




Comparative Study of the Support of Academic Clouds Based on Apache CloudStack and Proxmox VE Platforms

Vasyl P. Oleksiuk^{1,2}^a, Olesia O. Oleksiuk³^b and Oleg Spirin^{2,4}^c

¹*Ternopil Volodymyr Hnatiuk National Pedagogical University, 2 M. Kryvonosa Str., Ternopil, Ukraine*

²*Institute for Digitalisation of Education of the National Academy of Educational Sciences of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine*

³*Ternopil Regional Municipal Institute of Postgraduate Education, 1 V. Hromnytskogo St., Ternopil, 46027, Ukraine*

⁴*University Of Educational Management of NAES of Ukraine, 52 A, Sichovykh Striltsiv Street, Kyiv, Ukraine
oleksyuk@fizmat.tnpu.edu.ua, o.oleksyuk@ippo.edu.te.ua, oleg.spirin@gmail.com*

Keywords: cloud computing, academic cloud, Apache CloudStack, Proxmox VE, Google Workspace

Abstract: Cloud technologies provide users with efficient and secure tools for data management, computing, storage and other services. The article analyzes the projects for the introduction of cloud technologies in education and identifies the main advantages and risks in creating a cloud infrastructure for the university. Such startups contribute to the formation of a new paradigm of education. It involves the virtualization of education, the introduction of mobile and blended learning, ie the combination of cloud computing with modern learning concepts. In this paper, we compare some aspects of deployment and support in our experience in improving the academic cloud for the training of a bachelor's degree in computer science. This is through the integration of the Proxmox VE platform into existing computing power by deploying the Proxmox VE system. In the study, we reveal some technical and methodological aspects of the organization of the educational process using this corporate cloud platform. The scheme of the organization of physical components of cloud infrastructure (nodes, virtual networks, routers, domain controller, VPN-server, backup system of students' virtual machines) is given. All characteristics of this environment and possibilities of their application are studied.


1 INTRODUCTION


Today, many universities are creating their own cloud-based learning environments (CBLE). Although there is currently no single concept for CBLE, scientists understand it as similar concepts (Glazunova and Shyshkina, 2018), (Bykov et al., 2020), (Korotun et al., 2019), (Spirin et al., 2022). In general, it can be understood as an IT system consisting of cloud services and providing learning mobility, group collaboration of teachers and students to achieve educational goals (Lytvynova, 2017).


As the analysis of the literature shows, many university CBLE are usually deployed according to the hybrid model (Shyshkina, 2016), (Markova et al., 2018), (Vorozhbyt, 2019). One of the most important components in the structure of this environment is the private academic cloud (Glazunova, 2015). It

is now deployed according to the most productive IaaS service model. A hybrid cloud is a cost-effective way to solve the problem of insufficient computing resources. The private academic cloud allows the university to meet the peak demand of students and faculty through a combination of local infrastructure and one or more public clouds (Wang et al., 2020a).

Various commercial and free platforms are used in universities to build private academic clouds (Wang et al., 2020b), (AL-Mukhtar and Mardan, 2014), (Ilin et al., 2016), (Amiel et al., 2020). A productive method of deploying private academic clouds is to use solutions from leading cloud vendors such as Google Inc., Microsoft, Amazon and others. Google Inc. offers researchers, universities, faculties, faculty and students grants and loans for teaching and research. In particular, leading European educational institutions can access Google Cloud within the Internet2 project (GCP, 2020). Unfortunately, these opportunities are not currently available for our country (Ukraine). For example, Microsoft Educator Grant is a program designed specifically to provide access to

^a  <https://orcid.org/0000-0003-0789-0272>

^b  <https://orcid.org/0000-0001-5659-4774>

^c  <https://orcid.org/0000-0002-9594-6602>

Microsoft Azure to college and university professors teaching advanced courses. As part of this program, faculty teaching Azure in their curricula are awarded subscriptions to support their course (SDR, 2020). In general, these programs are very useful and productive. However, they are usually provided on a temporary basis and therefore cannot completely replace the cloud-based IT infrastructure of universities and colleges.

Among the free platforms for cloud infrastructure deployment, CloudStack, Openstack, Proxmox VE, and Eucalyptus are the most suitable. Each of them has its advantages and disadvantages. There are many attempts to compare these platforms. The authors of articles comparing such platforms state (SDR, 2020), (Fylaktopoulos et al., 2016):

- OpenStack has large community, offers wide integration with storage, network and compute technologies, but is too complex to deploy and configure;
- Eucalyptus the longest-standing open source project is banking on its very tight technical ties to Amazon Web Services (AWS). The platform is configurable, but not very customizable;
- Proxmox is open source platform. It can provide easy way to deployment cloud infrastructure. But it is not very suitable as a platform for a private cloud in the CLBE;
- CloudStack has well rounded GUI, can provide an advanced cloud infrastructure, but it is very GUI centric built on single Java core.

We have deployed a private academic cloud based on the Apache CloudStack platform for the last 8 years. We have been deploying a private academic cloud based on the Apache CloudStack platform for the past 8 years. For the last 2 years we have been deploying an academic cloud based on Proxmox VE as well. Apache CloudStack deployment is housed in the infrastructure building. It contains a management server, 4 hosts, 4 primaries and 1 secondary storage. We decided to use hypervisors instead of containers. This is because the former are more versatile. In addition, the use of hypervisors is safer than containers. To save computing resources, we have installed primary storage on the hosts. We used VLANs to distribute traffic across individual networks. These networks can be allocated to groups or individual students.

We have so far rented one refurbished physical server based on a used Intel Xeon processor with 32 Gb of RAM for deployment of Proxmox VE. This was also done because Ukraine is currently at war and it

is difficult to say whether servers will be available as they are in university buildings.

In general, our private academic cloud provides (Fylaktopoulos et al., 2016):

- Development and Execution of student virtual machines;
- Aggregation of computing resources of several hosts;
- VM migration between repositories;
- VM connections to each other through guest networks;
- Launching VMs within other VMs;
- Integration with Active Directory;
- Distribution of student accounts according to their academic groups.

There are many advantages and disadvantages to using Apache CloudStack and Proxmox VE as an academic cloud platforms. There are some problems happens in the process private academic cloud using.

The purpose of the article is to compare some tasks in maintenance of academic clouds based on Apache CloudStack and Proxmox VE platforms.

The following tasks are required to achieve the goal of the research:

1. Analysis of maintenance tasks of academic clouds in foreign and Ukrainian universities.
2. Definition of maintenance tasks of academic clouds based on Apache CloudStack and Proxmox VE platforms.
3. Describe and compare methods for solving these tasks.

We used a set of research methods such as theoretical (analysis of scientific, technical literature, experience; generalization of experience of using cloud computing in education) and empirical (modelling, designing, developing of scripts).

2 THE PRIVATE CLOUDS MAINTENANCE TASKS

As the experience of cloud infrastructure maintenance shows, this is an ongoing process. It requires constant attention from engineers, network system administrators, teachers, and student involvement. Scientists describe the experience of deploying an academic private cloud, including determining the performance of hypervisors and storage (Spirin et al., 2018). The biggest challenge for researchers was the transition from a prototype of an academic cloud to

a productive one. In this context, they addressed the problem of load balancing, elastic hypervisors, security threats. Storage backup tasks are also important for such clouds.

The authors of the book "Data Backups and Cloud Computing" offer the concept of backup to cloud storage. They say that both the cloud provider and the cloud consumers have to take comprehensive steps to ensure appropriate configurations, hardening of the CBLE, appropriate design and development, appropriate interoperability, and adequate testing (Khmelevsky and Voytenko, 2015).

Scientists from the Institute of Physics and Mechanics of the National Academy of Sciences of Ukraine and Lviv Polytechnic National University have developed an effective method of deduplication and distribution of data in cloud storage during the creation of backups. Researchers have developed an intelligent system for such deduplication and tested it (Rao and Nayak, 2014).

Junfeng Tian, Zilong Wang and Zhen Li also studied cloud data backup. The authors propose a scheme for data separation and backup and encryption. They state that their own scheme resolves the conflict between data security and the survivability of the IT infrastructure with the help of encrypted backup (Rusyn et al., 2019).

The Apache CloudStack cloud infrastructure redundancy model developed by Paul Angus is very useful for our study. It creates a vendor agnostic API and UI in CloudStack for end users. The author's Framework abstracts the specifics of solutions, such that through the use of a plugin, a 3rd party solution can deliver backup and recovery solutions (Rusyn et al., 2019).

Here are some tasks for servicing our sample private academic cloud:

1. Making changes to the cloud infrastructure.
2. Creating and verification of the academic cloud performance model.
3. Designing and realization an academic clouds' backup model.

Let's consider these tasks in more detail.

3 Making changes to the cloud infrastructure

Maintenance of our developed cloud infrastructure involves the implementation of tasks such as:

- changing the parameters of the components of the cloud infrastructure - zones, clusters, hosts, storages;

- creating and routing of virtual networks for individual groups or students;
- creating and modifying templates of compute offering services that determine the performance of VM;
- creating and modifying network offering service templates such as VPN, DHCP, DNS servers, Firewall, Load Balancer and others;
- creating projects for VM sharing by students.

This task also involves creating student and faculty accounts. We authenticate users of the academic cloud from a centralized database - LDAP-directory (Microsoft Active Directory). This approach makes it possible to use single registration data to access all hybrid IT infrastructure services. We used CloudStack domains to distribute students according to academic groups. Adding users to them is possible in automatic (using links at the first successful authentication) and manual mode. Unfortunately, due to the incompatibility of our users' logins with the Apache CloudStack platform, we had to choose the manual mode. To reduce the technical work involved in finding LDAP directory entries, we have created several queries to filter user account data.

Proxmox VE also provides authentication based on the LDAP directory (Active Directory). To do this, cloud administrator need to specify some parameters such as socket to connect to the domain controller, user data for reading data from the directory, some user attributes, filters to restrict access.

To log in to the system, the administrator can add it manually, for example using the web interface. However, it is possible to automatically synchronize the directory with the `pveum realm sync ;realm;` command. Like Apache CloudStack domains, Proxmox VE can create groups. For these groups, cloud administrator can define privileges for access to cloud infrastructure.

Apache CloudStack users automatically have access to their own VMs. This is good for students to complete individual tasks. The creation and providing permission to VMs isn't a trivial task in the Proxmox VE Platform. We offer two approaches to do this. The first involves the administrator creating copies of VMs and giving students access to these VMs. Permissions may vary depending on the operations to be performed with these VMs. For example, cloud administrators can provide students with administrative access to such VMs. Another approach involves students cloning their own machines from the templates. It requires providing some rights. They are displayed as a table 1.

Creating permissions that match the last row of the

Table 1: Permissions are required to clone student VMs from templates

| Permissions | Path | Propagate |
|------------------|------------|-----------|
| PVEAuditor | / | + |
| PVEDatastoreUser | /storage | + |
| PVETemplateUser | /pool | + |
| PVEVMUser | /nodes | + |
| Administrator | /vms/VM_ID | - |

table requires defining *students_login - VM_ID* pairs. Finally, permission is granted through the command: *pveum acl modify /vms/VM_ID user username -role Administrator*

Later, using a script, we obtained a list of users from the LDAP directory and set the appropriate IDs for them. It is important that when cloning VM, students must indicate the ID provided by the teacher for VM.

3.1 The academic cloud performance model

Performance calculations are needed to understand the number of VMs that can run in the cloud. It is also needed to understand the computing performance of these machines, which is required to create their templates.

Both platforms use CPU and memory redundancy depending on the number of VMs. Proxmox allows cloud administrators to do this more flexibly when running containers instead of VMs.

When creating the service offering templates, we compared the characteristics of the hardware hosts (CPU frequency, RAM) with the minimum guest OS requirements and the number of students. To do this, we used the inequality:

$$FRQ = N_{st} * F_{OS} < FRQ_{hosts},$$

where FRQ – the total frequency of VMs processors; N_{st} – amount of students; F_{OS} – the minimum frequency is recommended for the guest OS; FRQ_{hosts} – total frequency of hardware host processors. The last value can be found from the ratio:

$$FRQ_{hosts} = \sum_{i=1}^n (N_{ci} F_{ci}),$$

where N_{ci} – the number of cores in the processor of the i -th host, F_{ci} – CPU frequency of the i -th host. Unlike Apache Cloudstack, Proxmox does not have performance templates. Appropriate parameters must be set for each VM during its installation. If a student clones a VM from a template, he/she cannot change its performance settings. Instead, when using the VM,

Table 2: Main Characteristics of Academic Cloud's Hardware

| N | N_{ci} | F_{ci} | MEM_{ci} | FRQ_{hosts} |
|----------------|----------|----------|------------|---------------|
| Host0 | 4 | 3200 | 16384 | 12800 |
| Host1 | 4 | 3100 | 24576 | 12400 |
| Host2 | 4 | 3100 | 16384 | 12400 |
| Host3 | 4 | 3700 | 32768 | 14800 |
| All_CloudStack | 16 | | 90112 | 52400 |
| Proxmox | 8 | 3500 | 28000 | 31980 |

these parameters are available for change. However, the last two formulas are also valid for the Proxmox platform. In order to compare the computing capabilities of the studied cloud platforms, we will consider the hardware characteristics of the respective servers (see table 2).

As table 2 shows, the private academic cloud based on Apache CloudStack has a total frequency of about 50GHz. And the total amount of memory is about 90 Gb. Unfortunately, we do not have the equipment to deploy identical academic clouds based on the CloudStack and Proxmox VE platforms. However, we can assume that the performance of the physical server with Proxmox VE should be equal to the computing power of the two servers with CloudStack (Host0 and Host2). Therefore, in the CloudStack deployment, we disabled Host1 and Host3. During the testing, Host1 stayed online as a management server in the CloudStack infrastructure. But it didn't run any VMs.

Regarding the frequency, two other opposite factors should be taken into account:

- table 2 shows the base frequency, and processors can run faster thanks to Turbo-Boost technology;
- hosts run other software (OS, databases, management servers, hypervisors). It also consumes resources.

It is well known that the frequency of a modern processor is not constant. It can increase or decrease depending on the mode of operation of the CPU. That's why we use Processor Base Frequency in the tables and formulas above. Processor Base Frequency describes the rate at which the processor's transistors are open and close. The KVM hypervisor measures this processor frequency. It is software for launching and executing virtual machines in the Apache CloudStack and Proxmox platforms.

Similarly, to determine the required amount of memory we used the inequality:

$$MEM = N_{st} * MEM_{OS} < MEM_{hosts}$$

Table 3: Some templates types of our academic Cloud’s performance

| Template | F_{OS} | MEM_{OS} | N_{st} | FRQ | MEM |
|------------|----------|------------|----------|-------|-------|
| LinuxCLI | 500 | 500 | 20 | 10000 | 10000 |
| LinuxGUI | 1500 | 2000 | 20 | 30000 | 40000 |
| WindowsWs | 1000 | 2000 | 20 | 20000 | 40000 |
| WindowsSrv | 1500 | 2000 | 20 | 30000 | 40000 |
| AdvLinux | 2500 | 4000 | 20 | 50000 | 80000 |

Table 4: Comparison of the number of VMs running on Apache CloudStack and Proxmox VE platforms

| Template | CloudStack | Proxmox VE |
|------------|------------|------------|
| LinuxCLI* | 50 | 120 |
| LinuxCLI | 112 | 120 |
| LinuxGUI | 18 | 19 |
| WindowsWs | 20 | 22 |
| WindowsSrv | 16 | 16 |
| AdvLinux | 12 | 8 |

3.2 Verification of the academic cloud performance model

To check computation performance of both clouds we loaded the same VMs on two clouds until their CPU load was 90%. There were several types of test virtual machines (see table 3). These types corresponded to the VM templates that were created for the various training courses. For example, templates LinuxGUI, WindowsWs were used in the course of operating systems, AdvLinux in the course of computer networks, WindowsSrv in the course of administration. The Linux OS with CLI worked as Proxmox VE containers and as regular VMs in CloudStack. We use the EVE-NG platform based on AdvLinux template for modelling in the study of computer networks. It launches its own VMs inside the main Apache CloudStack VM (Spirin et al., 2019). Such nested virtualization requires more resources.

The number of VMs that were loaded is shown in table 4.

Comparing the data in table 2 and table 3, we can conclude that our deployments of both platforms provides about 50 VMs with Linux with command line interface (CLI), more than 40 VMs with Windows Workstation, about 35 VMs with Windows Server and OC Linux with GUI (see table 4). This test showed that the number of virtual machines loaded on Proxmox VE is slightly higher than in Apache CloudStack (except for the template AdvLinux).

To verify the performance model, we performed academic cloud testing. The experiment was performed according to the method described in the article (Algarni et al., 2018). The authors’ model used traditional benchmarking systems to measure

system performance and individual components like CPU, memory, cache and disk performance. Like the previous case, we compared the performance of two CloudStack hosts and one Proxmox VE. To test performance of clouds we have created 16 identical VMs (8 on Apache CloudStack and 8 on Proxmox VE). They had equivalent 2-core processors, the same amount of RAM (2 Gb) and the size of disks (14 Gb). The same Ubuntu server 20.04 x64 was installed on all these VMs.

We have performed real-tasks tests such as:

- Ab is a tool for benchmarking Apache web server. Ab was used for measuring of time taken for the system to respond to a request from a web client. This can estimate the response efficiency of each VM by measure the number of static web page requests a server can fulfil in one second.
- The LZMA benchmark measures the amount of time consumed in compressing a file using its own compression algorithm.
- John the Ripper is a tool for measure CPU throughput. In this benchmark, the efficiency of the process management of the hypervisors is evaluated. This benchmark is a common decipher application which uses diverse ciphers such as DES and MD5.
- IOzone benchmark is used to evaluate performance of file system. IOzone generates and assesses many file operations such as read, write and random read.

Ab was used for measuring of time taken for the system to respond to a request from a web client. This can estimate the response efficiency of each VM. Our script have completed 100000 requests with concurrency level 1000 simultaneous requests. The results of all test available by the URL <https://drive.google.com/file/d/1m-3oS7cOWyCFUveBe8t4EnHOcll6ApyX>. Shared table contains the designations of VMC for VMs Cloudstack and VMP for Proxmox VE ones. The average values of ab-test are given in the table 5

Table 5: Ab benchmark results

| Platform | Requests per sec | Transfer rate, Kb/s |
|------------|------------------|---------------------|
| CloudStack | 2115,68 | 95150,89 |
| Proxmox | 1787,62 | 80437,84 |

In the Apache benchmark, CloudStack performance is 18% higher in terms of both the number of requests and transfer rate. A slightly smaller advantage of Apache CloudStack (near 7%) was shown by the LZMA test. We performed John the Ripper benchmarks for the case of coding according to DES 128

and MD5 algorithms. Because the test was run on a single CPU core, the "real" and "virtual" results were the same. We analyzed the results for the single-salt case. They indicate (see table 6) that Apache CloudStack has significantly advantage under Proxmox VE (38% and 54% for DES 128 and MD5 algorithms).

Table 6: John the Ripper benchmark results

| Platform | MD5 hash | DES 128 hash |
|------------|----------|--------------|
| CloudStack | 14526,13 | 61055,88 |
| Proxmox | 10503,75 | 39524,38 |

Similar results were obtained in the performance test of virtual HDDs (see fig. 1)

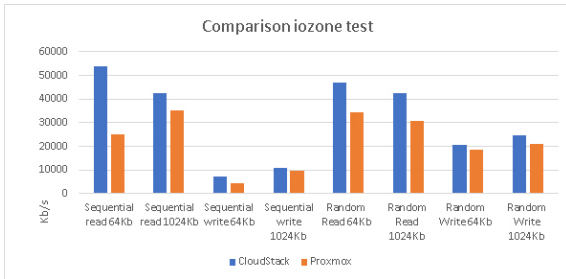


Figure 1: Comparison of the results of iозone test

Therefore, it is reasonable to predict that the performance of the virtual machines themselves will be higher on the Cloudstack platform. It is hoped that students will feel this in their academic research work. However, another important factor in this assessment is the functionality and usability of the interface. The Proxmox VE console interface is more advanced than Apache Cloudstack. This fact, together with a good mobile version of the platform allows you to work with the cloud from different devices.

The previous test concerned the performance of the virtual machines themselves. We decided to investigate the performance of hardware servers (hosts) with their running hypervisors. This test will allow us to evaluate the efficiency of the use of computing resources by cloud platforms. To do this we used the formula to rank hypervisors' performance proposed by P.Vijaya Vardhan Reddy and K. Shyamala (Reddy and Shyamala, 2016). According to it, the hypervisor's score scores of its CPU and memory.

$$Score_{HSI} = Score_{HCPU} + Score_{HMEM}$$

Each of these scores is calculated as the sum of the CPU and RAM performance of any VM in the cloud.

$$Score_{HCPU} = \sum_{i=1}^n CP_{VMi}$$

$$Score_{MEM} = \sum_{i=1}^n MP_{VMi}$$

The CP_{VMi} and MP_{VMi} values are calculated as follows

$$CP_{VMi} = \alpha_{CPU} \times \frac{ACPU_{VMi}}{HCP_{VMs}}$$

and

$$MP_{VMi} = \alpha_{MEM} \times \frac{AM_{VMi}}{HMP_{VMs}},$$

where α_{CPU} and α_{MEM} are CPU and memory weight factors.

$ACPU_{VMi}$ and AM_{VMi} are available CPU and memory resources of VMi;

HCP_{VMs} and HMP_{VMs} are all VMs relative CPU(RAM) utilization with respect to host.

The formulas for calculating the values HCP_{VMs} and HMP_{VMs} are as follows

$$HCP_{VMs} = \frac{TotalCPU - \sum_{i=1}^n ACPU_{VMi}}{TotalCPU} \times 100,$$

$$HMP_{VMs} = \frac{TotalMEM - \sum_{i=1}^n AMEM_{VMi}}{TotalMemory} \times 100,$$

where $ACPU_{VMi}$ and $AMEM_{VMi}$ are available CPU(RAM) of VMi.

Alpha coefficients are the ratio of processor resources (memory) of one VM to all VMs. Since we have all 8 machines the same, for each platform $\alpha_{CPU} = \frac{1}{8}$ and $\alpha_{MEM} = \frac{1}{8}$.

We applied the above formulas and obtained the following values for the studied platforms (see table 7)

Table 7: The obtained scores of Cloudstack and Proxmox platforms

| | AB | Izma | iозone |
|------------------|-------|-------|--------|
| Score_Cloudstack | 77,24 | 68,55 | 349,29 |
| Score_Proxmox | 79,03 | 87,18 | 284,85 |

Thus, we can state that the score of both platforms based on the Apache benchmark test is almost the same (the difference is about 1.5%). Proxmox VE showed the best results in the lzma test (about 27%). The opposite result of the advantage of Apache Cloudstack is observed in the iозone test. However,

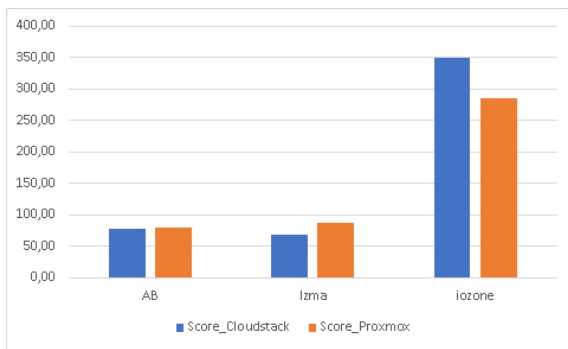


Figure 2: Comparison of the Apache CloudStack and Proxmox VE scores

we have doubts about this case, because the disk subsystems of the two platforms are significantly different (Proxmox VE uses a software raid, and Cloudstack generally contains two separate physical hosts with HDDs). Diagrams of the obtained ratings are shown in the figure 2.

So, based on the verification of the cloud platform performance model, we can say that the Apache Cloudstack and Proxmox VE have roughly equal ratings. In naked academic clouds, the performance of 2 physical Apache Cloudstack hosts is approximately equal to the performance of one Proxmox host. In this case, other factors of the cloud platform at the university will be decisive.

4 Designing and realization an academic clouds' backup model

The backup model depends on which platform will be stored. There are general backup principles that are valid for both platforms. However, the technical implementation will be different. We had to develop scripts for the Apache Cloudstack. Proxmox VE has powerful built-in backup tools.

Experience shows that the task of backup is very important and time consuming. This is primarily due to the large amounts of student VMs data in the private academic cloud. Large companies develop a disaster recovery plan in this case. But in educational institutions, IT services work to perform such tasks. Therefore, they need to develop a model, identify potential risks in the IT infrastructure, consider and implement an appropriate backup system.

The development of a backup strategy requires the definition of the main goals and objectives of the backup, tools and regulations. In general, the problem of back-up is relevant for almost all IT infrastructures. When choosing a backup method, the following criteria are important (Angus, 2020):

- backup time to the storage;
- recovery time from backup;
- the number of copies that can be stored;
- risks due to inconsistency of backups, imperfection of the backup method, complete or partial loss of backups;
- overhead costs: the level of load on the servers when performing copying, reducing the speed of service response, etc;
- the cost of renting all services and storage.

Currently, there are 3 main backup schemes such as:

- *Full*. This type of backup creates a complete copy of all data.
- *Incremental*. In this case, only files that have changed since the previous backup are copied. The following incremental backup only adds files that have been modified since the previous backup.
- *Differential*. The backup program copies each file that has changed since the last full backup. Differential copying speeds up the recovery process (Paulsen, 2011).

To save material costs, we use almost no server equipment and powerful and high-speed network storage in our academic cloud installation. Instead, we decided to use cloud services. For example, the Google Drive service within the G Suite for education package offers virtually unlimited disk space (Oleksiuk et al., 2017). The disadvantage of such a repository is the significant time to upload or download backups. This speed will be limited by the bandwidth of the university's Internet channel. The latter requirement can be considered acceptable, as our implementation of the academic cloud is used primarily for training rather than for production.

To use Google Drive in our own scripts, we need to use the API of this service. This interface is accessible through Google Developers Console, a software developer service. First you need to create your own project. Credentials were created to access this project. We have chosen to access OAuth 2.0 accounts. OAuth is an open authorization standard that allows a user or application to give and access data without having to enter a login and password. Access tokens are used for this purpose. Each access token provides access to a specific client to specific resources and for a specified period of time (Oleksiuk et al., 2017). After adding a new project, we created new data for authentication, selected the type of application (desktop) and activated the appropriate API (Google Drive API).

Because templates and ISO images in the CloudStack infrastructure do not change, but only new ones are added, we chose the incremental method to back up the secondary storage. Its implementation was based on the use of a ready-made utility for synchronizing storage files. Unfortunately, there is currently almost no such high-quality utility like Google backup and Sync, which is developed for OC Windows. We analysed several tools such as:

- Gdrive (grive2). Google Drive client with the support for the new Drive REST API and partial sync. It can't provide continuously wait for changes in file system or in Google Drive to occur and upload.
- Gnome-online-accounts. It is system utility located within system's settings in Gnome GUI. But it can only be executed in a graphical interface.
- GoSync is a Google Drive client with GUI support for Linux. It is designed under the GNU General Public License. The client is not perfect enough, for example, it has automatic regular synchronization every 10 minutes.
- Google-drive-ocamlfuse is a FUSE (Filesystem in Userspace) filesystem for Google Drive, written in OCaml. FUSE is a free module for the kernel of Unix-like operating systems. It allows developers to create new types of file systems available for users to mount without the root privileges of Google Drive on Linux (Yallop et al., 2018).

We used the latest utility. Here are its main features:

- full read/write access to ordinary files and folders;
- read-only access to google docs, sheets, and slides;
- multiple account support;
- duplicate file handling;
- access to trash;
- storing Unix permissions and ownership;
- support symbolic links;
- streaming through read-ahead buffers.

Some problem was that the utility requires authorization using a browser in a graphical interface. Therefore, we used an alternative authorization mode. Since we already had our own OAuth2 client ID and client secret, we specified them in the command:

```
google-drive-ocamlfuse -id
12345678.apps.googleusercontent.com -secret
abcde12345
```

As the command tries to start the browser on the server where there is no GUI we formed the necessary

URL as it is written in the documentation on Google Developers Console. After going to this address, we received a verification code. This code gave access to folder synchronization to the Google Drive.

For security reasons, we decided to sync not the Apache Cloudstack secondary storage itself, but a copy of it from the backup drive. So, we first synchronized local folders with the command:

```
rsync -azvh /export/secondary /export/sync_secondary/arch_cloud
```

where */export/secondary* – the secondary storage of Apache CloudStack infrastructure;

/export/sync_secondary/arch_cloud – the local copy of this storage.

To synchronize the */export/sync_secondary/arch_cloud* folder, the following command has been added to the server task scheduler: *google-drive-ocamlfuse /export/sync`secondary*

It runs every time a server with secondary storage is loaded.

A backup of all databases is required to restore the Apache CloudStack cloud infrastructure. These are such databases:

- *Cloud*. It contains all objects of cloud infrastructure.
- *Cloud`usage*. A database that contains generalized data on resource consumption by the end user. It is used to obtain statistics and compile reports.

Since the backup of these databases is quite small, we decided to store all backups in the cloud storage (Backup.Database task, see Figure ??). The traditional database for the Apache CloudStack platform is MySQL. The main utility for backing up MySQL databases is mysqldump. Its syntax involves entering a login name and password. Because the shell script in Linux is written as a plain-text file, it will contain the name of the user's password (usually the root) of the database. This is a potential security risk for the entire server. In order not to leave open the data for authorization of the database user, we used the "login path" option. A "login path" is an option group containing options that specify which MySQL server to connect to and which account to authenticate as. To create or modify a login path file, we have used the mysql_config editor utility. In general, the commands for creating and archiving a database dump are as follows:

```
/usr/bin/mysqldump -login-path=DailyBackup -u root -A
$BACKUP`DIR`"cloud`all`""`$date`daily`""`.sql"
```

The variable *\$date`daily* contains the current date of the archive. This allows you to see the date of

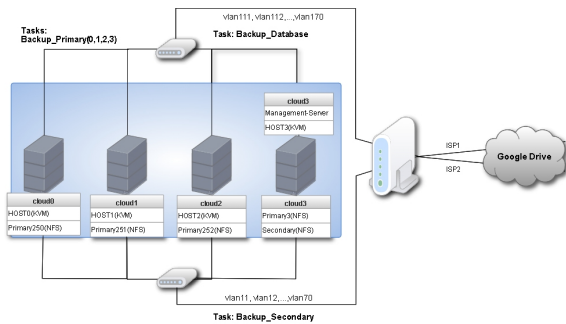


Figure 3: Apache Cloudstack infrastructure backup scheme

archiving directly in the file name.

```
tar -czf $BACKUP_DIR/"archive_cloud_all_$
date_daily.sql.tgz"
$BACKUP_DIR/"archive_cloud_all$date_daily.sql"
```

To upload the files to the server, we used a ready-made script from github (?). Here is its launch:

```
upload.sh "arch_cloud/DB" "$entry" $supl_file
folder_ID "application/x-gzip"
```

where

- *arch_cloud/DB* – folder for uploading files;
- *\$entry* – full path to the file;
- *\$supl_file* – file name to download;
- *folder_ID* – Google Drive folder ID;
- *application/x-gzip* – file MIME-type.

A special refresh token is required to provide long-term access of the upload.sh script to Google Drive. It can be obtained by curl-calling a URL such as:

```
curl --silent "https://accounts.google.com/o/oauth2
/token" --data "code=access_token &
client_id=client_ID &client_secret=client_secret
&redirect_uri=urn:ietf:wg:oauth:2.0:oob&
grant_type=authorization_code"
```

In general, the scheme of backuping of Apache Cloudstack infrastructure is shown in figure ??.

Performing backup of primary repositories (Backup Primary task (cloud0, cloud1, cloud2, cloud3)) has some difficulties. An analysis of Internet sources, management server databases, and storages files showed that the Apache CloudStack platform does not typically use full copies of disk templates for each VM. This means that full backups should be made to reduce the risk of inconsistencies in primary repository archives.

Additionally, it is necessary to prepare a cloud platform, stopping all VMs. Of course, students need to form an understanding of the need to turn off their own VMs. However, in practice this is not always

possible. Therefore, it is necessary to stop all VM programmatically, by means of a script. This can be done using the API features of the Apache Cloud-Stack platform. Using API functions allows the developer to access data about cloud infrastructure objects. It is also possible to change the state of these objects. To generate a query that contains API functions, you must specify:

- URL of the management server;
- Service construct "api?". It contains the path to a certain API-function, and indicates the beginning of the parameters that are transmitted using the GET method.
- Command. It is the name of the API-function.
- ApiKey. The key, that can be generated for each user account.
- Additional query options separated like GET queries using the "&" character.
- Response format (JSON or XML).
- Signature of the request.

Regardless of the protocol (HTTP or HTTPS) used to access the Apache Cloud-Stack API functions, the request must be signed. This allows the platform to confirm that the request was sent from a trusted accounting request that has the authority to execute the appropriate command. To sign a request, the developer must have an API key and an account secret key. They are generated by the platform administrator (OAUTH2, 2020).

Here is our bash-script to stop all working users' VMs.

```
mysql --login-path=DailyBackup -D cloud -e
"SELECT uuid FROM vm_instance WHERE type =
"User" and state = "running";" > uuid.txt
sed -i '1d' uuid.txt
while read LINE; do php -q cloudstackapi.php
"$LINE" ; done > uuid.txt
```

In the first line we receive in a file from a database the list of user VM with a running state. The next command clears the first line because it does not contain a VM. The third line runs the cloudstackapi.php script. It generates a signature and calls the stopVirtualMachine API.

Another way to back up the current state of the VM is to create their snapshots. The Apache Cloud-Stack platform provides 2 types of images (Rusyn et al., 2019):

- VM Snapshot – a hypervisor-driven point-in-time image of a virtual machine's disks. The exact mechanism of this is dependent on the hypervisor.

- Volume snapshot – a point-in-time image of a specific volume. The process usually involves taking a VM snapshot and then copying the required volume to secondary storage and the deleting the VM snapshot.

This approach requires additional space on the secondary storage or data coping on the user’s local disk. Such images can be taken by students from the web interface of the Apache CloudStack platform. Performing this action and turning off their own VMs after the end of their use are important components of ICT competence of the student.

However, experience shows that not all students perform these actions. Therefore, these are also worth automating with scripts. Among the API functions of the Apache Cloudstack platform are relevant (JoeySneddon, 2017).

Another task of backing up our academic cloud is to estimate the time required to upload data to the cloud storage. Currently (June 2022) the sizes of our academic cloud storage is approximately as follows:

- primary250 – 132 Gb;
- primary251 – 70 Gb;
- primary252 – 126 Gb;
- primary253 – 88 Gb;
- secondary – 110 Gb.

Since we make a full copy of the primary storage, we need to download about 400 GB to the cloud storage each time. Let the speed of the Internet channel at night be 80 Mbps (10 Mbytes per second). Then it will take 11 hours to download 400*1024 MB. That’s a lot. Therefore, we balanced Internet access through 2 providers. At the time of backup, our router routes hosts cloud0 and cloud1 through the first provider, and cloud2 and cloud3 through the second. In this case, a full backup takes about 6 hours. This time is also significant, but is acceptable.

Another disadvantage of our scheme is the significant time required to download backups from the Google Drive service. However, this time will be significant if the management or storage servers fail. This means that we must back up the entire OS of the management server to fast local area network storage.

As mentioned earlier Proxmox VE provides many tools, including storage and backup types. To back up all cloud infrastructure, it is advisable to install and configure Proxmox Backup Server. It is an enterprise backup solution, for backing up and restoring VMs, containers, and physical hosts. By supporting incremental, fully deduplicated backups, Proxmox Backup Server significantly reduces network load and saves valuable storage space. Unfortunately,

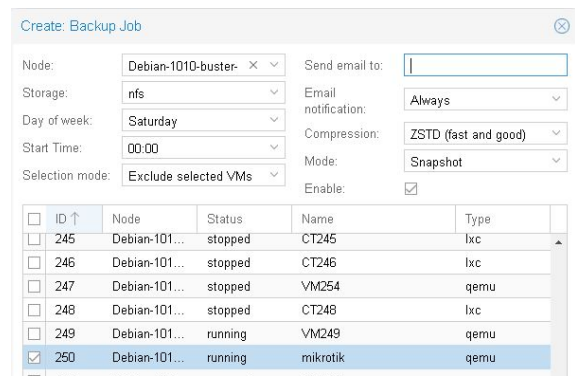


Figure 4: Proxmox backup job

we do not have the physical resources to install a separate Proxmox Backup Server. Nevertheless, we have performed some tasks from the cloud infrastructure backup model.

Stop students’ VMs. There is no need to use Proxmox API to perform this task. We already have student VM_ID (used them to establish access rights). To stop these VMs on the Proxmox host, we executed the commands `qm shutdown VM_ID` or `pct stop container.ID`. It is advisable to use such commands to enable (disable) student VMs and containers at the beginning (end) of classes.

VM reservation. We have set up a scheduled task to back up the cloud infrastructure to an NFS share. It is connected to our university local network via an OpenVPN tunnel. This allows us to save the condition of all machines.

Standard Proxmox tools allow cloud administrator to perform this task flexibly. The following backup options are available (figure ??):

- host (node) to be copied;
- storage for which to perform job;
- date and time of the task;
- backup objects such as some VMs, all VMs, all except some VMs;
- compression method (LZO, GZIP, ZSTD, without compression)
- copy mode (snapshot, suspend, or turn off VM)
- email notification about the status of the job or errors during its execution

The advantage of this method is the reservation of working VMs "on the fly". Like Apache CloudStack we use VMs backup from Proxmox cloud infrastructure into Google Drive storage.

5 CONCLUSIONS

The private academic clouds should be used in cloud based learning environment, as they are necessary for education of future ICT-specialists. Despite the availability of educational grants from leading cloud vendors, many universities are deploying their own private academic clouds. Cloud administrators have a lot of work to do to maintain and support these academic clouds. Among these tasks, one of the most important is to ensure the productivity and elasticity of the cloud. Solving them will allow them to load the maximum number of VMs in the cloud infrastructure.

In this study, the performance of academic clouds based on Apache Cloudstack and Proxmox VE was evaluated. Despite some assumptions in comparison, we can state that the platforms have approximately the same performance and therefore can provide the same amount of VM for students. This is because they both use the same KVM hypervisor. As our experience has shown, the process of cloud deployment based on Proxmox VE is technically simpler and faster. This is even though we have installed Apache CloudStack many times and Proxmox VE for the first time.

An important task in the maintenance of the academic private cloud is the backup of its components. To solve it effectively, you need to use different backup schemes such as full, incremental, differential. To save data, it is advisable to use both cloud and local storage. In any case, administrators should determine how long it will take to build and restore the entire cloud infrastructure. It is also advisable to use the API functions of the cloud platform. This will automate some maintenance tasks.

To back up Apache Cloudstack, we had to use API and develop our own tools. In any case, this process requires large repositories and fast networks. If the first requirement is not met, it is advisable to use cloud storage. We did so and copy the backups to Google Drive. Backup of Proxmox infrastructure is technically simpler. To ensure the full functionality of this process, it is advisable to install a Proxmox Backup Server. All of the above factors indicate that it is time to migrate our academic cloud to the Proxmox platform. Maybe for a while we'll use them both together.

REFERENCES

- AL-Mukhtar, M. M. and Mardan, A. A. A. (2014). Performance evaluation of private clouds eucalyptus versus cloudstack. *International Journal of Advanced Computer Science and Applications*, 5(5).
- Algarni, S., Ikbal, M. R., Alroobaea, R., Ghiduk, A. S., and Nadeem, F. (2018). Performance evaluation of xen, kvm, and proxmox hypervisors. *International Journal of Open Source Software and Processes (IJOSSP)*, 9(2):39–54.
- Amiel, T., ter Haar, E., Vieira, M. S., and Soares, T. C. (2020). Who benefits from the public good? how eor is contributing to the private appropriation of the educational commons. In Burgos, D., editor, *Radical Solutions and Open Science. Lecture Notes in Educational Technology*, pages 69–89. Springer, Singapore.
- Angus, P. (2020). Cloudstack backup and recovery framework. <https://www.slideshare.net/ShapeBlue/cloudstack-backup-and-recovery-framework>.
- Bykov, V., Mikulowski, D., Moravcik, O., Svetsky, S., and Shyshkina, M. (2020). The use of the cloud-based open learning and research platform for collaboration in virtual teams. *Information Technologies and Learning Tools*, 76(2):304–320.
- Fylaktopoulos, G., Goumas, G., Skolarikis, M., Sotiropoulos, A., and Maglogiannis, I. (2016). An overview of platforms for cloud based development. *SpringerPlus*, 5(38):1–13.
- GCP (2020). Google cloud platform. <https://www.internet2.edu/products-services/cloud-services-applications/google-cloud-platform/>.
- Glazunova, O. (2015). *Theoretical and methodological bases for the design and application of an e-learning system for future IT specialists in an agrarian university*. PhD thesis, Institute of Information Technologies and Learning Tools of the NAPS of Ukraine, Kyiv, Ukraine. Manuscript.
- Glazunova, O. and Shyshkina, M. (2018). The concept, principles of design and implementation of the university cloud-based learning and research environment. In Ermolayev, V., editor, *Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops*, volume 2104, pages 332–347. CEUR Workshop Proceedings.
- Ilin, D. Y., Volovich, M., and Filatov, V. (2016). Analysis of cloudstack platform suitability for management of different cloud infrastructure configurations. <https://www.semanticscholar.org/paper/Analysis-of-CloudStack-Platform-Suitability-for-of-Ilin-Volovich/0d0107f52929243803156389125986eb4376e2f2>.
- JoeySneddon (2017). Mount your google drive on linux with google-drive-ocamlfuse. <https://www.omgubuntu.co.uk/2017/04/mount-google-drive-ocamlfuse-linux>.
- Khmelevsky, Y. and Voytenko, V. (2015). Hybrid cloud computing infrastructure in academia. In *WCCCE 2015 - the 20th Western Canadian Conference on Computing Education*. Vancouver Island University, Nanaimo, British Columbia, Canada.
- Korotun, O. V., Vakaliuk, T. A., and Soloviev, V. N. (2019). Model of using cloud-based environment in training databases of future it specialists. In Kiv, A. E. and Shyshkina, M. P., editors, *Proceedings of the 7th Workshop on Cloud Technologies in Education (CTE*

- 2019), volume 2643, pages 281–292. CEUR Workshop Proceedings.
- Lytvynova, S. H. (2017). Cloud-oriented learning environment of secondary school. In Semerikov, S. O. and Shyshkina, M. P., editors, *Proceedings of the 5th Workshop on Cloud Technologies in Education (CTE 2017)*, volume 2168, pages 7–12. CEUR Workshop Proceedings.
- Markova, O. M., Semerikov, S. O., Striuk, A. M., Shalatska, H. M., Nechypurenko, P. P., and Tron, V. V. (2018). Implementation of cloud service models in training of future information technology specialists. In Kiv, A. E. and Soloviev, V. N., editors, *Proceedings of the 6th Workshop on Cloud Technologies in Education (CTE 2018)*, volume 2433, pages 499–515. CEUR Workshop Proceedings.
- OAUTH2 (2020). Using oauth 2.0 to access google apis. <https://developers.google.com/identity/protocols/oauth2>.
- Oleksiuk, V., Oleksiuk, O., and Berezitskyi, M. (2017). Planning and implementation of the project “cloud services to each school”. In Ermolayev, V., editor, *Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*, volume 1844, pages 372–379. CEUR Workshop Proceedings.
- Paulsen, K. (2011). 12 - archives, backups, and linear tape. In Paulsen, K., editor, *Moving Media Storage Technologies*, pages 327–377. Focal Press, Boston.
- Rao, U. H. and Nayak, U. (2014). Data backups and cloud computing. In *The InfoSec Handbook*, pages 263–288. Apress, Berkeley, CA.
- Reddy, P. V. V. and Shyamala, K. (2016). New scoring formula to rank hypervisors’ performance complementing with statistical analysis using doe. *Future Generation Computer Systems*, 61:54–65.
- Rusyn, B., Pohreliuk, L., Vysotska, V., and Osypov, M. (2019). Method of data dedublication and distribution in cloud warehouses during data backup. *Information systems and networks*, 6:1–12.
- SDR (2020). Student developer resources. <https://azure.microsoft.com/en-us/developer/students/>.
- Shyshkina, M. P. (2016). The hybrid cloud-based service model of learning resources access and its evaluation. In Ermolayev, V., editor, *Proceedings of the 12th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer*, volume 1614, pages 241–256. CEUR Workshop Proceedings.
- Spirin, O., Oleksiuk, V., Balyk, N., Lytvynova, S., and Sydorenko, S. (2019). The blended methodology of learning computer networks: Cloud-based approach. In Ermolayev, V., editor, *Proceedings of the 15th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops*, volume 2393, pages 68–80. CEUR Workshop Proceedings.
- Spirin, O., Oleksiuk, V., Oleksiuk, O., and Sydorenko, S. (2018). The group methodology of using cloud technologies in the training of future computer science teachers. In Ermolayev, V., editor, *Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops*, volume 2104, pages 294–304. CEUR Workshop Proceedings.
- Spirin, O. M., Vakaliuk, T. A., Ievdokymov, V. V., and Sydorenko, S. I. (2022). Criteria for selecting a cloud-based learning management system for a higher education institution. *Information Technologies and Learning Tools*, 89(3):184–202.
- Vorozhbyt, A. V. (2019). Creation of multimedia content of the cloud-based learning environment in technical lyceum. *New Computer Technology*, 17:59–63.
- Wang, B., Wang, C., Song, Y., Cao, J., Cui, X., and Zhang, L. (2020a). A survey and taxonomy on workload scheduling and resource provisioning in hybrid clouds. *Cluster Computing*.
- Wang, B., Wang, C., Song, Y., Cao, J., Cui, X., and Zhang, L. (2020b). A survey and taxonomy on workload scheduling and resource provisioning in hybrid clouds. *Cluster Computing*, 23(4):2809–2834.
- Yallop, J., Sheets, D., and Madhavapeddy, A. (2018). A modular foreign function interface. *Science of Computer Programming*, 164:82–97.