Some experience in maintenance of an academic cloud

Vasyl P. Oleksiuk[1,4], Olesia R. Oleksiuk[2], Oleg M. Spirin[3,4], Nadiia R. Balyk[1] and Yaroslav P. Vasylenko[1]

[1] *Ternopil Volodymyr Hnatiuk National Pedagogical University, 2 M. Kryvonosa Str., Ternopil, 46027, Ukraine*

[2] *Ternopil Regional Municipal Institute of Postgraduate Education, 1 V. Hromnytskogo Str., Ternopil, 46027, Ukraine*

[3] *University of Educational Management, 52-A Sichovykh Striltsiv Str., Kyiv, 04053, Ukraine*

[4] *Institute of Information Technologies and Learning Tools of NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine*

Abstract

The article is devoted to the systematization of experience in the deployment, maintenance and servicing of the private academic cloud. The article contains model of the authors' cloud infrastructure. It was developed at Ternopil Volodymyr Hnatiuk National Pedagogical University (Ukraine) on the basis of the Apache CloudStack platform. The authors identify the main tasks for maintaining a private academic cloud. Here they are making changes to the cloud infrastructure; maintenance of virtual machines (VM) to determine the performance and migration of VM instances; work with VMs; backup of all cloud infrastructure. The analysis of productivity and providing students with computing resources is carried out. The main types of VM used in training are given. The number and characteristics of VM that can be served by a private academic cloud are calculated. Approaches and schemes for performing backup are analysed. Some theoretical and practical experience of using cloud services to perform backup has been studied. Several scripts have been developed for archiving the platform database and its repositories. They allow you to upload backups to the Google Drive cloud service. The performance of these scripts for the author's deployment of private cloud infrastructure was evaluated.

Keywords

cloud computing, private academic cloud, Apache CloudStack, G Suite, Google Drive

1. The problem statement

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 [1, 2, 3, 4], 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 [5].

CTE 2020: 8th Workshop on Cloud Technologies in Education, December 18, 2020, Kryvyi Rih, Ukraine

oleksyuk@fizmat.tnpu.edu.ua (V. P. Oleksiuk); o.oleksyuk@ippo.edu.te.ua (O. R. Oleksiuk); oleg.spirin@gmail.com (O. M. Spirin); nadbal@fizmat.tnpu.edu.ua (N. R. Balyk); yava@fizmat.tnpu.edu.ua (Y. P. Vasylenko)

https://www.fizmat.tnpu.edu.ua/chairs/inform/infoemployees/186-oleksyuk (V. P. Oleksiuk); https://www.fizmat.tnpu.edu.ua/chairs/inform/infoemployees/137-nadbal (N. R. Balyk)

0000-0003-2206-8447 (V. P. Oleksiuk); 0000-0002-1454-0046 (O. R. Oleksiuk); 0000-0002-9594-6602 (O. M. Spirin); 0000-0002-3121-7005 (N. R. Balyk); 0000-0002-2520-4515 (Y. P. Vasylenko)

As the analysis of the literature shows, many university CBLE are usually deployed according to the hybrid model [6, 7, 8]. One of the most important components in the structure of this environment is the private academic cloud [9]. 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 [10].

Various commercial and free platforms are used in universities to build private academic clouds [11, 12, 13, 14]. 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 [15]. 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 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 [16]. 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, 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 [16, 17]:

- 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 rounted GUI, can provide an advanced cloud infrastructure, but it is very GUI centric built on single Java core.

We have deployed private academic cloud based on the Apache CloudStack platform.

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.

In general, our private academic cloud provides [17]:

- 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 problems and corresponding tasks in the process private academic cloud using.

The purpose of the article is to systematize author's experience in maintenance the Cloud Based Learning Environment.

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 cloud which has deployed by authors.
3. Listing and systematization of author's experience in maintenance of academic cloud of Ternopil Volodymyr Hnatiuk National Pedagogical University.

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 [18]. 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 [19].

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 [20].

Junfeng Tian, Zilong Wang and Zhen Li also studied cloud data backup [21]. 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.

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 [22].

3. Definition of maintenance tasks of authors' academic cloud

Here are the main tasks for servicing our sample private academic cloud:

1. Work with student accounts;
2. Making changes to the cloud infrastructure;
3. Creating VM templates;
4. VM service (system, student, teacher);
5. Determining the performance of individual hosts and the cloud as a whole;
6. Migration of VM specimens;
7. Stopping and restarting physical hosts;
8. Cloud infrastructure backup.

The first task 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.

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.

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.

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. This processor frequency is measured by each hypervisor in the Apache CloudStack platform.

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

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

As table 1 shows, the private academic cloud has a total frequency of about 50GHz. And the total amount of memory is about 90 Gb. Regarding the frequency, two other opposite factors should be taken into account:

- Table 1 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.

Table 1
Main Characteristics of Academic Cloud's Hardware

	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
Sum			90112	52400

Comparing the data in table 2 and table 1, we can conclude that our academic cloud provides about 50 VM with Linux without a graphical user interface (GUI), more than 40 VM with Windows Workstation, about 35 VM with Windows Server and OC Linux with GUI.

Table 2
Basic calculations of our academic cloud's performance

OS	F_{OS}	MEM_{OS}	N_{st}	FRQ	MEM
OSLinuxNoGUI	500	500	20	10000	10000
OSLinuxGUI	1500	2000	20	30000	40000
OSWindowsWs	1000	2000	20	20000	40000
OSWindowsSrv	1500	2000	20	30000	40000
OSAdvLinux	2500	4000	20	50000	80000

We use the EVE-NG platform for modelling in the study of computer networks. It launches its own VMs inside the main Apache CloudStack VM [23]. Such nested virtualization requires more resources. Therefore, Table 1 has a row named OSAdvLinux. For this OS, our cloud can run about 20 instances. We have created in our cloud infrastructure some compute offering templates based on the data provided.

Regarding VM migration, we used the approach described in [21]. Its authors propose to evaluate the efficiency of the cloud infrastructure as an integrated indicator of the use of resources of each instance of the VM. The authors indicate that a specific instance needs to be migrated to another host to resolve the issue. They propose the concept of non-uniformity, which is determined by the ratio:

$$N_R^p = \sqrt{\sum_{i=1}^{n} \left(\frac{r_i - \bar{r}}{\bar{r}}\right)^2},$$

where n is the number of resources, r_i is the projected use of the i-th resource, $\bar{r}$ is the average predicted value of the use of all resources of the p-th server. Academic cloud administrators should minimize value N_R^p.

To define an overloaded host, the concept of "hot spot" is used. The host will be "hot" if at least one of its resources exceeds the limit value ("temperature"). To determine the host's "temperature" the amount of use of all its resources.

$$t^* = \sum_{t \in R}^{n} (r_i - r_t)^2 \,,$$

If the value of t^* is greater than zero, then virtual machines should be migrated from the appropriate host. Apache CloudStack system implements the appropriate functionality. Root or domain administrator can transfer both disks of virtual machines and run them on another host.

Another way to solve the problem of lack of computing resources is CPU and RAM overcommit. In this case, the Apache CloudStack system administrator sets the multiplier. This number is multiplied by the total CPU frequency or amount of RAM. However, this method should not be abused. This can lead to unpredictable consequences, such as denial of service to virtual machines.

4. Designing and realization an academic clouds' backup model

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. Large companies are developing 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 [24].

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 backup is relevant for almost all IT infrastructures. When choosing a backup method, the following criteria are important:

- 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.

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 [25]. 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 [25]. 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).

Our research was performed at the Joint Laboratory of the Institute of Information Technologies and Learning Tools of the National Academy of Educational Sciences of Ukraine, and Ternopil Volodymyr Hnatiuk National Pedagogical University.

Our academic cloud deployment contains the following objects:

- One management server;
- Four hosts for running VMs instances;
- Four primary repositories containing disks of these VMs;
- One secondary repository for saving templates and ISO images.

Because templates and ISO images 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 back-up 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.

We used the latest utility. Here are its main features [26]:

- 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 secondary storage itself, but a copy of it from the backup drive (Backup_Secondary task, see figure 1). 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.

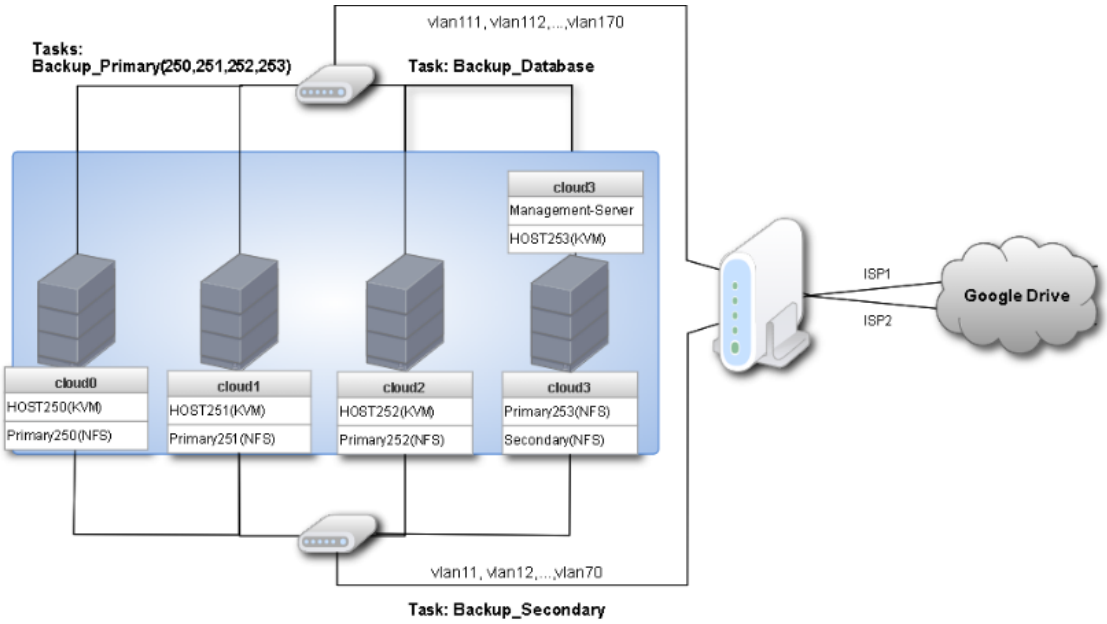

Figure 1: Academic cloud infrastructure backup scheme.

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 1). 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 ac-count 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 >
$BACKUP_DIR/"archive_cloud_all_""$date_daily"".sql"

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

The variable *$date_daily* contains the current date of the archive. This allows you to see the date of archiving directly in the file name.

To upload the files to the server, we used a ready-made script from github [27]. Here is its launch:

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

where

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

A special refresh_token token is required to provide long-term access of the up-load.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 cloud infrastructure is shown in figure 1.

Performing backup of primary repositories (Backup_Primary task (250,251,252,253)) 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 would be good to prepare a cloud platform, stopping all VMs. Of course, students need to form an understanding of the need to turn off their own VM. 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 CloudStack 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 [28].

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 CloudStack platform provides 2 types of images [22]:

- 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 Cloud-Stack platform are relevant [29].

Another task of backing up our academic cloud is to estimate the time required to upload data to the cloud storage. Currently (October 2020) the sizes of our academic cloud storage is approximately as follows:

- primary250 – 120 Gb;
- primary251 – 80 Gb;
- primary252 – 140 Gb;
- primary253 – 80 Gb;
- secondary – 100 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 5 hours and 30 minutes. 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.

5. Conclusions

The private academic clouds should be used in cloud based learning environment, as they are necessary for education of future ICT specialists. Virtualization is one of the most up-to-date and advanced technologies for modelling many ICT objects. Despite the availability of educational grants from leading cloud vendors, many universities are deploying their own private academic clouds. During the production phase, 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.

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.

We see the prospects for our further research of our installation of a private academic cloud in the development of more efficient scripts based on a differential circuit. They should reduce the time it takes to create and copy all backups. According-ly, the time to recover data from it will be reduced. Also relevant the study of new versions of cloud platforms regarding the emergence of ready-made modules for backup. Probably, they will allow to solve many current problems.

References

[1] P. Merzlykin, M. Popel, S. Shokaliuk, Services of SageMathCloud environment and their didactic potential in learning of informatics and mathematical disciplines, CEUR Workshop Proceedings 2168 (2017) 13–19.

[2] O. Glazunova, M. Shyshkina, The concept, principles of design and implementation of the university cloud-based learning and research environment, CEUR Workshop Proceedings 2104 (2018) 332–347.

[3] O. V. Korotun, T. A. Vakaliuk, V. N. Soloviev, Model of using cloud-based environment in training databases of future it specialists, CEUR Workshop Proceedings 2643 (2019) 281–292.

[4] V. Bykov, D. Mikulowski, O. Moravcik, S. Svetsky, M. Shyshkina, The use of the cloud-based open learning and research platform for collaboration in virtual teams, Information

Technologies and Learning Tools 76 (2020) 304–320. URL: https://journal.iitta.gov.ua/index.php/itlt/article/view/3706. doi:10.33407/itlt.v76i2.3706.

[5] S. H. Lytvynova, Cloud-oriented learning environment of secondary school, CEUR Workshop Proceedings 2168 (2017) 7–12.

[6] M. Shyshkina, The hybrid cloud-based service model of learning resources access and its evaluation, CEUR Workshop Proceedings 1614 (2016) 241–256.

[7] O. M. Markova, S. O. Semerikov, A. M. Striuk, H. M. Shalatska, P. P. Nechypurenko, V. V. Tron, Implementation of cloud service models in training of future information technology specialists, CEUR Workshop Proceedings 2433 (2018) 499–515.

[8] A. V. Vorozhbyt, Creation of multimedia content of the cloud-based learning environment in technical lyceum, New Computer Technology 17 (2019) 59–63.

[9] O. Glazunova, Theoretical and methodological bases for the design and application of an e-learning system for future IT specialists in an agrarian university, D.Sc. thesis, Institute of Information Technologies and Learning Tools of the NAES of Ukraine, Kyiv, Ukraine, 2015.

[10] B. Wang, C. Wang, Y. Song, J. Cao, X. Cui, L. Zhang, A survey and taxonomy on workload scheduling and resource provisioning in hybrid clouds, Cluster Computing 23 (2020) 2809–2834. URL: https://doi.org/10.1007/s10586-020-03048-8. doi:10.1007/s10586-020-03048-8.

[11] M. Despotović-Zrakić, K. Simić, A. Labus, A. Milić, B. Jovanić, Scaffolding environment for e-learning through cloud computing, Journal of Educational Technology & Society 16 (2013) 301–314. URL: https://www.researchgate.net/publication/286123164_Scaffolding_environment_for_e-learning_through_cloud_computing.

[12] M. M. AL-Mukhtar, A. A. A. Mardan, Performance Evaluation of Private Clouds Eucalyptus versus CloudStack, International Journal of Advanced Computer Science and Applications 5 (2014). URL: http://dx.doi.org/10.14569/IJACSA.2014.050516. doi:10.14569/IJACSA.2014.050516.

[13] D. Y. Ilin, M. Volovich, V. Filatov, Analysis of CloudStack Platform Suitability for Management of Different Cloud Infrastructure Configurations, Cloud of science 3 (2016) 433–443. URL: http://web.archive.org/web/20180422194515/https://cloudofscience.ru/sites/default/files/pdf/CoS_3_433.pdf.

[14] T. Amiel, E. ter Haar, M. S. Vieira, T. C. Soares, Who benefits from the public good? how oer is contributing to the private appropriation of the educational commons, in: D. Burgos (Ed.), Radical Solutions and Open Science: An Open Approach to Boost Higher Education, Springer Singapore, Singapore, 2020, pp. 69–89. URL: https://doi.org/10.1007/978-981-15-4276-3_5. doi:10.1007/978-981-15-4276-3_5.

[15] GCP, Google Cloud Services (GCP), 2021. URL: https://internet2.edu/services/google-cloud-platform/.

[16] Microsoft Azure, Student developer resources, 2021. URL: https://azure.microsoft.com/en-us/developer/students/.

[17] G. Fylaktopoulos, G. Goumas, M. Skolarikis, A. Sotiropoulos, I. Maglogiannis, An overview of platforms for cloud based development, SpringerPlus 5 (2016) 38. URL: https://doi.org/10.1186/s40064-016-1688-5. doi:10.1186/s40064-016-1688-5.

[18] O. Spirin, V. Oleksiuk, O. Oleksiuk, S. Sydorenko, The group methodology of using

cloud technologies in the training of future computer science teachers, CEUR Workshop Proceedings 2104 (2018) 294–304. URL: http://ceur-ws.org/Vol-2104/paper_154.pdf.

[19] Y. Khmelevsky, V. Voytenko, Hybrid cloud computing infrastructure in academia, in: WCCCE 2015 - the 20th Western Canadian Conference on Computing Education, Vancouver Island University, Nanaimo, British Columbia, Canada, 2015. URL: https://www.researchgate.net/profile/Youry_Khmelevsky/publication/282778407_Hybrid_Cloud_Computing_Infrastructure_in_Academia/links/56873fe108ae1e63f1f5b884.pdf. doi:10.13140/RG.2.1.4082.6647.

[20] U. H. Rao, U. Nayak, Data backups and cloud computing, in: The InfoSec Handbook: An Introduction to Information Security, Apress, Berkeley, CA, 2014, pp. 263–288. URL: https://doi.org/10.1007/978-1-4302-6383-8_13. doi:10.1007/978-1-4302-6383-8_13.

[21] J. Tian, Z. Wang, Z. Li, Low-cost data partitioning and encrypted backup scheme for defending against co-resident attacks, EURASIP Journal on Information Security 2020 (2020) 7. URL: https://doi.org/10.1186/s13635-020-00110-1. doi:10.1186/s13635-020-00110-1.

[22] B. Rusyn, L. Pohreliuk, V. Vysotska, M. Osypov, Method of data dedublication and distribution in cloud warehouses during data backup, Information systems and networks 6 (2019) 1–12. doi:10.23939/sisn2019.02.001.

[23] O. Spirin, V. Oleksiuk, N. Balyk, S. Lytvynova, S. Sydorenko, The blended methodology of learning computer networks: Cloud-based approach, CEUR Workshop Proceedings 2393 (2019) 68–80.

[24] P. Angus, Cloudstack backup and recovery framework, 2020. URL: https://www.slideshare.net/ShapeBlue/cloudstack-backup-and-recovery-framework.

[25] V. Oleksiuk, O. Oleksiuk, M. Berezitskyi, Planning and Implementation of the Project "Cloud Services to Each School", CEUR Workshop Proceedings 1844 (2017) 372–379. URL: http://ceur-ws.org/Vol-2393/paper_231.pdf.

[26] R. Hantimirov, A. Mikryukov, Model distribution of resources in the operation of cloud computing environment, Open education 5 (2015).

[27] Comparison of backup methods, 2014. URL: https://habr.com/ru/company/selectel/blog/226831/.

[28] Using OAuth 2.0 to Access Google APIs, 2020. URL: https://developers.google.com/identity/protocols/oauth2.

[29] Joey Sneddon, Mount Your Google Drive on Linux with google-drive-ocamlfuse, 2017. URL: https://www.omgubuntu.co.uk/2017/04/mount-google-drive-ocamlfuse-linux.