Approaches, services, and monitoring in a distributed heterogeneous computing environment for the MPD experiment*

A.A. Moshkin, I.S. Pelevanyuk, D.V. Podgainy, O.V. Rogachevsky,
O.I. Streltsova, M.I. Zuev

Joint Institute for Nuclear Research

The article discusses some issues of creating a heterogeneous computing environment for processing and storing data from the MPD experiment of the NICA megaproject, which is being implemented at the Joint Institute for Nuclear Research. The creation of such an environment requires combining many different concepts and techniques based on the paradigm of distributed computing and a hierarchical data processing and storage system. The created environment must meet the requirements of high performance, reliability and availability for various groups of users, which is achieved, among other things, through an advanced monitoring system. The DIRAC Interware platform, which supports computing at all levels: workload management, data management, workflow management, user management, configuration management, etc., was chosen as the basis for creating such an environment. Owing to DIRAC, computational resources and the hierarchical hyperconverged data processing and storage system of the "Govorun" supercomputer were implemented in the heterogeneous computing environment. Moreover, the "Govorun" supercomputer not only plays a key role in the created environment, but also, due to its flexibility, provides an opportunity to test in practice both the latest architectural solutions and software solutions in the field of computing and data processing.

Keywords: data processing and storage, middleware for distributed computing, user job monitoring, high-performance computing, computing for megascience projects.

1. Introduction

One of the main challenges in scientific computing is the processing of large amounts of data for experiments in high-energy physics, which can be classified as one of the most demanding scientific tasks in terms of computational resources. The amount of data obtained in modern experiments reaches several tens of petabytes per year. The data processing procedure includes a number of operations, such as the reconstruction of events, simulation of the installation, and data analysis, each of which is a rather complex task and requires significant computational resources. For example, in the ATLAS experiment at the Large Hadron Collider (LHC), an average of 600 events are written per second, which results in a constant data stream with a transfer rate of 300 MB/s, demanding further processing. The reconstruction time for one event is 10 s, therefore, only for the reconstruction of events at the same rate, the simultaneous operation of 6,000 processors is required [1]. Even more resources are needed for experimental installation simulation tasks, which are an integral part of data analysis. As a result, the computing resource demands for LHC data processing went far beyond the capabilities of a single, even large, data center and led to the creation of a distributed data processing system. At the same time, the time required to process data and quickly obtain physical results directly depends on the performance of the computing environment as a whole. With this in mind, computing is currently being created for the NICA (Nuclotron-based Ion Collider Facility) megaproject, which is being implemented at the Joint Institute for Nuclear Research (JINR) [2]. NICA is a new accelerator complex designed to study the properties of dense baryonic matter. After putting the NICA collider into operation, JINR scientists will be able to create in the Laboratory a special state of matter in which our Universe stayed shortly after the Big Bang, i.e., Quark-Gluon Plasma (QGP). The first experiment at the NICA complex will be MPD (Multi-Purpose Detector) [3].

* The studies in this direction were supported by the RFBR special grant ("Megascience – NICA"), No.18-02-40101.

The MPD apparatus is designed as a 4π spectrometer capable of detecting charged hadrons, electrons and photons in heavy-ion collisions at high luminosity in the energy range of the NICA collider. To reach this goal, the detector will comprise a precise 3D tracking system and a high-performance particle identification (PID) system based on time-of-flight measurements and calorimetry.

The basic design parameters are determined by physical processes in nuclear collisions at NICA and by several technical constraints guided by a trade-off of effective tracking and PID against a reasonable material budget. At the design luminosity, the event rate in the MPD interaction region is about 7 kHz; the total charged particle multiplicity exceeds 1,000 in the most central Au+Au collisions at $\sqrt{s_{NN}} = 11$ GeV.

The main program blocks of the experiment are the following systems:

- Data acquisition (DAQ, online);
- Multilevel data selection (trigger, online);
- Data processing (particle tracks recognition and reconstruction, online+offline);
- Physical analysis (offline).

The expected data flow is up to 50 GB per second. Two sessions of the accelerator operation per year, each lasting up to 4 months, are planned. The total volume of incoming data per year can reach 1 exabyte. To reduce the size of the required storage, it is necessary to carry out as many stages of data processing in real time as possible, i.e., to implement in real time (during data acquisition) not only a trigger, but also the maximum possible number of stages of the recognition and reconstruction of elementary particle tracks.

For efficient data processing of the MPD experiment, a heterogeneous, geographically distributed computing environment is currently being created on the basis of the DIRAC Interware [4]. The DIRAC Interware is an open-source development platform for the integration of heterogeneous computing and storage resources. It was originally developed for the LHCb computing infrastructure, but later DIRAC was released as a general-purpose solution for scientific groups. Right now, it is used by many experiments in high-energy physics, particle physics and astronomy: LHCb, Belle-II, BES-III, CTA, CLIC, ILC. The purpose of DIRAC is to ensure access to various computing and storage resources through standard interfaces, i.e., web, command line, API, and REST. Another purpose is to provide a set of standard tools for workload management, data management, accounting, workflow management, user management, etc. An important point for the stability of such an environment is the advanced monitoring of tasks performed on various computing sites and data flows between the sites.

At the moment, the heterogeneous, geographically distributed computing environment includes the Tier1 and Tier2 grid sites, the JINR MICC cloud component, the computing clusters of VBLHEP (Veksler and Baldin Laboratory of High Energy Physics JINR) and UNAM Mexico, however, the key element of the environment being created is the "Govorun" supercomputer [5, 6], which allows one to carry out massive calculations, as well as, due to the flexibility of its architecture, IT research in order to create an effective computing model for the NICA megaproject [7].

This article briefly describes an approach to creating a geographically distributed computing environment that currently allows automating the process of modeling physical events for the future MPD experiment, and after the launch of the NICA accelerator complex, it will be used to process real experimental data. The article also presents the main properties of DIRAC Interware as a basic software element for creating such an environment and describes the data processing model on the "Govorun" supercomputer as one of the key elements of this environment.

2. Data processing model on the "Govorun" supercomputer

The main HPC resource of JINR is the "Govorun" supercomputer, which comprises two components, GPU and CPU [5, 6]. The GPU component is based on 5 NVIDIA DGX-1 servers with 8 NVIDIA Tesla V100 GPUs. The CPU component contains 21 RSC Tornado nodes based on Intel® Xeon Phi™ and 88 RSC Tornado nodes based on Intel® Xeon® Platinum 8268 and has two Intel® SSDs DC P4511 (NVMe, M.2), 2TB each. Additionally, the "Govorun" supercomputer has 4 special nodes with 12 high-speed, low-latency solid state drives Intel® Optane™ SSD DC P4801X 375GB M.2 series with Intel® Memory Drive Technology (IMDT), which allows getting 4.2 TB for very hot data per node.

The internal network of the “Govorun” supercomputer consists of three main parts: a communication and transport network, a control and monitoring network and a task control network. The communication and transport network uses Intel OmniPath 100 Gbps technology. The network is built on a "thick tree" topology based on 48-port Intel OmniPath Edge 100 Series switches with full liquid cooling. The control and monitoring network enables the unification of all compute nodes and the control node into a single Fast Ethernet network. This network is built using Fast Ethernet switches HP 2530-48. The job control network connects all compute nodes and the control node into a single Gigabit Ethernet network. The job control network is built using HPE Aruba 2530 48G switches.

The total peak performance of the “Govorun” supercomputer is 780 TFlops for double precision (12th and 21st places in the current TOP50 list http://top50.supercomputers.ru/list). The supercomputer also has local Lustre storage with a capacity of 288 TB. The “Govorun” supercomputer took 31st place in the IO500 list (https://io500.org/) with a bandwidth of more than 35 GiB/s and a metadata performance of more than 230 kIOP/s.

It should be noted here that the CPU component of the “Govorun” supercomputer is a hyperconverged software-defined system and ranks 16th in “10 node challenge” in the current edition of the IO500 list (https://io500.org/list/isc21/ten) under the DAOS (Distributed Asynchronous Object Storage) operation. The property of hyperconvergence makes the “Govorun” system very flexible for customizing the user’s job, ensuring the most efficient use of the computing resources of the supercomputer.

It is noteworthy that modern HPC systems are used not only as traditional computing environments for performing massively parallel calculations, but also as systems for Big Data analysis and artificial intelligence tasks that arise in different scientific and applied problems. At the same time, despite the increase in supercomputer performance, memory and data storage bandwidths are becoming bottlenecks. To accelerate work with data for tasks of different types, solved on the “Govorun” supercomputer, a hierarchical hyperconverged data processing and storage system with a software-defined architecture was developed and implemented. This system is based on the disaggregated RSC [8] solution for data processing and storage, which enhances the efficiency of solving user problems, as well as increases the efficiency of using both computational resources and data storage resources. It realizes a new paradigm for working with data, i.e., the integration of computing elements and novel types of data storage elements (Intel Optane PMem, Intel SSD) into a unified computing environment.

According to the speed of accessing data, the system is divided into levels available for the user’s choice, namely, a hot layer implemented on the basis of Intel Optane PMem, a warm layer based on Intel SSD NVMe under the management of the Lustre file system, a warm layer implemented as “an on-demand storage system”, which can be managed by different file systems defined by the user, a cold layer implemented on HDD of sufficient volume, which ensures data storage, but does not meet the peak requirements of a computational task. Each layer of the data storage system under development can be used both independently and as part of data processing workflows. It should be pointed out that a part of the cold storage is managed by the geographically distributed EOS file system (https://eos-web.web.cern.ch/eos-web/), which allows connecting the data processing and storage system implemented on the “Govorun” supercomputer to geographically distributed storages, the so-called Data-Lakes. The hierarchical hyperconverged data processing and storage system with a software-defined architecture represents an extremely convenient tool for processing large amounts of data, which can cardinally speed up work with data and is already actively used for the experiments of the NICA mega-project (Figure 1).

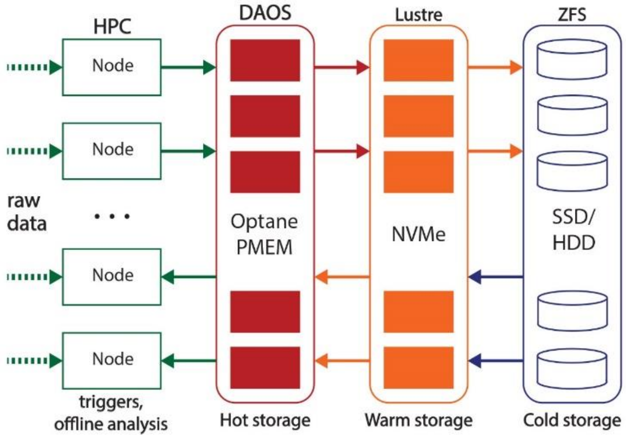

Figure 1. Multilevel system for processing and storing data on the “Govorun” supercomputer

It can be illustrated by the experience of data mass production tasks for the NICA MPD experiment. First of all, the use of Intel SSD allowed us to organize two fast and stable on-demand storages under the Lustre and NFS file systems on the basis of RSC technologies. Secondly, the use of Intel Optane PMem made it possible to utilize all computational cores on the nodes to solve MPD tasks that require a large amount of RAM per core, while without Intel Optane PMem less than half of the computational cores on the nodes can be used for such tasks. And thirdly, the use of an expanded test site on DAOS (Distributed Asynchronous Object Storage) showed a significant acceleration of working with data (up to two times) and the possibility of forming a unified data buffer for various JINR computing clusters.

3. DIRAC at JINR

A service based on this platform was deployed and configured at the Joint Institute for Nuclear Research in 2016. Various tests were carried out to estimate its possibilities and performance. By 2019, standard workflow tests related to general Monte-Carlo generation were successfully completed, which proved the possibility of using DIRAC for real scientific computations [9]. At that time, MPD started experiencing the necessity of massive computing to perform centralized Monte-Carlo generation for the needs of scientific groups. At first, only Tier1 and Tier2 were used for that work, but later the “Govorun” supercomputer and the VBLHEP cluster were integrated into the system. The JINR EOS storage system was integrated and accessed via the root protocol. Authentication is based on x509 certificates, and authorization is regulated by both DIRAC and VOMS (Virtual Organization Membership Service).

There are several reasons why DIRAC Interware platform has been chosen as a tool for organization of distributed computing for MPD. First of all, it is the only solution that supports computing on all levels: workload management, data management, workflow management, user management, configuration management, etc. Second, DIRAC proved to be able to handle hundreds of thousands of jobs daily on LHCb infrastructure. It supports horizontal scaling by design: with increasing number of jobs running it is possible to setup additional copies of core services responsible for jobs and data processing. And third, it has wide and active community of users across different organizations and experiments.

Right now at JINR, DIRAC is handling two important aspects of massive computing, i.e., workload management and data management. Workload management relies on the pilot mechanism. The main idea of this approach is that user jobs are never sent directly to a queue on the computing resources. Instead, a special program called “Pilot” is sent. Its function is, firstly, to occupy a slot in the queue, and secondly, to start execution on a particular work node. When it starts, it checks various parameters of the work node it is running on. These parameters are then sent to the DIRAC Matcher service with a request for a user job. DIRAC Matcher choses a job that meets the specified parameters and submits it to the pilot. After that, the pilot initiates the user job as a child process. It enables the monitoring of the execution process, and in the case of a failure, a corresponding message is sent to DIRAC to mark the job as failed. Another feature of this approach is the simplification of the integration of heterogeneous computing resources into a unified system. Integrating a computing resource means finding a way to run DIRAC Pilot there.

The DIRAC Data Management system is also actively used during MPD generation campaigns. Surely, all files generated by user jobs can be saved to a local storage, however, it would require an additional step to go through all local storages and collect result data. The use of the DIRAC Data Management system allows writing data to a central storage element accessible via the root protocol. After all, the files are present in a storage element from where they can be accessed either via grid protocols or through local access.

Currently, DIRAC at JINR provides all the necessary tools to support massive and efficient high-throughput job submission. It can be improved by making more extensive use of DIRAC complex features, such as the Workflow Management system, and by integrating more heterogeneous resources.

4. Organization of data processing in a distributed heterogeneous computing environment

Each production (mass data physical event generation and reconstruction) goes through several steps. At each step, there are specific tools and approaches to its implementation. The following steps are distinguished:

1. Physicists make a request for data generation in the forum thread https://mpdforum.jinr.ru/c/mcprod/26. This request should contain all information about which software to use, which parameters to apply, how much data to generate, as well as other information relevant to generation.
2. When a request appears, the production manager needs to check if there is enough information to start generation. The first step is to generate a small fraction of the requested data and submit it to physicists for review. This step is important since without it the whole generation can go wrong and end up useless. Partial generation can be performed directly on one computing resource without using a distributed infrastructure for the sake of simplicity. In our case, it is usually done on the "Govorun" supercomputer. If the results are satisfactory, the next step should be taken.
3. Preparation for distributed execution. At this step, the production manager needs to analyze the profile of the computational task and decide how to run jobs on different resources. The first question is which software to use. At best, the software is already hosted in CERN Virtual Machine File System (CVMFS, https://cernvm.cern.ch/fs/). However, this is not always the case, and the production manager is responsible for compiling and testing the applied software on all resources that will be used during the production campaign. Another important aspect is to study the profile of an average computational job. By this we mean figuring out how long it takes to complete a job, what is RAM and network consumption, how much space is required in a local storage on the work node. With this information, it is possible to configure the execution of a workload on heterogeneous resources. To support this process, a special system was developed. This system will be described in more detail in the next section.
4. Distributed execution on heterogeneous computing resources. At this step, the production manager generates jobs to submit in the DIRAC system. It is important to keep in mind that some jobs can be completed with an error for various reasons. The recomputation of failed jobs should be easily accomplished. It can be achieved either by rescheduling failed jobs directly in the DIRAC system, or submitting new jobs with some fixes. During this step, the production manager keeps an eye on DIRAC Job Monitoring to identify any malfunctioning resources. Previous misbehavior is noticed, and fewer jobs need to be resubmitted or rescheduled, which means better resource utilization. During this step, it is good to have an approach and a tool to monitor the performance of different resources. A special approach was elaborated to provide this kind of data to production managers, resource administrators, and management. This approach will be described in the section 5.
5. When the generation campaign is done, the production manager needs to verify that all files are correctly placed in a storage. If not, he needs to resubmit jobs that for some reason have not written data to the storage. This work usually takes little time and in some cases can be done directly on one of the computing resources without using DIRAC to manage the jobs. After that, a message about the completion of the generation campaign should be posted on the forum, and physicists can use the generated data for their research.

5. Individual user job monitoring

When running a computer task on a local computer, it is quite easy to check its CPU load, RAM usage, and network use with simple built-in tools. However, when jobs are running in a distributed system, this problem is not so easy to tackle. It is possible to login to the work node for manual verification, but there is no direct access to many resources. Another issue is that it is usually useful to have a history of the load, i.e., a profile. Standard tools do not provide this.

To provide such a tool to DIRAC users, a special approach was proposed. It was necessary to give control over monitoring to users in those cases when they really needed it. The issue with total monitoring is related to data flow and future data analysis. For each job, CPU load, RAM usage, network I/O usage metrics should be collected and written every 30 seconds. This means that an average job that runs for 6 hours produces around 3,000 records. While for an average production campaign with 30-40 thousand jobs, the number of records may reach 100 million.

To solve this problem, a special Python application was developed to collect metrics from a running process. The procedure for enabling this monitoring requires the user to run a computational process that should be monitored in a background mode. The PID of the process should be passed to the developed application. The application will check the metrics of the running process and its children and submit them to the InfluxDB database. The collected data is accessed via the standard Chronograf interface (Figure 2).

With such monitoring, it is clearly visible that the job runs in three stages: the first short stage is generation, the second stage is simulation (it produces a large amount of data), and the third stage is reconstruction, which reads all the data and produces the result. During the reconstruction part, a memory leak in the software was detected.

The monitoring service enables users to check how their workload is behaving on the work node. It can be used when needed, and the results will be presented in a web interface with the possibility to form custom queries to the database to find jobs with the highest RAM usage, to define the average CPU usage of each job, or I/O consumption per one job.

Figure 2. Job monitoring. The plot demonstrates RAM and I/O usage.

6. Analysis of computing resources performance

When real user jobs run on a distributed infrastructure, execution information is a valuable resource for performance analysis and the comparison of different computing resources. A new approach to analyze and visualize this data was proposed at JINR [10].

Each user job submitted by the user runs not directly on a work node, but under a special process called DIRAC Pilot. When the pilot process starts, it checks all the relevant parameters of the work node it is running on: total RAM, free RAM, free disk space and CPU performance. CPU performance is measured by a special benchmark called DIRAC Benchmark 12 (DB12). This benchmark provides an estimate of the speed of a single CPU core. The data is saved to the internal DIRAC database. After completing the user job, DIRAC saves information about its wall-time duration. When there are thousands of similar jobs running in the system, it is possible to compare their durations on different resources.

The approach was elaborated for extracting data on completed jobs from the DIRAC database and presenting it in a comprehensive form. The data is visualized in a two-axis scatter plot. The X axis represents the DB12 test result for a particular job. The Y axis represents the wall time of a particular job. Thus, each job can be represented by a single dot in the plot. To add even more information, each dot is colored according to the resource the job was completed on. An example of the plot is shown in Figure 3.

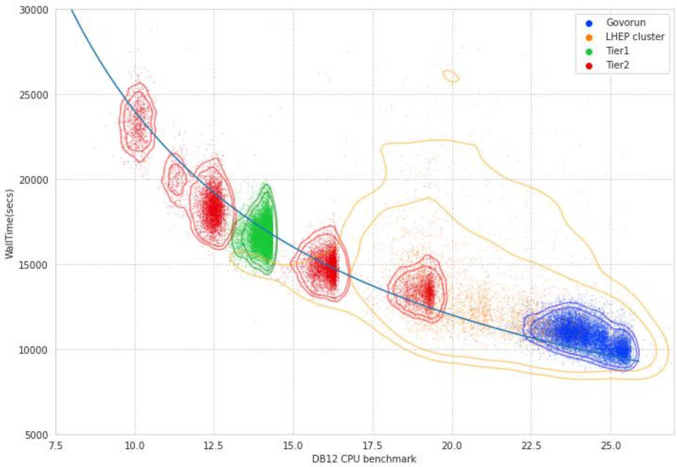

Figure 3. Example of the scatter plot with information about job duration, DB12 results, and resource ownership

The plot demonstrates that the DB12 results correlate well with the wall time. The blue line in the plot was empirically chosen to show where the dots should be placed in the ideal condition. From this plot, we can see that most of the resources give consistent duration time, except for the LHEP cluster. This is due to the lower network speed between the LHEP cluster and the storage system where result data should be placed. The two colored borders around the dot clusters represent the area with 90% and 95% of all jobs. It means that we can predict the duration of a particular job on different resources and plan the duration of the generation campaign more precisely.

This approach has a great advantage since it does not require submitting an artificial workload to the computing resources. Right now, it is required to prepare data and upload it to a Jupyter notebook for visualization. A dedicated system providing this functionality is currently under development. It should help to compare different processors and work nodes with each other on a real user workload.

7. Conclusion

The creation of a distributed heterogeneous computing environment based on the DIRAC software with advanced monitoring tools has pursued two main goals, namely, to meet the current need for the generation and reconstruction of model data and to create computing elements for the real MPD experiment. The created environment has made it possible to perform almost 857,000 jobs for the MPD experiment to date. Each job ran on average 6 hours 15 minutes on one core. The total use of computing power was 6 million Mega HEP-SPEC06 days. This effort has generated more than 650 million events, over 250 million of which have also been reconstructed (over 270 TB in storage). The practice of using various computing resources of JINR and other institutes of the MPD collaboration has shown that at the moment the most flexible resource is the “Govorun” supercomputer. This property is achieved due to the unique hardware design of the “Govorun” supercomputer, which comprises a hierarchical data processing and storage system and compute nodes with a large amount of RAM, as well as the RSC [8] software, enabling the creation of systems on demand for a specific user task. About 40% of the total number of tasks has been performed directly on the “Govorun” supercomputer.

The solution to the computing problem of the NICA megaproject required the creation of a number of research polygons, which would allow testing in practice both the latest architectural solutions and software solutions in the field of computing and data processing. The “Govorun” supercomputer plays the role of the main platform for such research. At the moment, a polygon of 8 servers containing Intel Optane PMem has been created, on which DAOS has been deployed. Research carried out at this testing site has shown a substantial performance increase on the same hardware. The IO500 benchmark data on IOPS has grown significantly from the Luster 50-clients run to the Intel DAOS 10-clients run, with a more profound advantage for more irregular (“hard”) operations, up to more than 10x. The DAOS technology looks promising in terms of enhancing the speed of working with data, and in the near future it is planned to integrate it into the distributed heterogeneous computing environment for the MPD experiment.

References

1. ATLAS experiment. URL: https://atlas.cern (Date of reference: 30.08.2021).
2. NICA – Nuclotron-based Ion Collider fAcility. URL: https://nica.jinr.ru (Date of reference: 30.08.2021).
3. Multi-Purpose Detector (MPD). URL: https://nica.jinr.ru/projects/mpd.php (Date of reference: 30.08.2021).
4. Tsaregorodtsev A. and the DIRAC Project. DIRAC Distributed Computing Services // J. Phys.: Conf. Ser. 2014. Vol. 513, No. 3. DOI: 10.1088/1742-6596/513/3/032096.
5. Heterogeneous platform “HybriLIT”. URL: http://hlit.jinr.ru (Date of reference: 30.08.2021).
6. Belyakov D.V., Nechaevskiy A.V., Pelevanuk I.S. et al. “Govorun” Supercomputer for JINR Tasks // CEUR Workshop Proceedings. 2020. No. 2772. P. 1-12. URL: http://ceur-ws.org/Vol-2772/1-12-paper-1.pdf (Date of reference: 30.08.2021).
7. Kutovskiy N., Mitsyn V., Moshkin A. et al. Integration of Distributed Heterogeneous Computing Resources for the MPD Experiment with DIRAC Interware // PEPAN. 2021. Vol. 52, No. 4.
8. RSC Group. URL: https://rscgroup.ru (Date of reference: 30.08.2021).
9. Pelevanyuk I. Performance Evaluation of Computing Resources with DIRAC Interware // AIP Conference Proceedings. 2021 (to be published).
10. Korenkov V., Pelevanyuk I., Tsaregorodtsev A. Integration of the JINR Hybrid Computing Resources with the DIRAC Interware for Data Intensive Applications // Data Analytics and Management in Data Intensive Domains. 2020. P. 31-46. DOI: 10.1007/978-3-030-51913-1_3 (Date of reference: 30.08.2021).