



# JOINT INSTITUTE FOR NUCLEAR RESEARCH

The Laboratory of Information Technologies

## Analysis of methods for JINR cloud integration with grid infrastructures

JINR Summer Student Program project report

**Supervisors:**

N. Kutovskiy

N. Balashov

**Technical assistance:**

A. Baranov

**Student:**

E. Kotsareu, Belarus

Sukhoi State Technical University of Gomel

**Participation period:**

July 5 – Aug 27

Dubna, 2016

## **Contents:**

Introduction.....	1
Cloud resources integration methods into grid infrastructures .....	1
What is the lifecycle manager for? .....	2
VCondor (Virtual Condor) .....	3
VAC.....	4
VCycle .....	5
Work performed.....	6
References.....	8
Acknowledgments.....	9

## **Abstract**

This report covers common principles, software and architecture solutions for cloud resources integration into grid infrastructures. Document reviews each component highlighting its features, pros and cons, describing why it was selected to accomplish given project goal.

## **Introduction**

Nowadays one of the most important direction of cloud technologies development is a creation of methods for integration of various cloud infrastructures. Lack of own computing resources and a necessity to attract additional ones makes this topic very important in academic field. Another trend is an integration of cloud resources with grid infrastructures in order to increase computing facilities of the last ones.

The JINR participates in the NOvA experiment which uses Open Science Grid (OSG) infrastructure for data storage, processing and analysis. To fulfill JINR's commitments in NOvA project as well as in other ones it needs to contribute JINR storage and computing resources into experiments' infrastructures. Most OSG computational resources are integrated into that grid infrastructure via HTCondor CE and underlying HTCondor batch system. Thus there is a necessity to analyze a possible ways of cloud resources integration methods into grid infrastructures in general and JINR cloud facility into OSG infrastructure in particular.

## **Cloud resources integration methods into grid infrastructures**

There are several cloud management platforms (OpenStack, OpenNebula, Amazon EC2, RackSpace, CloudStack, etc). The JINR cloud is based on the OpenNebula one.

As it was mentioned above the NOvA experiment relies on the OSG infrastructure to store, process and analyze its data. To integrate computing resources running under particular Local Resources Management System (so called “LRMS”) into the OSG infrastructure two types of Computing elements (CEs) are used [1]: a HTCondor CE and a Globus GRAM CE. With help of some additional software components the HTCondor CE allows to integrate cloud resources as its computing back-end. Such software components are VCondor, VAC and VCycle. A short description, features and advantages of each of them are given below.

## **What is the lifecycle manager for?**

Lifecycle manager allows to deploy multiple virtual machines using pre-created OS images with all important software and system tweaks and settings already baked inside this image. Such approach lets an administrator to easily modify original system image once and deploy it to any number of VMs using it as a template. Physical server provides enough computing power to deploy multiple virtual machines as an additional level of abstraction which allows user to control (add, remove, start, stop) VMs independently from physical server. Considering flexible on-demand virtual computing power distribution between different virtual instances the lifecycle manager takes off VM and resource management responsibility from user.

# VCondor (Virtual Condor)

## Features:

- Use HTCondor as a new scheduler
- Push mode
  - VM joins cluster automatically on startup
- VM pool management
  - Access group: which group user belongs to.
  - Resource group: which resource is used by group.
  - Mapping table. [2]

## VCondor structure

### Vcondor structure

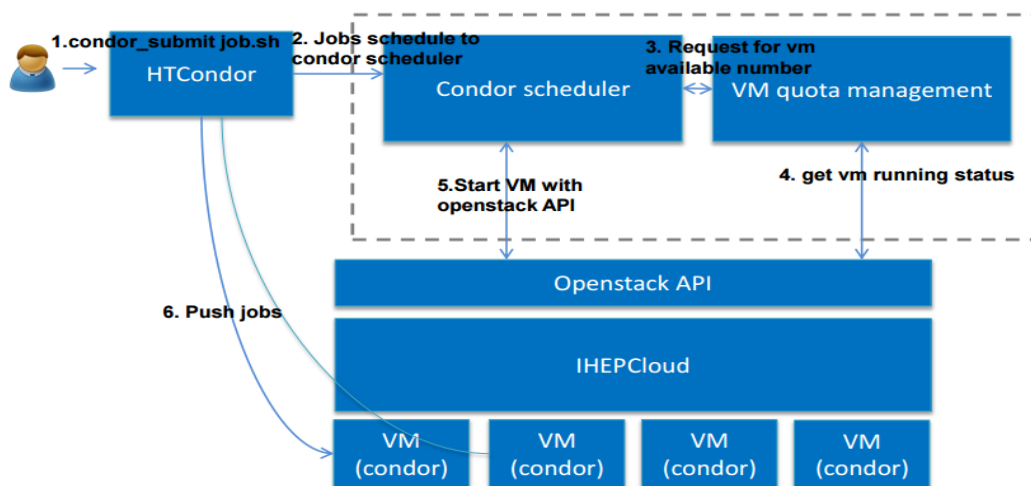


Figure 1: VCondor structure.

## Cons

- There is no support for OCCI API (under development)
- Lack of documentation (under preparation)

# VAC

## Features

- OpenStack-compatible environment to VMs using autonomous hypervisors (VM factories)
- Light-weight implementation: 3500 lines of Python vs 1,000,000 for OpenStack
- VacQuery UDP refactor. Protocol is used for inter-VM-factory communication to make it more scalable.
- System is more robust against even high (50%) packet loss levels
- Squid-on-factory configuration included in Puppet module
- Machine/Job Features updated for HSF-TN-2016-02. [3]

## VAC structure

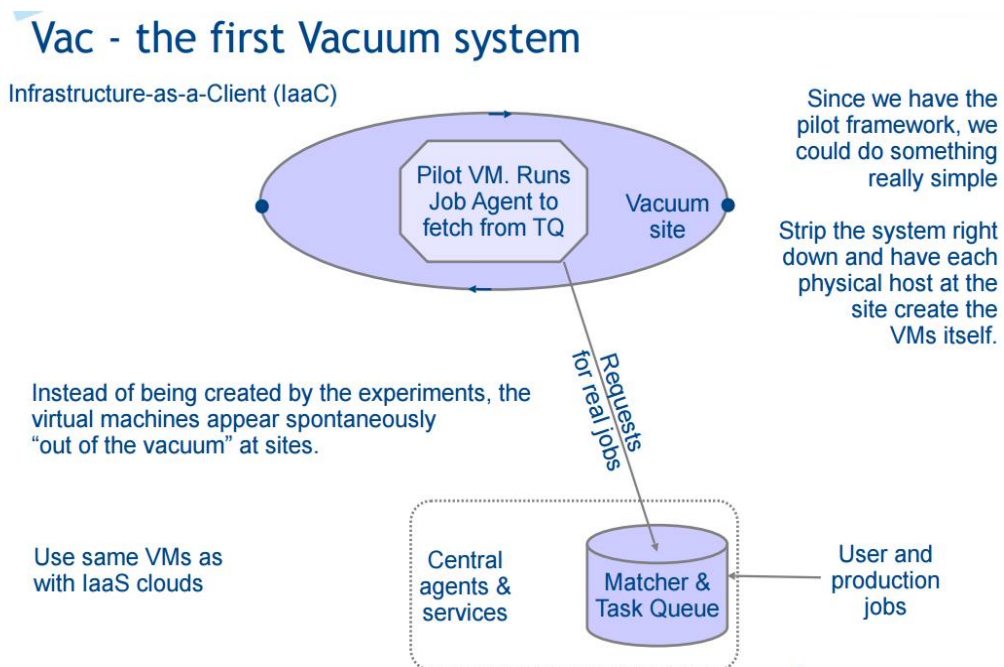


Figure 2: VAC structure.

## Cons

- Infrastructure-as-a-Client (IaaS)
- Vac - autonomous hypervisors

## • VCycle

### Features

- VMs created via Cloud API in response to observed demand for each type of VM
- Distributed as OpenStack plugin
- EC2 plugin in development
- OCCI (EGI), DBCE, and Azure (MS) plugins contributed by CERN
- Vcycle is used to manage LHCb OpenStack tenancy at CERN (500 VMs)
- Also LHCb tenancy at CC-IN2P3 and GridPP at Imperial
- Code and man pages good [3]

### VCycle structure

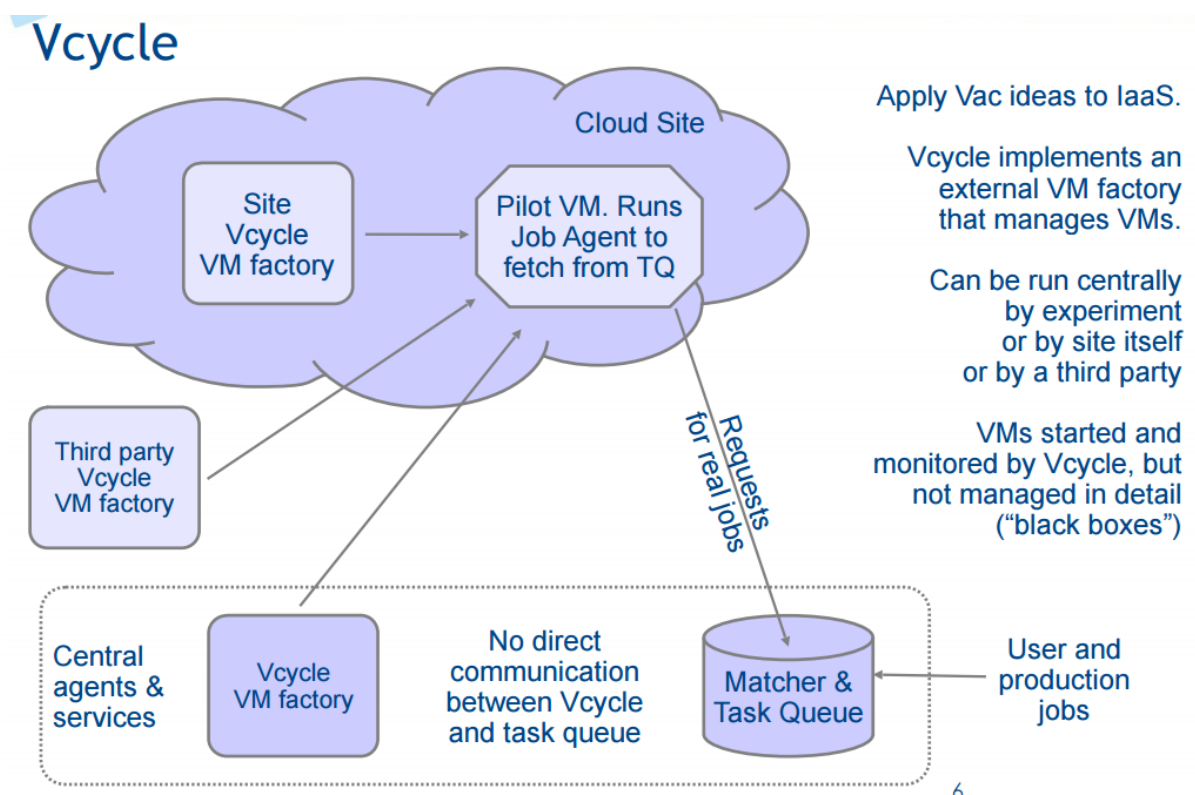


Figure 3: VCycle structure.

### Cons

- Poor admin documentation (man pages only) [4]

## Work performed

According to research made in a field of cloud computing, there are several possible ways of cloud resources integration into grid infrastructures: VAC, VCondor, VCycle. To make a decision it is important to examine documentation of every project and consider exact purposes it was made for:

1. **VAC** cannot be used due to differences in model principles (OpenNebula uses IaaS model while VAC uses IaaS model)
2. **VCondor** is also not a best choice in this case due to lack of documentation. It will be hard for newbie to set up everything without proper guides.
3. **VCycle** is a good choice considering that official page of VAC also advises users to use it in IaaS model-system.

To set up a simple OpenNebula testbed, it is vital to create at least 2 virtual machines: one if for cluster node (CN) and one is for front-end node (FN). Hardware configuration of each node are following:

- 1 KVM VM for OpenNebula CN
  - Dual-core CPU, 2GB RAM and 20 GB HDD
- 1 OpenVZ container (CT) for OpenNebula FN
  - Single-core CPU, 1 GB RAM and 20 GB HDD

The VCycle lifetime manager was deployed on a dedicated CT with following hardware resources:

- Single-core CPU with 1 GB RAM and 20 GB HDD



HTCondor requires 2 CT instances with following configuration:

- HTCondor FN
  - Single-core CPU,1 GB RAM and 20 GB HDD
- HTCondor CN
  - Single-core CPU,1 GB RAM and 20 GB HDD

VCycle interacts with OpenNebula via OCCI interface using OCCI-client which communicates with cloud by means of OCCI server linked with OpenNebula FN. One of the possible deployment scenario is to install the OCCI-client on the VCycle node and the OCCI-server – on the OpenNebula FN.

The main problem during such configuration deployment is lack of appropriate documentation on linking VCycle and OpenNebula with each other. VCycle was tested to be working with OpenStack cloud management platform and it's declared to support OCCI interface which means it should work with OpenNebula either. Because of lack of the documentation the VCycle developers were contacted for a help with connecting OpenNebula and VCycle. Contacted person answered that VCycle was only tested with OpenStack but not against OpenNebula. So for the time being neither documentation nor VCycle developers were able to help with connecting VCycle with OpenNebula.

## References

[1] <https://twiki.opensciencegrid.org/bin/view/Documentation/Release3/WebHome>

[2] <https://indico.cern.ch/event/466991/contributions/1143599/>

[3]

<https://indico.cern.ch/event/477023/contributions/1155276/attachments/1268241/1878275/20160412-mcnab-vac-vcycle.pdf>

[4]

<https://indico.cern.ch/event/304944/contributions/1672217/attachments/578470/796600/mcnab-vac-vcycle-13apr15.pdf>

## **Acknowledgments**

I am immensely grateful to my supervisors N. Kutovskiy, N. Balashov and A. Baranov for their assistance in performing of this work, motivation, and immense knowledge. Also I would like to thank the University Center of the Joint Institute for Nuclear Research (JINR) for giving me the possibility to practice in the Laboratory of Information Technologies.

At last, I would like to express my gratitude to the Management of the Laboratory of Information Technologies for the financial support of my summer practice, for new great experience and excellent working conditions.