УДК 004.75
DOI: 10.25559/SITITO.14.201801.061-072

JINR CLOUD SERVICE FOR SCIENTIFIC AND ENGINEERING COMPUTATIONS

Nikita A. Balashov[1], Maxim V. Bashashin[1], Ruslan I. Kuchumov[2], Nikolay A. Kutovskiy[1], Ivan A. Sokolov[1]

[1] Joint Institute for Nuclear Research, Dubna, Russia

[2] Saint Petersburg State University, Saint Petersburg, Russia

Abstract

Pretty often small research scientific groups do not have access to powerful enough computational resources required for their research work to be productive. Global computational infrastructures used by large scientific collaborations can be challenging for small research teams because of bureaucracy overhead as well as usage complexity of underlying tools. Some researchers buy a set of powerful servers to cover their own needs in computational resources. A drawback of such approach is a necessity to take care about proper hosting environment for these hardware and maintenance which requires a certain level of expertise. Moreover a lot of time such resources may be underutilized because a researcher needs to spend a certain amount of time to prepare computations and to analyze results as well as he doesn't always need all resources of modern multi-core CPUs servers.

The JINR cloud team developed a service which provides an access for scientists of small research groups from JINR and its Member State organizations to computational resources via problem-oriented (i.e. application-specific) web-interface. It allows a scientist to focus on his research domain by interacting with the service in a convenient way via browser and abstracting away from underlying infrastructure as well as its maintenance. A user just sets a required values for his job via web-interface and specify a location for uploading a result. The computational workloads are done on the virtual machines deployed in the JINR cloud infrastructure.

About the authors:

Nikita A. Balashov, software engineer, Laboratory of Information Technologies, Joint Institute for Nuclear Research (6 Joliot-Curie St., Dubna 141980, Moscow region, Russia); ORCID: http://orcid.org/0000-0002-3646-0522, balashov@jinr.ru

Maxim V. Bashashin, software engineer, Laboratory of Information Technologies, Joint Institute for Nuclear Research (6 Joliot-Curie St., Dubna 141980, Moscow region, Russia); ORCID: http://orcid.org/0000-0002-2706-8668, bashashinmv@jinr.ru

Ruslan I. Kuchumov, master degree student, Saint Petersburg State University (7-9B Universitetskaya Emb., Saint Petersburg 199034, Russia); ORCID: http://orcid.org/0000-0002-8927-2111, kuchumovri@gmail.com

Nikolay A. Kutovskiy, candidate of Physical and Mathematical Sciences, senior researcher, Laboratory of Information Technologies, Joint Institute for Nuclear Research (6 Joliot-Curie St., Dubna 141980, Moscow region, Russia); ORCID: http://orcid.org/0000-0002-2920-8775, kut@jinr.ru

Ivan A. Sokolov, software engineer, Laboratory of Information Technologies, Joint Institute for Nuclear Research (6 Joliot-Curie St., Dubna 141980, Moscow region, Russia); ORCID: http://orcid.org/0000-0003-0295-5372, isokolov@jinr.ru

Keywords

Cloud technologies; software as a service; web-application; data center; virtualization; virtual machine; software-defined storage.

ОБЛАЧНЫЙ СЕРВИС ОИЯИ ДЛЯ НАУЧНЫХ И ИНЖЕНЕРНЫХ РАСЧЁТОВ

Н.А. Балашов[1], М.В. Башашин[1], Р.И. Кучумов[2], Н.А. Кутовский[1], И.А. Соколов[1]

[1] Объединенный институт ядерных исследований, г. Дубна, Россия

[2] Санкт-Петербургский государственный университет, г. Санкт-Петербург, Россия

Аннотация

Часто небольшие научные коллективы сталкиваются с дефицитом вычислительных мощностей для продуктивного выполнения своих исследовательских работ. Использование глобальных вычислительных инфраструктур, являющихся основой ИТ-платформ больших научных коллабораций, зачастую может сдерживаться высокими бюрократическими накладными расходами, а также сложностью инструментария. Некоторые исследовательские коллективы для покрытия своих нужд в вычислительных ресурсах приобретают мощные сервера, однако такой подход имеет свои негативные стороны: необходимость обеспечить соответствующие условия работы для оборудования и наличие соответствующего уровня квалификации для ввода в эксплуатацию и последующего обслуживания. Более того, подготовка запуска вычислительных задач, а также обработка и анализ их результатов требуют некоторого количества времени, в течение которого данное оборудование будет простаивать. Помимо этого, не всегда существует необходимость загружать все мощности центрального процессора современных многоядерных вычислительных систем, что тоже снижает эффективность использования системы.

Сотрудники Лаборатории информационных технологий Объединённого института ядерных исследований, занимающиеся развитием облачных технологий, разработали сервис, который предоставляет учёным и небольшим исследовательским группам ОИЯИ и организаций из его стран-участниц, испытывающим дефицит в вычислительных мощностях, возможность получения доступа к счётным ресурсам посредством проблемно-ориентированного веб-интерфейса. Данный сервис позволяет исследователю полностью сосредоточиться на выполнении научной части своей работы и абстрагироваться от сложности организации вычислительного процесса и инфраструктуры, а также её обслуживании. От пользователя данного сервиса требуется всего лишь указать в веб-интерфейсе значения для параметров его задачи и адрес для загрузки результатов счёта. Вся вычислительная часть будет выполнена на виртуальных машинах облачной инфраструктуры ОИЯИ.

Ключевые слова

Облачные технологии; программное обеспечение как услуга; веб-приложение; центр обработки данных; виртуализация; виртуальная машина; программно-определяемое хранилище данных.

Introduction

The JINR scientists participate in a lot of scientific research projects. Some JINR research groups are part of the large international collaborations which use global computing infrastructures (e.g. such as World-wide LHC Computing Grid [1]) for data management, processing and analysis. But some research teams are doing on their own including an in-house development of applications. Pretty often such researchers do not have access to powerful enough resources required for their work to be productive. It's pretty wide-spread practice to do computations on own laptops or PCs which may last days or even weeks. Others buy powerful desktop servers and offload computational work there. However possible drawbacks of such approach are the following:

A necessity to take care about proper hosting environment for these hardware (it's common to place such hardware in the office where temperature and humidity might be far from optimal for servers operational mode especially during warm seasons. That leads to additional expenses for climate equipment what in turn might decrease a comfort level for humans located in the same office not only due to additional source of noise and lower temperature. Moreover it's very desirable to run such pretty expensive hardware as servers using uninterruptible power supply (UPS) which is not always the case for office;

Equipment maintenance (operating system installation and updates, firewall settings, shared network storage if needed, so on) what requires a certain level of expertise;

Resources under-utilization (researcher needs to spend a certain amount of time to prepare computations and to analyze results as well as he doesn't always need all resources of modern multi-core CPUs servers).

So taking listed above drawbacks into account the JINR cloud team developed a service which provides an access for scientists of small research groups from JINR and its Member State organizations to computational resources via problem-oriented (i.e. application-specific) web-interface.

Implementation

One of the motivation behind the JINR cloud infrastructure [2] deployment were to simplify an access for scientists to computational resources as well as to increase a hardware utilization what cloud technologies cover both.

Normally a user interacts with the JINR cloud via web-interface which allows him to manage virtual machines (VMs) within his quotas and use those VMs as intended. But even such workflow can be simplify by creating application-specific web-forms where user just needs to set a proper values for parameters of certain application, define an amount of required cloud resources and specify a location for job results. An implementation of such approach transforms the JINR cloud from pure Infrastructure-as-a-Service platform (IaaS) into Software-as-a-Service (SaaS) one – the JINR cloud service for scientific and engineering computations (or JINR SaaS for short).

The main components of that service as well as a basic schema of a workflow are shown on the Figure 1 and described below.

Web-portal

A user interacts with the whole service via web-portal only which is a client-server web-application. Its server part is based on a Django framework [3] which implements so called "Model-Template-View" architecture. That web-application communicates with IdleUtilizer component via remote procedure calls (RPC) encoded in XML format (i.e. via XML-RPC).

A client part of the service which user interacts with via his browser was developed using Bootstrap 3, AJAX, JavaScript, HTML5 and CSS. It allows to select a particular application, define its input parameters (it also validates them) and to initiate a job submission. The server part of the web-app gets these information and based on it forms a request for IdleUtilizer.

IdleUtilizer

Initially the IdleUtilizer (IU) was developed to increase an overload efficiency of the JINR cloud facility utilization at the cost of loading its idle resources by jobs from the HTCondor batch system [4] used by NOvA experiment [5]. But when a necessity to develop the JINR cloud service for scientific and engineering computations appeared it was decided to extend IU's functionality.

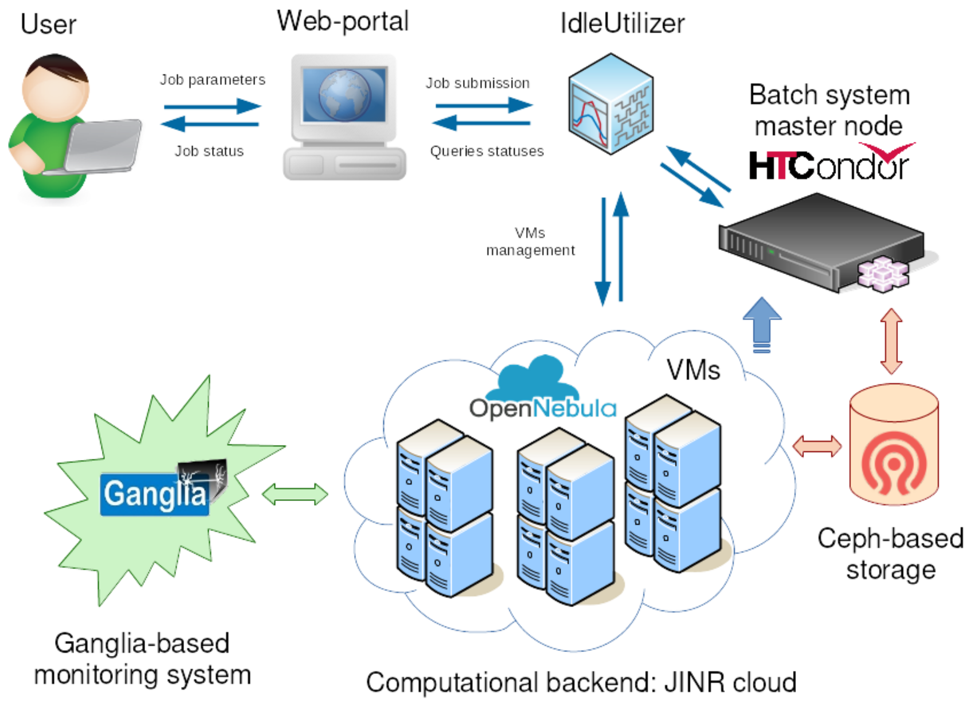


Figure 1. A basic schema of workflow and main components of the JINR cloud service for scientific and engineering computations

In that context the IdleUtilizer plays two main tasks:

- gets information defined by user for computational jobs execution including characteristics of required resources as well as input parameters for computational job;
- manages VMs in the cloud (instantiates requests for VMs creation, checks requests status, deletes VMs, etc) and users' jobs in a batch system (submits user job, checks its status, cancel submitted job upon user request and so on).

All IU elements were designed to be flexible, testable and easily modifiable in case of changes in external services' interfaces. At the time of writing that article the IU supports OpenNebula-based [6] clouds and HTCondor-based batch systems.

Currently IU consists of the following elements:

src.config which is responsible for reading of configuration files as well as their representation in a form understandable by other IU's elements);

src.config_generator allows to create configuration file with default values;

src.logger initializes and defines IU events logging;

src.daemon provides an ability to work in daemon mode in a background;

src.rpcapi defines RPC API methods and wraps other modules methods provided via RPC API;

src.rpcserver which is responsible for RPC stream and its API registration;

libs.schedulers.htcondor is a driver for HTCondor. It hides low-level and HTCondor-specific methods. Interactions with HTCondor master node is done by executing BASH-scripts via SSH-protocol.

src.schedulers loads a corresponding batch driver and delegates queries to it;

libs.jobs.htcondorjob allows to create job files for HTCondor using a template, upload them to remote server and delete them;

src.jobs which is responsible for loading job template for certain driver and delegates queries to it;

libs.clouds.opennebula is a driver for OpenNebula. It hides low-level and OpenNebula-specific methods. Interactions with that cloud platform is done via its XML-RPC protocol.

src.clouds loads cloud driver and delegates queries to it;

src.request_stogate provides an interface for database. Only TinyDB which keeps its state in a JSON-file is supported at the time of writing that article.

src.request implements operations needed for requests processing (creation of VM, job file and so on).

HTCondor batch system

Each separate user job initiates a creation of dedicated set of computational resources – virtual machines in the JINR cloud infrastructure. These VMs become worker nodes for HTCondor-based batch system.

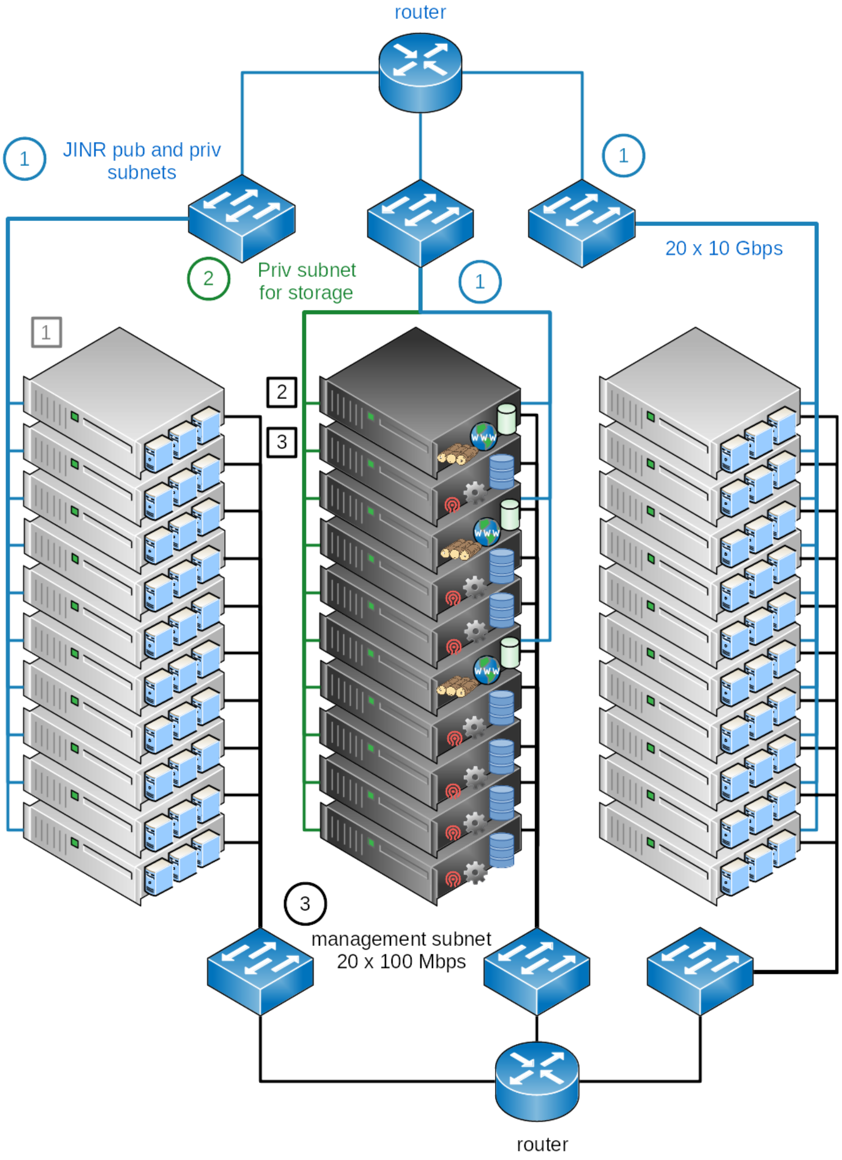


Figure 2. The JINR cloud Raft HA architecture

OpenNebula-based cloud infrastructure

The JINR SaaS implements modular architecture to enable job submission to various computational back-ends. At the moment of writing that article it's possible to use OpenNebula-based clouds where VMs are created for running a real workload. The JINR cloud is used as a computational back-end for the service for scientific and engineering computations. That cloud infrastructure a schema of which architecture is shown on the Figure 2 uses a distributed consensus protocol to provide fault-tolerance and state consistency across its services. According to OpenNebula documentation a consensus algorithm relies on two concepts:

System State what in the case of OpenNebula-based clouds means the data stored in the database tables (users, ACLs, or the VMs in the system);

Log what is a sequence of SQL statements that are consistently applied to the OpenNebula DB in all servers to evolve the system state.

To preserve a consistent view of the system across servers, modifications to system state are performed through a special node called the "leader". The OpenNebula cloud front-end nodes (CFNs) elect a single node to be the leader. The leader periodically sends heartbeats to the other CFNs called followers to keep its leadership. If a leader fails to send the heartbeat followers promote to candidates and start a new election.

Whenever the system is modified (e.g. a new VM is added to the cluster), the leader updates the log and replicates the entry in a majority of followers before actually writing it to the database. It increases the latency of DB operations but enables a safe replication of the system state and the cluster can continue its operation in case of leader failure.

Following the OpenNebula documentation recommendations the JINR cloud has odd number of front-end nodes (it is three in our case and they are represented on the Figure 2 identically to one marked by the black numeral "2" in the same color square) which provides a fault-tolerance for 1 node.

KVM-based Virtual machines (VMs) [7] and OpenVZ-based containers (CTs) [8] are running on cloud worker nodes (CWNs) marked on the Figure 2 by numeral "1" in a grey square.

All CFNs and CWNs are connected through 10 GbE network interfaces to the corresponding rack switch which in its turn are connected to the router.

Apart from the locally deployed CWNs the JINR cloud has some amount of external resources from the partner organizations of JINR Member State (see [9] for more details) which also can be transparently accessed via the same web-interface.

Ceph-based software defined storage

Other key component of the JINR cloud infrastructure and the JINR SaaS service is a software-defined storage based on Ceph [10]. It delivers object, block and file storage in one unified system. According to Ceph documentation each ceph server can play a single or few roles which are the following:

Monitor (mon). It maintains maps of the cluster state, including the monitor map, manager map, the OSD map, and the CRUSH map. These maps are critical cluster state required for Ceph daemons to coordinate with each other. Monitors are also responsible for managing authentication between daemons and clients. At least three monitors are normally required for redundancy and high availability;

Manager (mgr). It is responsible for keeping track of runtime metrics and the current state of the Ceph cluster, including storage utilization, current performance metrics, and system load. The Ceph Manager daemons also host python-based plugins to manage and expose Ceph cluster information, including a web-based dashboard and REST API. At least two managers are normally required for high availability;

Object storage daemon (OSD). It stores data, handles data replication, recovery, rebalancing, and provides some monitoring information to Ceph Monitors and Managers by checking other Ceph OSD Daemons for a heartbeat. At least 3 Ceph OSDs are normally required for redundancy and high availability;

Metadata Server (MDS). It stores metadata on behalf of the Ceph Filesystem. Ceph MDSs allow POSIX file system users to execute basic commands (like ls, find, etc.) without placing an enormous burden on the Ceph Storage Cluster.

Apart from these daemons there is RADOS gateway (RGW) which provides interfaces for interacting with the storage cluster.

The JINR Ceph-based SDS deployment schema as well as each server roles are shown on the Figure 3.

Ganglia-based monitoring system

To get an information about various metrics (CPU and memory usage as well as disk and network i/o) of HTCondor WNs what are deployed and run on cloud VMs a ganglia-based monitoring system [11] is used. One of its advantage over other similar systems is that topology-agnostic what means nodes monitored by ganglia-agents (called gmond) can dynamically appears and disappears without a necessity to restart core services on the ganglia head node, i.e. a topology of a monitored system can dynamically changed.

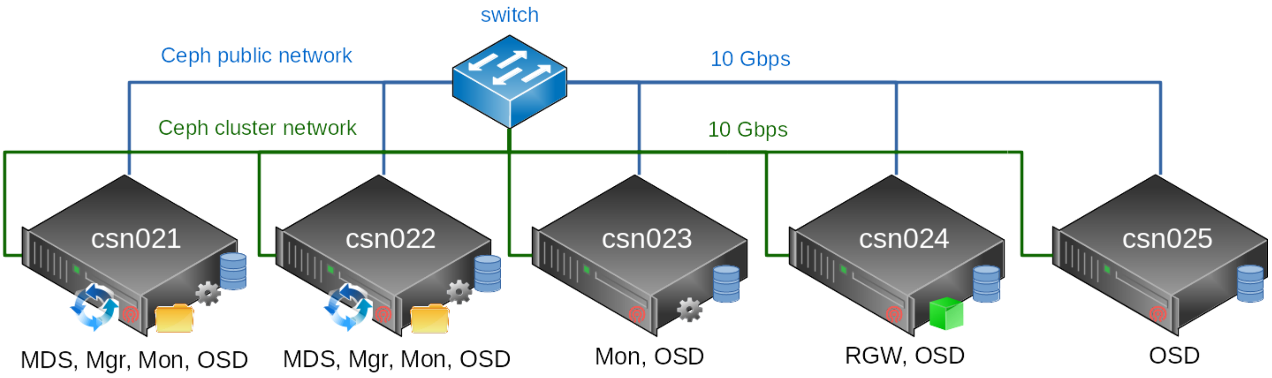


Figure 3. The schema of the Ceph-based software-defined storage deployed at JINR

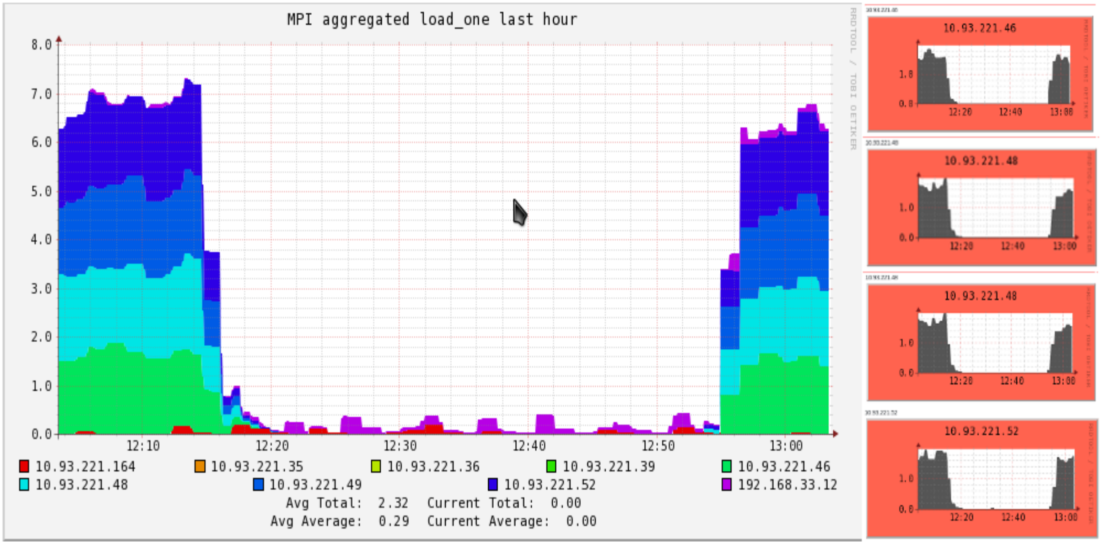


Figure 4. A screenshot of ganglia web-interface with overview of WNs and its CPU load

It's exactly the case when HTCondor WNs are created on cloud VMs upon user requests by submitting a job via the JINR SaaS web-portal.

A screenshot of ganglia web-interface with overview of WNs and its CPU load is shown on the Figure 4.

So the complete workflow can be described as below.

A user logs in on the web-portal via his web-browser. Initially a first tab "Creating a job" is active where the user needs to do few actions:

- choose a certain app he wants to run;
- define a set of values for the parameters of selected app if there are any;
- specify an URL for uploading results including job's stderr and stdout files;
- press "Submit" button.

As soon as "Submit" button is pressed the user is redirected to "Jobs results" tab of the web-interface. At the same time the server part of web-app contacts IdleUtilizer by sending to it a request with user-defined values for computational resources and input parameters. IU validates these request and queries the JINR cloud service to create a specified number of VMs with given size in terms of CPU cores and amount of RAM. If a total amount of requested resources is within user's quotas then VMs are instantiated from predefined template and disk image. At that stage the user sees "pending" status in a "Status" column of the web-interface. After that IU tries to reach VMs via network what corresponds to "booting" status. As it was mentioned before all VMs are created from single VM template and image. The last one has a set of pre-installed software components such as HTCondor worker node (to transform that VM into batch system computational host), ceph client (for mounting network share from distributed storage where application itself is installed and will be executed from as well as output results will be produced) and ganglia monitoring client (for enabling a possibility to track VM's metrics and status). As soon as operating system (OS) on VMs is booted the ganglia monitoring client starts to send monitoring metrics to ganglia head node and the ceph client mounts remote share as well as the HTCondor worker nodes (WNs) daemon starts and tries to reach the HTCondor master node to report about own readiness to take a load. During that stage the IU attempts to get WNs hostnames from HTCondor master node what corresponds to "bootstrapping" status in terms of IU. As soon as WNs hostnames become known the IU submits a user job to HTCondor queue. "Executing" status means that the job is executing on the WNs. When the job is finished successfully the user in the web-portal sees status "terminating" which means the IU sends a request for VMs termination. After that the job gets status "Done" and the user can download the output files at the specified URL.

For the moment there are interfaces in the web-portal for two applications:

"Hello test" which is used for testing purposes and "Long Josephson junctions simulation" which allows to simulate Long Josephson junctions (LJJ) using in-house developed application [12] by collaboration of colleagues from Bogoliubov Laboratory of Theoretical Physics and Laboratory of Information Technologies (LIT).

A brief description of the web-interface for LJJ simulation is given below.

The web-portal of the JINR SaaS is available at the URL: http://saas.jinr.ru. Its login page looks like on the Figure 5.

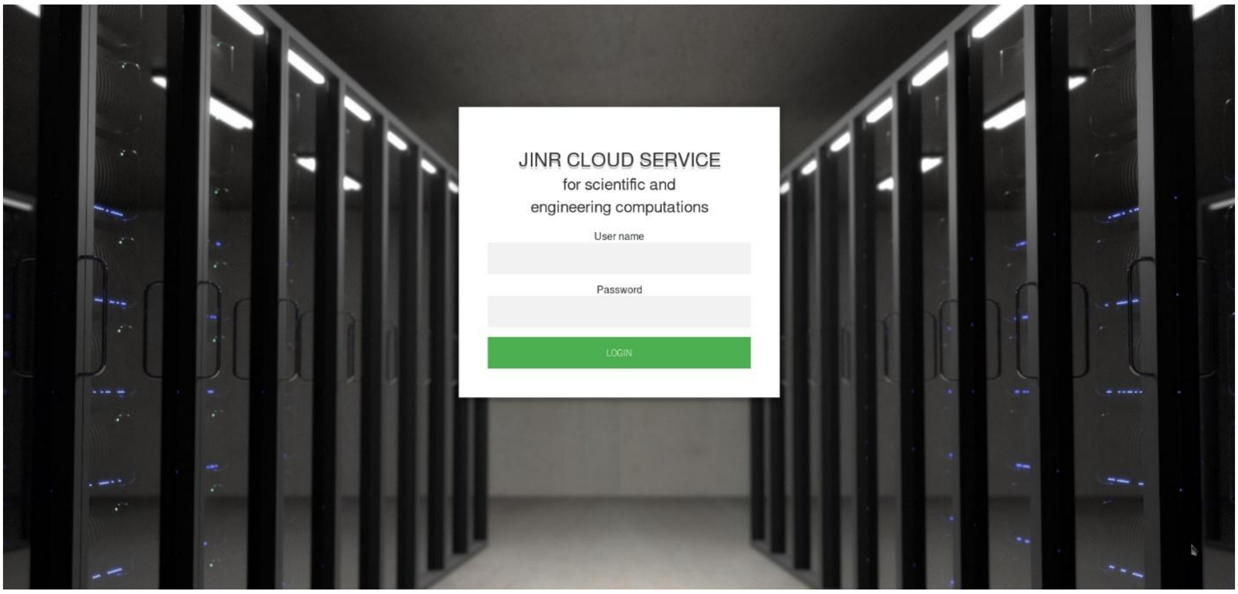
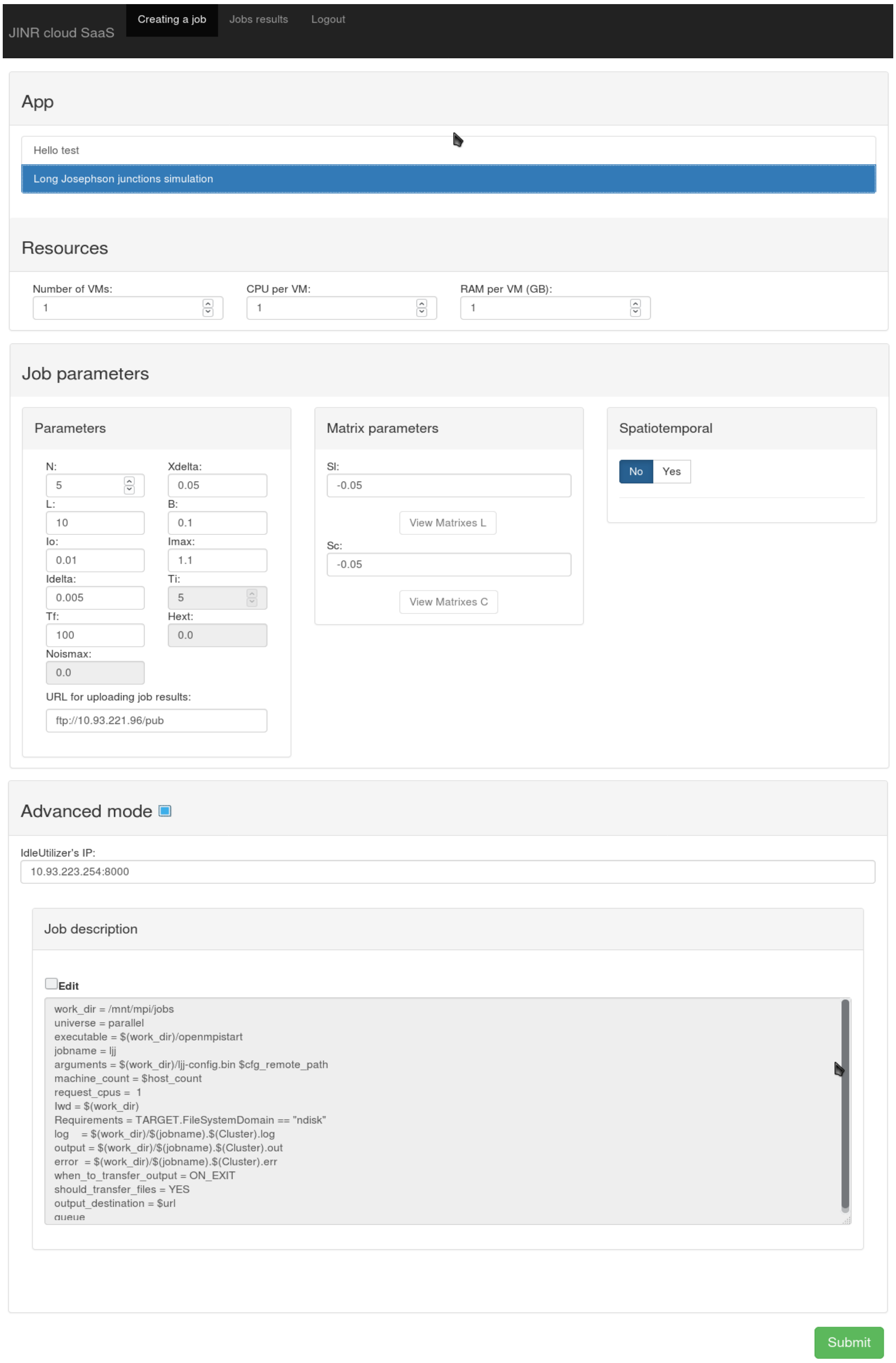

JINR CLOUD SERVICE
for scientific and
engineering computations

Figure 5. A screenshot of the login page of the JINR SaaS



App

Hello test

Long Josephson junctions simulation

Resources

Number of VMs: 1

CPU per VM: 1

RAM per VM (GB): 1

Job parameters

Parameters

N: 5

Xdelta: 0.05

L: 10

B: 0.1

Io: 0.01

Imax: 1.1

Idelta: 0.005

Ti: 5

Tf: 100

Hext: 0.0

Noismax: 0.0

URL for uploading job results: ftp://10.93.221.96/pub

Matrix parameters

Sl: -0.05

Sc: -0.05

Spatiotemporal

[✓ No] Yes

Advanced mode ☑

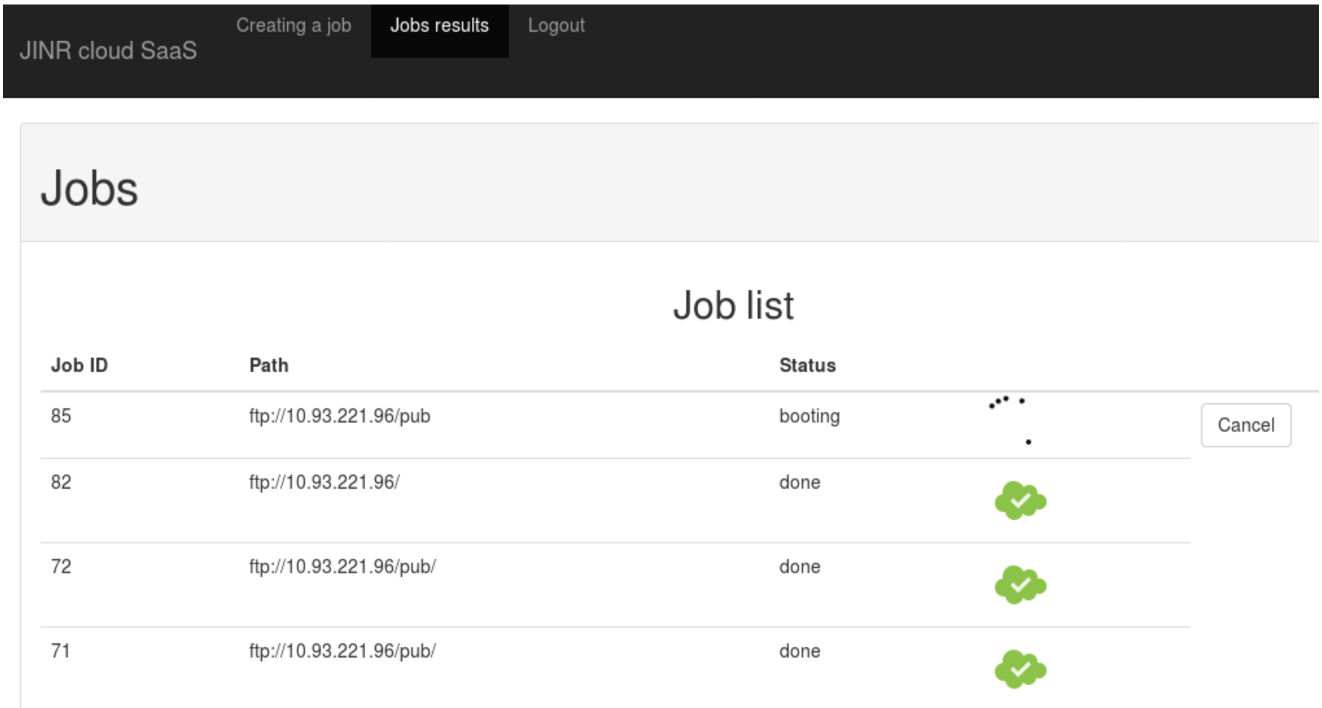
IdleUtilizer's IP: 10.93.223.254:8000

Job description

□ Edit

```
work_dir = /mnt/mpi/jobs
universe = parallel
executable = $(work_dir)/openmpistart
jobname = ljj
arguments = $(work_dir)/ljj-config.bin $cfg_remote_path
machine_count = $host_count
request_cpus =  1
lwd = $(work_dir)
Requirements = TARGET.FileSystemDomain == "ndisk"
log    = $(work_dir)/$(jobname).$(Cluster).log
output = $(work_dir)/$(jobname).$(Cluster).out
error  = $(work_dir)/$(jobname).$(Cluster).err
when_to_transfer_output = ON_EXIT
should_transfer_files = YES
output_destination = $url
queue
```

Figure 6. A screenshot of the web-page where the user can choose required application and set a values for its parameters



Job list

Job ID	Path	Status		
85	ftp://10.93.221.96/pub	booting		Cancel
82	ftp://10.93.221.96/	done	✓	
72	ftp://10.93.221.96/pub/	done	✓	
71	ftp://10.93.221.96/pub/	done	✓	

Figure 7. A screenshot of the "Jobs results" tab with job statuses

To start working a user has to be authenticated by the JINR SaaS service. After successful authentication the user will be able to choose certain application as well as set a values for its parameters (see Figure 6). An advanced mode provides an ability to change IdleUtilizer's IP address as well as to edit a HTCondor job description. That mode can be activated by enabling a corresponding checkbox. It's useful for debug purposes and normally shouldn't be used by the regular users.

As soon as the user set the values for all required parameters one needs to click on the "Submit" button to submit a job. After that step the user will be redirected to the tab "Jobs results" where he can track a progress of a job execution. A screenshot of corresponding web-page is shown on the Figure 7.

Resources

Some amount of computational resources in the JINR cloud researchers get at no cost. But if a necessity in them is higher than such share then scientists may consider a scenario of buying more resources at own cost, hosting them at the LIT data center with proper environment already set (UPSes, climate equipment, censors and monitoring system) and integrating bought capacities into the JINR cloud. A corresponding user or group quotas on resources are set to the values within bought ones. It guarantees that the owner of these resources will always have at his service such amount of them with some agreed maximum delay. The other side of the agreement is that idle resources can be used for other projects and tasks. Whenever an owner of such resources will request them he will get ones within certain period because running on those resources jobs need to be finished.

Future plans

It is planned in the nearest future to add a HybriLIT heterogeneous cluster as one more computational back-end of the JINR SaaS service.

Apart from that more applications developed by research groups from JINR and its Member State organizations are planned to add into that service.

A work on adding on the "Creating a job" tab an information about free and occupied resources within user quotas is in progress. Such feature will help users to determine currently available cloud resources and will make the job submission process less error-prone.

Conclusion

The JINR cloud team developed a service which provides an access for scientists of small research groups from JINR and its Member State organizations to computational resources via application-specific web-interface. It allows scientist to focus on his research domain by interacting with the service in a convenient way via browser and abstracting away from underlying infrastructure as well as its maintenance. A user just set a required values for his job via web-interface and specify a location for uploading a result. The computational workload are done on the VMs deployed in the JINR cloud infrastructure.

Some amount of the resources are provided for the researcher for free. But if more ones are required there is an option to buy capacities at own cost and integrated them into the JINR cloud at the LIT data center where proper environment for hardware is already set. That way an overall hardware utilization efficiency is increased to due a possibility to be used by other users of the JINR cloud when such capacities are idle.

Acknowledgment

A work on the JINR cloud service for scientific and engineering computations was supported by RFBR grant 15-29-01217.

REFERENCES

[1] Bird I. Computing for the Large Hadron Collider. *Annual Review of Nuclear and Particle Science*. 2011; 61:99-118. DOI: https://doi.org/10.1146/annurev-nucl-102010-130059.

[2] Baranov A.V., Balashov N.A., Kutovskiy N.A., Semenov R.N. JINR cloud infrastructure evolution. *Physics of Particles and Nuclei Letters*. 2016; 13(5):672 – 675. DOI: https://doi.org/10.1134/S1547477116050071

[3] Greenfeld D., Greenfeld A. Two Scoops of Django: Best Practices for Django 1.8 (3rd ed.). Publisher: Two Scoops Press. p. 531.

[4] Thain D., Tannenbaum T., Livny M. Distributed Computing in Practice: The Condor Experience. *Concurrency and Computation: Practice and Experience*. 2005;17(2-4): 323-356. DOI: https://doi.org/10.1002/cpe.938

[5] Adamson P. et al. (NOvA Collab.) First measurement of electron neutrino appearance in NovA. *Physical Review Letters*. 2016; 116(15), id. 151806. DOI: https://doi.org/10.1103/PhysRevLett.116.151806

[6] Moreno-Vozmediano R., Montero R.S., Llorente I.M. IaaS Cloud Architecture: From Virtualized Datacenters to Federated Cloud Infrastructures. *IEEE Computer*. 2012; 45(12):65-72. DOI: https://doi.org/10.1109/MC.2012.76

[7] Kivity A., Kamay Y., Laor D., Lublin U., Liguori A. KVM: the Linux virtual machine monitor. Proceedings of the Linux Symposium, 27–30 June, 2007. Vol. 1, Ottawa, Canada, 2007. p. 225–230.

[8] Marshall D. Virtualization Comes in More than One Flavor // Virtualization Technology News and Information, January 13, 2007. Available at: http://vmblog.com/archive/2007/01/13/virtualization-comes-in-more-than-one-flavor.aspx#.WtdwhNRubcu (accessed 10.01.2018).

[9] Balashov N.A. et al. JINR Member States cloud infrastructure. *CEUR Workshop Proceedings*. 2017; 2023:202-206. Available at: http://ceur-ws.org/Vol-2023/122-128-paper-19.pdf (accessed 10.01.2018).

[10] Weil S.A., Brandt S.A., Miller E.L., Long D.D.E., Maltzahn C. Ceph: a scalable, high-performance distributed file system. *Proceedings of the 7th symposium on Operating systems design and implementation (OSDI '06)*. USENIX Association, Berkeley, CA, USA, 2006. p. 307-320.

[11] Massie M., Chun B., Culler D. The Ganglia Distributed Monitoring System: Design, Implementation, and Experience. *Parallel Computing*. 2004; 30:817-840. DOI: https://doi.org/10.1016/j.parco.2004.04.001

[12] Bashashin M.V., Zemlyanay E.V., Rahmonov I.R., Shukrinov J.M., Atanasova P.C., Volokhova A.V. Numerical approach and parallel implementation for computer simulation of stacked long Josephson Junctions. *Computer Research and Modeling*. 2016; 8(4):593-604. Available at: http://crm.ics.org.ru/uploads/crmissues/crm_2016_4/16.08.01.pdf (accessed 10.01.2018). (In Russian)

[13] Alexandrov E.I. et al., Research of Acceleration Calculations in Solving Scientific Problems on the Heterogeneous Cluster HybriLIT. *RUDN Journal of Mathematics, Information Sciences and Physics*. 2015; 4:30-37. Available at: http://journals.rudn.ru/miph/article/view/8218 (accessed 10.01.2018).

Submitted 10.01.2018; Revised 20.02.2018; Published 30.03.2018.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] *Bird I.* Computing for the Large Hadron Collider // Annual Review of Nuclear and Particle Science. 2011. Vol. 61. Pp. 99-118. DOI: https://doi.org/10.1146/annurev-nucl-102010-130059.

[2] *Baranov A.V., Balashov N.A., Kutovskiy N.A., Semenov R.N.* JINR cloud infrastructure evolution // Physics of Particles and Nuclei Letters. 2016. Vol. 13, no 5. Pp. 672 – 675. DOI: https://doi.org/10.1134/S1547477116050071

[3] *Greenfeld D., Greenfeld A.* Two Scoops of Django: Best Practices for Django 1.8 (3rd ed.). Publisher: Two Scoops Press, p. 531.

[4] *Thain D., Tannenbaum T., Livny M.* Distributed Computing in Practice: The Condor Experience // Concurrency and Computation: Practice and Experience. 2005. Vol. 17, no. 2-4. Pp. 323-356. DOI: https://doi.org/10.1002/cpe.938

[5] *Adamson P.* et al. (NOvA Collab.) First measurement of electron neutrino appearance in NovA // Physical Review Letters. 2016. Vol. 116, no. 15, id. 151806. DOI: https://doi.org/10.1103/PhysRevLett.116.151806

[6] *Moreno-Vozmediano R., Montero R.S., Llorente I.M.* IaaS Cloud Architecture: From Virtualized Datacenters to Federated Cloud Infrastructures // IEEE Computer. 2012. Vol. 45, no. 12. Pp. 65-72. DOI: https://doi.org/10.1109/MC.2012.76

[7] KVM: the Linux virtual machine monitor / A. Kivity, Y. Kamay, D. Laor, U. Lublin, A. Liguori // Proceedings of the Linux Symposium, 27–30 June, 2007. Vol. 1, Ottawa, Canada, 2007. Pp. 225–230.

[8] *Marshall D.* Virtualization Comes in More than One Flavor // Virtualization Technology News and Information, January 13, 2007. URL: http://vmblog.com/archive/2007/01/13/virtualization-comes-in-more-than-one-flavor.aspx#.WtdwhNRubcu (дата обращения: 10.01.2018).

[9] JINR Member States cloud infrastructure / N. Balashov [et al.] // CEUR Workshop Proceedings. 2017. Vol. 2023. Pp. 202-206. URL: http://ceur-ws.org/Vol-2023/122-128-paper-19.pdf (дата обращения: 10.01.2018).

[10] Ceph: a scalable, high-performance distributed file system / S.A. Weil, S.A. Brandt, E.L. Miller, D.D.E. Long, C. Maltzahn C. // Proceedings of the 7th symposium on Operating systems design and implementation (OSDI '06). USENIX Association, Berkeley, CA, USA, 2006. Pp. 307-320.

[11] *Massie M., Chun B., Culler D.* The Ganglia Distributed Monitoring System: Design, Implementation, and Experience // Parallel Computing. 2004. Vol. 30. Pp. 817-840. DOI: https://doi.org/10.1016/j.parco.2004.04.001

[12] *Башашин М.В., Земляная Е.В., Рахмонов И.Р., Шукринов Ю.М., Атанасова П.Х., Волохова А.В.* Вычислительная схема и параллельная реализация для моделирования системы длинных джозефсоновских переходов // Компьютерные исследования и моделирование. 2016. Т. 8, № 4. С. 593-604. URL: http://crm.ics.org.ru/uploads/crmissues/crm_2016_4/16.08.01.pdf (дата обращения: 10.01.2018)

[13] *Александров Е.И.* и др. Исследование ускорения вычислений при решении научных задач на гетерогенном кластере HybriLIT // Вестник Российского университета дружбы народов. Серия: Математика. Информатика. Физика. 2015. № 4. С. 30-37. URL: http://journals.rudn.ru/miph/article/view/8218 (дата обращения: 10.01.2018).

Поступила 10.01.2018; принята к публикации 20.02.2018; опубликована онлайн 30.03.2018.

Об авторах:

Балашов Никита Александрович, инженер-программист 1 категории, Лаборатория информационных технологий, Объединённый институт ядерных исследований (141980, Россия, Московская область, г. Дубна, ул. Жолио-Кюри, д. 6); ORCID: http://orcid.org/0000-0002-3646-0522, balashov@jinr.ru

Башашин Максим Викторович, инженер-программист, Лаборатория информационных технологий, Объединённый институт ядерных исследований (141980, Россия, Московская область, г. Дубна, ул. Жолио-Кюри, д. 6); ORCID: http://orcid.org/0000-0002-2706-8668, bashashinmv@jinr.ru

Кучумов Руслан Ильдусович, магистрант, Санкт-Петербургский государственный университет (199034, Россия, г. Санкт-Петербург, Университетская наб., д. 7–9Б); ORCID: http://orcid.org/0000-0002-8927-2111, kuchumovri@gmail.com

Кутовский Николай Александрович, кандидат физико-математических наук, старший научный сотрудник, Лаборатория информационных технологий, Объединённый институт ядерных исследований (141980, Россия, Московская область, г. Дубна, ул. Жолио-Кюри, д. 6); ORCID: http://orcid.org/0000-0002-2920-8775, kut@jinr.ru

Соколов Иван Александрович, инженер-программист, Лаборатория информационных технологий, Объединённый институт ядерных исследований (141980, Россия, Московская область, г. Дубна, ул. Жолио-Кюри, д. 6), ORCID: http://orcid.org/0000-0003-0295-5372, sokolov@jinr.ru