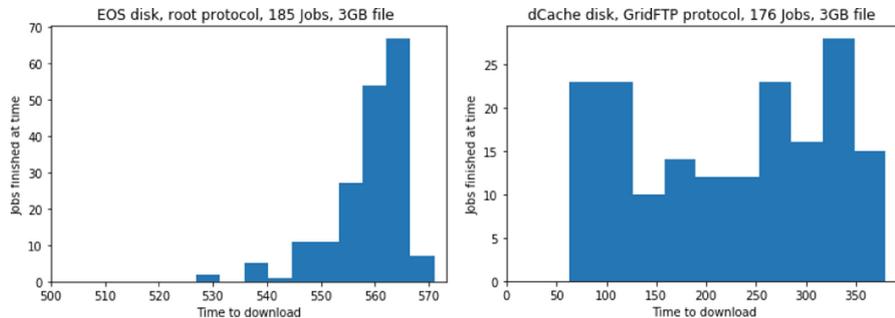


Fig. 1. Data download times in running jobs

be caused by other activities. On the other hand, the demonstrated plots represent real transfers performed under normal conditions.

The measurement of the download speed provides information to estimate how much data can be simultaneously downloaded to the computing element. But it does not give evidence of the most important reasons for the transfer speed limitation. It may be network performance issues, SE architectural performance limitation, bad performance of a hardware, wrong configuration, or heavy use of the storage resources by other jobs. Nevertheless, this test is a good starting point for understanding which components should be optimized to increase the overall performance. It also provides useful numbers for the users who need to download data during their job's execution.

5.2 Performance Test of Computing Resources

The data downloaded to a worker node is processed by the user application. The rate of the data processing impacts the rate of data download. On the other hand, the rate of processing depends on the performance of the CPU. Measurements of computing resources performance are usually based on some synthetic tests chosen to represent most closely typical computational tasks. It is impossible to have a single fully representative benchmark test. The results depend a lot on the properties of the test load and the performance measurements correspond only approximately to the performance with real jobs.

For our purposes, the best approach is to use one of the existing benchmarks. HEP-SPEC2006 is a standard benchmark in computing for high energy physics [15]. But it is mostly used by local administrators of computing resources since it is based on non-free benchmark SPEC2006. The test is rather heavy and takes several hours to run. For our use case, the DIRAC Benchmark 2012 (DB12) turned out to be a good alternative to HEP-SPEC2006 [16]. This benchmark was originally created for prediction of the duration of LHCb Monte-Carlo tasks. It is fast (takes around 60 s), and it runs every time the DIRAC pilot job agent starts execution.

The fact that DB12 benchmark is executed in each pilot job agent allows us to use the already running pilots for the estimation of the resource's performance. We used the test approach described in the previous section: we submit many jobs on one particular computing resource and start all benchmarks simultaneously. This is done to be sure that

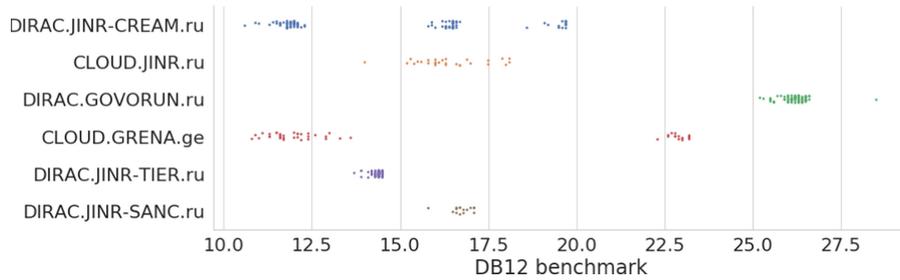


Fig. 2. Results of DB12 benchmark on different resources.

the benchmark on a particular CPU core is performed when other CPU cores are also busy. The benchmark tests in regular pilot jobs cannot guarantee that.

The results are presented in Fig. 2. The tests were performed on several resources. The biggest ones are: Tier1 grid site (DIRAC.TIER.ru), Tier2 grid site (DIRAC.JINR-CREAM.ru), the Govorun Supercomputer (DIRAC.GOVORUN.ru), and the JINR Cloud sites (CLOUD.JINR.ru and others). Other resources are also presented on the plot just for the reference. Every dot is a particular test job result. Dots related to a particular resource are placed randomly within some margin in order to demonstrate the amount of different results with the same value and avoid overlapping. The more to the right the point is - the more performing the resource that it represents.

If one resource has twice better results in DB12 benchmark than another one, it means that the computing task should run on the first resource twice faster. Or, in other words, the first resource should provide twice more results per time period than the second one. The benchmark values represent the performance not of the whole CPU but only one core of it.

The results of this test should not be used to estimate the “power” of a computing resource as a whole because the benchmarks examine only one core of a processor and tell nothing about the number of cores on the resource. To get this information one should take the total amount of job slots provided by the resource and multiply it by the average benchmark result.

To validate the results of DB12 benchmark in our infrastructure we decided to analyze the information about user jobs running under the control of the DIRAC WMS. Every job has information about the performance of the core it is running on. This information

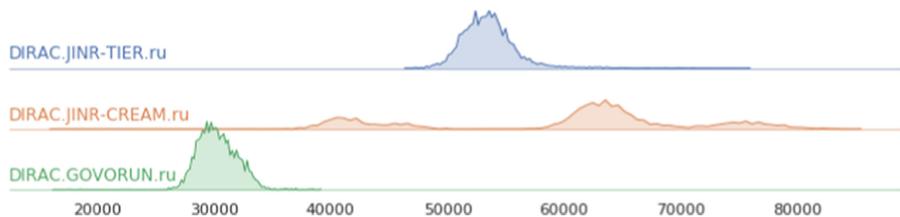


Fig. 3. Distribution of users’ jobs duration in seconds on different resources.

is stored in the DIRAC database. We retrieved this information for jobs with similar user payload properties.

In Fig. 3 the duration of the user jobs is presented. The data here corresponds well to the results of dedicated tests presented in Fig. 2. In this test, only Tier1, Tier2 sites and the Govoron Supercomputer were used. Discrepancies here may be due to several reasons: input data transfer delays; output data transfer delays; random variations in the user application execution time (Monte-Carlo simulation).

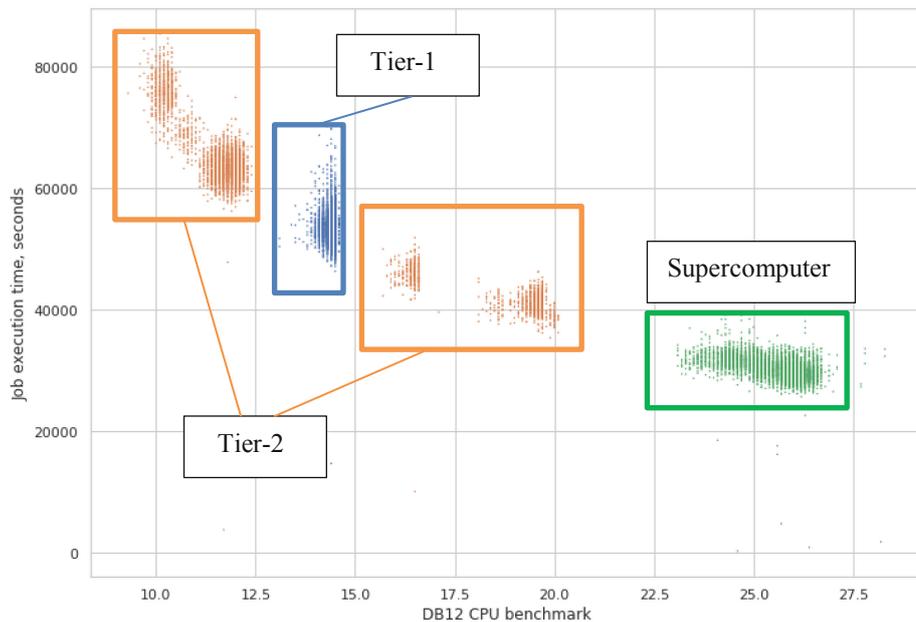


Fig. 4. Execution times of user jobs on CPU cores with different benchmark results.

The same results are presented in Fig. 4 to demonstrate the correlation between the CPU performance benchmark and the real jobs execution time. Each dot in the plot corresponds to a single job. As one can see, all the dots are fitting a hyperbola shape area which shows a strong linear correlation of the job execution speed and the DB12 benchmark of the corresponding CPU. There are distinct clusters of the dots in the plot corresponding to computing clusters with different CPU models having different performance.

The results of the tests show the accuracy of user jobs execution time prediction on a given CPU using the DB12 benchmark provided that the user payload is first characterized by running on some reference processor with a known performance benchmark. The accuracy is estimated to be around 15%. The benchmark-based job execution time estimation can be used by users for monitoring their jobs. In some cases, users can choose resources with higher performance if the job turnaround time is important. The benchmark information can be also used by the DIRAC WMS job scheduler to place

jobs with high CPU requirements or higher priority jobs to a more performant computing resource.

6 Conclusion

The Joint Institute for Nuclear Research is a large scientific organization with several big computing and storage subsystems. Most of the time they are used separately by scientific groups and there is no simple way to aggregate them to allow workflows spanning the whole computing center. Different components could be separated by a slower network, different authentication systems and different protocols. This problem becomes more visible when one of the resources is overloaded while others are underused. If a good interoperability between the components could be ensured, it would be possible to easily switch between them in order to balance their loads. There are many tasks that are not bound to use some particular type of resources neither because of technical compatibility nor because of the adopted usage policies.

With the interware technology, it turns out to be easy to integrate computing resources and access them as a single meta-computer. This leads to significant improvements in efficiency of usage of the JINR computing infrastructure. Multiple JINR scientific groups can benefit from the uniform interfaces to various computing facilities. Therefore, integration services based on the software provided by the DIRAC Interware project were set up and evaluated.

The DIRAC services were installed at JINR in order to integrate resources used by big experiments like MPD and BM@N. The initial tests and measurements demonstrated the possibility to use it for data reconstruction and Monte-Carlo generation on all the JINR resources: Grid cluster, Computing Cloud and Govorun supercomputer.

While integrating heterogeneous computing resources, it is still important to keep track of the properties and performance metrics of each component in the integrated system. Therefore, a set of tests to evaluate the performance of different clusters were introduced and validated in a production environment with real users' tasks. The resulting metrics and benchmarks led to a better understanding of the performance of the integrated system as a whole and provided the necessary information for users and for the task meta-scheduling mechanism to optimize the task placement in different computing clusters depending on the task resources requirements and priorities.

References

1. Bozzi, C., Roiser, S.: The LHCb software and computing upgrade for Run 3: opportunities and challenges. *J. Phys.: Conf. Ser.* **898**, 112002 (2017). <https://doi.org/10.1088/1742-6596/898/10/112002>
2. SKA Telescope. <https://www.skatelescope.org/software-and-computing/>. Accessed 19 Aug 2019
3. Kalinichenko, L., et al.: Data access challenges for data intensive research in Russia. *Inform. App.* **10**(1), 2–22 (2016). <https://doi.org/10.14357/19922264160101>
4. Korenkov, V., Pelevanyuk, I., Tsaregorodtsev, A.: DIRAC system as a mediator between hybrid resources and data intensive domains. In: *Selected Papers of the XXI International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2019)*, vol. 2523, pp. 73–84, Kazan, Russia (2019)

5. France Grilles. <http://www.france-grilles.fr>. Accessed 01 Feb 2020
6. Britton, D., et al.: GridPP: the UK grid for particle physics. *Philos. Trans. R. Soc. A* **367**, 2447–2457 (2009)
7. European Open Science Cloud. <https://www.eosc-portal.eu>. Accessed 19 Aug 2019
8. Tsaregorodtsev, A.: DIRAC distributed computing services. *J. Phys: Conf. Ser.* **513**(3), 032096 (2014). <https://doi.org/10.1088/1742-6596/513/3/032096>
9. Gergel, V., Korenkov, V., Pelevanyuk, I., Sapunov, M., Tsaregorodtsev, A., Zrellov, P.: Hybrid distributed computing service based on the dirac interware. *Commun. Comput. Inf. Sci.* **706**, 105–118 (2017). https://doi.org/10.1007/978-3-319-57135-5_8
10. Baginyan, A., et al.: The CMS Tier1 at JINR: five years of operations. In: *Proceedings of VIII International Conference on Distributed Computing and Grid-technologies in Science and Education*, vol. 2267, pp. 1–10 (2018)
11. Baranov, A., et al.: New features of the JINR cloud. In: *Proceedings of VIII International Conference on Distributed Computing and Grid-technologies in Science and Education*, vol. 2267, pp. 257–261 (2018)
12. Adam, Gh., et al.: IT-ecosystem of the HybriLIT heterogeneous platform for high-performance computing and training of IT-specialists. In: *Proceedings of VIII International Conference on Distributed Computing and Grid-technologies in Science and Education*, vol. 2267, pp. 638–644 (2018)
13. Peters, A.J., et al.: EOS as the present and future solution for data storage at CERN. *J. Phys.: Conf. Ser.* **664**, 042042 (2015). <https://doi.org/10.1088/1742-6596/664/4/042042>
14. dCache, the Overview. <https://www.dcache.org/manuals/dcache-whitepaper-light.pdf>. Accessed 19 Aug 2019
15. Michelotto, M., et al.: A comparison of HEP code with SPEC benchmarks on multi-core worker nodes. *J. Phys.: Conf. Ser.* **219**, 052009 (2010). <https://doi.org/10.1088/1742-6596/219/5/052009>
16. Charpentier, P.: Benchmarking worker nodes using LHCb productions and comparing with HEPspec06. *J. Phys.: Conf. Ser.* **898**, 082011 (2017). <https://doi.org/10.1088/1742-6596/898/8/082011> (IOP Conf. Series)