Письма в ЭЧАЯ. 2016. Т. 13, № 5(203). С. 1046–1050

JINR CLOUD INFRASTRUCTURE EVOLUTION

A. V. Baranov [a], *N. A. Balashov* [a],
N. A. Kutovskiy [a,b,1], *R. N. Semenov* [a,b]

[a] Joint Institute for Nuclear Research, Dubna

[b] Plekhanov Russian University of Economics, Moscow

To fulfil JINR commitments in different national and international projects related to the use of modern information technologies such as cloud and grid computing, as well as to provide a modern tool for JINR users for their scientific research, a cloud infrastructure was deployed at the Laboratory of Information Technologies of the Joint Institute for Nuclear Research. OpenNebula software was chosen as a cloud platform. Initially it was set up in simple configuration with single front-end host and a few cloud nodes. Some custom development was done to tune JINR cloud installation to fit local needs: web form in the cloud web-interface for resources request, a menu item with cloud utilization statistics, user authentication via Kerberos, custom driver for OpenVZ containers. Because of high demand for that cloud service and its resources over-utilization it was redesigned to cover increasing users' needs in capacity, availability, and reliability. Recently a new cloud instance has been deployed in high-availability configuration with distributed network file system and additional computing power.

PACS: 89.20.Ff

INTRODUCTION

JINR participates in various scientific projects which are relied on the use of modern information technologies including grid and cloud ones. These activities assume not only software usage by scientists but a wide spectrum of such tasks as maintenance of hardware facilities, deployment of services, application software development, etc. To cover all these needs and to increase an overall efficiency of the Laboratory of Information Technologies (LIT) IT-infrastructure functioning (more efficient servers and services management, better hardware utilization, higher services and storage systems reliability), a cloud infrastructure (hereinafter referred to as "JINR cloud service", "cloud service", "cloud") was deployed at LIT JINR.

JINR cloud service is built upon the Infrastructure as a Service (IaaS) model. Such a model provides network access to computational, software, and information resources (networks, servers, storage devices, services, and application software), allowing one to allocate resources on-demand according to dynamically changing requirements. Cloud users can obtain, configure, and deploy VMs themselves with the minimal assistance of the IT specialists. The cloud service is expected to reduce the total cost of ownership of the computing infrastructure and also to reduce its support complexity. To tune cloud service for our users', administrators' and managers' needs, some in-house developments were done. All these aspects are described below.

[1] E-mail: kut@jinr.ru

CURRENT SETUP

Initially JINR cloud was deployed in a simple setup [1]: a set of Cloud Nodes (CNs), where virtual instances are run, and a single front-end (FN) cloud node with OpenNebula core, scheduler, MySQL database server, web and command line interfaces, datastores located on local FN hard disk.

But due to such disadvantages of simple setup as FN is a single point of failure, impossibility to extend FN disk space on the fly, maintenance downtime for server where FN is running, and because of high demand in JINR cloud service as well as its resources over-utilization, it has been redesigned to cover increasing users' needs in capacity, availability, and reliability. Two high available (HA) cloud configurations were deployed and tested.

The first one was based on two KVM [2] VMs hosted on two separated physical servers and configured in HA cluster with Red Hat HA Cluster suite [3]. Each KVM VM contained an identical set of OpenNebula components such as OpenNebula core, scheduler, MySQL database server, graphical Sunstone web-interface (accessible at shared public IP address), rOCCI-server [4], and Red Hat HA Cluster suite. All of them were running at the same time only on one FN called "active" (or FN1). The second FN ("passive" or FN2) had only two running components: 1) MySQL DB configured in "master"–"master" mode with the DB on the "active" FN to keep data consistency and 2) Red Hat HA Cluster suite was checked within defined period of time if the "active" FN was alive. If FN2 did not get a proper response from FN1, then the "passive" FN considered the "active" FN as dead and took its role: it changed network settings to keep all cloud services available at the same URLs and launched all cloud components.

The second HA testbed was deployed on OpenVZ [5] containers (CTs) with Distributed Replicated Block Device (DRBD) software [6] and the cluster manager Heartbeat [7]. Two physical servers were used as well. Each of them had OpenVZ containers support enabled and two partitions: one was for operating system; the second one, for where OpenVZ CT was running. One server was configured as a "master" DRBD; another, as a "slave". The Heartbeat service was installed on both servers. Each of them listened to the other. In case of failure of the "active" server where the CT with all cloud components was running, the Heartbeat service hosted on the "passive" server detected that failure and ran the same CT on the former "passive" node which became "active".

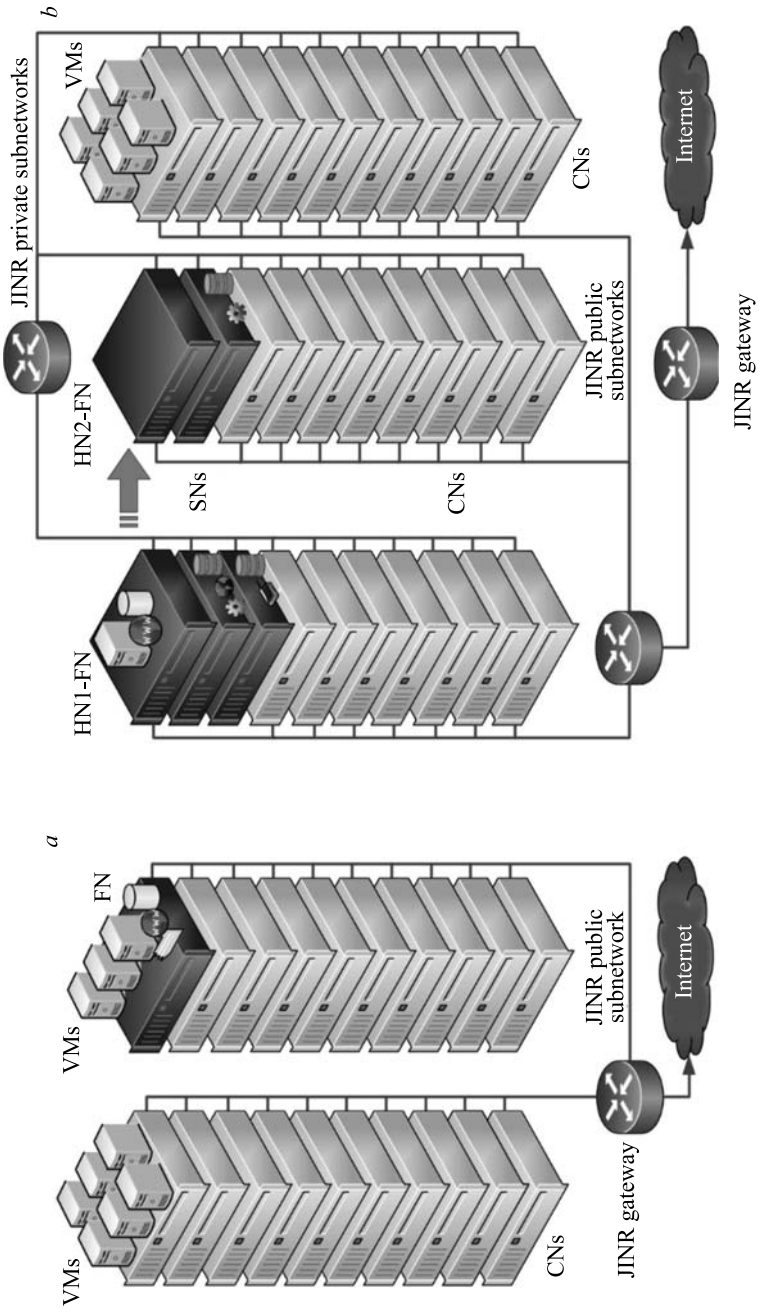
After a bunch of tests performed on both HA clusters the second one, i.e., based on OpenVZ containers, was chosen to be a basis for a new JINR HA cloud.

Schemas of the simple cloud setup and the HA one are shown in the Figure.

Two hardware nodes (HNs) marked in the Figure, *b* as HN1-FN and HN2-FN are physical servers of HA cluster with one dedicated DRBD partition in each node where CT with all cloud components is running.

To be able dynamically extend cloud datastores capacities and to increase a reliability of data preservation, a distributed network file system (DNFS) based on LizardFS [8] was deployed and configured as a storage backend for cloud. It consists of three storage nodes (SNs) with the following roles each: cloudfs1 is a master, chunk-server and web-based admin GUI; cloudfs2 is metalogger and chunk-server; cloudfs3 is a shadow and chunk-server. Total capacity of that storage is 16 TB, but due to redundancy set to 2 the effective size is 8 TB.

To split external network traffic (related to interactions with either cloud web-GUI or APIs, images uploading/downloading, etc.) and internal one, all cloud hosts were connected



Simple (*a*) and high available (*b*) schemas of JINR cloud setups

to both public and private JINR subnetworks. Because of poor network bandwidth among CNs (1 Gbit/s only), all VMs and CTs are running on local CN hard disk drives and not on DNFS share directly. All cloud resources are split into few clusters depending on virtualization (KVM or OpenVZ) and network (private or public) types. There are two user interfaces: graphical web-interface accessible at URL http://cloud.jinr.ru and command line interface accessible via ssh protocol on the same hostl.

Cloud servers and most critical cloud components are monitored by a dedicated monitoring service (available at URL http://cloud-mon.jinr.ru/nagios) based on Nagios software [9]. In case of any problem with monitored objects, Cloud administrators get notifications via SMS and email.

At the time of writing this article JINR HA cloud had 200 CPU cores, 400 GB of RAM, about 100 running VMs and CTs, as well as 77 registered users.

CLOUD USAGE

Currently the JINR cloud usage is developed in three directions:

1) test, educational and research tasks as part of participation in various projects using cloud and grid technologies;

2) systems and services deployment with high reliability and availability requirements;

3) extension of computing capacities of the grid-infrastructures.

The up to date list of services and testbeds currently deployed in JINR cloud is available online [10]. The JINR cloud is used not only by JINR users, developers, and system administrators to fulfil JINR commitments in different national and international projects but also as a computing facility in BESIII experiment. In order to let its computing infrastructure services interact with JINR cloud and run computing jobs there, the rOCCI-server was deployed on FN and configured to create VMs in our cloud on behalf of dedicated bes3 user within some computing resources share.

Moreover, a number of OpenNebula testbeds are deployed in the JINR cloud to develop and debug for current and new OpenNebula software releases all in-house implemented components such as OpenVZ and cloud bursting drivers, resources accounting and Kerberos authentication plugins. Each of such testbeds consists of 1 OpenVZ container as FN and 2–3 KVM VMs as CNs with OpenVZ kernel installed inside them.

Apart from that since the beginning of 2014 when JINR cloud had been launched, several training courses on OpenNebula cloud usage and administration were conducted at JINR upon users' or organizations from JINR Member States requests (see details online [11]).

FUTURE PLANS

To join resources of partner organizations for solving common tasks, as well as to distribute a peak load across them, a cloud bursting driver was developed by the JINR cloud team. It allows one to integrate the JINR cloud with partner clouds either OpenNebula-based one (and in that case it is possible to enable real time external cloud resources monitoring) or any other cloud platform which supports Open Cloud Computing Interface (OCCI), but then statically defined values of external cloud resources are only possible. The driver is based on ruby implementation of OCCI (rOCCI) and OpenNebula XML-RPC interface. The reason why both interfaces are used at the same time is that OCCI allows one to better reuse the code, since it is provided by other popular cloud platforms like OpenStack, while OpenNebula XML-RPC

is OpenNebula-specific and cannot be used for integration with other platforms. At the same time, OCCI does not provide monitoring operations, while OpenNebula XML-RPC does. It was decided to use rOCCI for management operations and XML-RPC for monitoring.

By the end of 2015, it is planned to integrate the JINR cloud with clouds of the following organizations: Institute of Physics of Azerbaijan National Academy of Sciences (Baku, Azerbaijan); Bogolyubov Institute for Theoretical Physics of the National Academy of Sciences of Ukraine (Kiev, Ukraine); Plekhanov Russian University of Economics — PRUE (Moscow, Russia). Apart from that it is planned to be integrated into EGI Federated cloud and provide cloud computing resources for ALICE experiment. Moreover, there is ongoing research work on cloud datastores usage in grid environments.

CONCLUSIONS

The JINR cloud service made it possible to provide a modern facility for experiments and various groups of users and allowed to better utilize hardware resources. It also significantly simplified the job of system administrators by automating many virtual machines and containers management tasks and by giving the users the ability to create and manage VMs and CTs by themselves within the limit of the granted quotas.

Open source code of OpenNebula platform as well as its flexibility and extensibility allowed one to fine-tune the JINR cloud service for local environment by developing in-house add-ons and drivers. The service is actively used to cover users' demands as well as to carry out JINR commitments in various local, national, and international scientific research projects.

Migration to HA-cloud setup and relocation cloud datastores to DNFS led to the increasing service reliability and availability.

In the nearest future it is planned to make the integration with private clouds of partner organizations and European cloud infrastructure.

Acknowledgements. Some work on the JINR cloud deployment and development was partially supported by the Russian Foundation for Basic Research (RFBR) grant #14-07-90405 "Cloud Computing Technologies Development and Integration into BITP and LIT JINR Tier-2 Grid Sites for ALICE Experiment Data Processing".

REFERENCES

1. *Baranov A. V. et al.* Cloud Infrastructure at JINR // Comp. Res. Modeling. 2015. V. 4, No. 3. P. 463–467.
2. KVM. http://www.linux-kvm.org/page/Main_Page.
3. Red Hat Cluster Suite: https://en.wikipedia.org/wiki/Red_Hat_cluster_suite.
4. rOCCI. https://github.com/EGI-FCTF/rOCCI-server.
5. OpenVZ. http://openvz.org.
6. Distributed Replication Block (DRDB). http://drbd.org.
7. Heartbeat. http://linux-ha.org/wiki/Heartbeat.
8. LizardFS. http://lizardfs.com.
9. Nagios. https://www.nagios.org.
10. https://miccom.jinr.ru/en/jinr-resources/jinr-cloud-service/#Cloud_service_utilization.
11. https://miccom.jinr.ru/en/jinr-resources/jinr-cloud-service/#Training.