

Enterprise Application Backup and Restore in Cloud Computing

Hung Ba Ngo*, Tru Cong Huynh

Cantho University, Campus II, 3/2 street, Ninhkieu district, Cantho city, Vietnam.

* Corresponding author. Email: nbhung@cit.ctu.edu.vn

Manuscript submitted December 10, 2015; accepted April 4, 2016.

doi: 10.17706/ijcee.2016.8.2.169-176

Abstract: Backing up and restoring enterprise applications (EAs) are essential operations that are usually done manually by an administrator or semi-automatically by tools. Backing up an enterprise application (EA) consists of backing up all its components together with the environments and operating systems in a consistent way so that the application can be restored and function properly later. There are many challenges for backing up and restoring EAs. This paper proposes a framework for automating the process of backing up and restoring EAs deployed in a cloud environment to ensure the continuous running of these enterprise applications and prevent the loss of data in case failure of applications.

Key words: Backup, cloud computing, enterprise application, restore.

1. Introduction

Enterprise Applications (EAs) [1] are now known as complex, scalable, distributed, component-based, and mission-critical applications to solve an enterprise-wide problem. From physical view, an enterprise application (EA) can be composed of three principal components: *application code*, *database* and *static resources* that are usually deployed in different machines to increase throughput of an application. In addition, each component needs an environment on which it is deployed. For EAs developed based on J2EE platform¹, their application codes need a Servlet container such as Tomcat² to operate. Application database has to be manipulated by a database management system (DBMS). Static resources of application (uploaded files) need a file server to manage them. Therefore, backing up an EA consists of backing up all its components together with the environments and operating systems in a consistent way so that the application can be restored and function properly later. Backing up and restoring applications are usually done manually by an administrator or semi-automatically by tools. There are many challenges for backing up and restoring EAs. It takes a lot of time for an administrator to backup all components of an EA. To decrease the effort of backup, administrators often backup only critical/core components of an EA such as application database. Many solutions mentioned [2] to help administrators to back up mainly application databases. Frequency of backup is not high. It increases a risk of data loss. It takes a lot of time to restore an application because many application's components have to be reinstalled from the crash. Recently, cloud computing paradigm [3] has made a significant change in the way of deploying applications. Virtual machine (VMs) provisioned from the cloud can be used to deploy applications instead of traditional

¹<http://www.oracle.com/technetwork/java/javaee/overview/index.html>

²<http://tomcat.apache.org/>

physical servers. Cloud platforms support snapshot mechanism [4] allowing administrators to back up a whole Virtual machine (VM) into a single image file. A VM can be restored later from its image file. Time for doing a snapshot or restoring a VM in a cloud environment is quite short, just in minutes. According to [5], the snapshot will be the main mechanism of backing up and restoring application in the near future. Although snapshot can back up and restore application components together with their runtime environment in minutes, the snapshot has not done backup and restore for a whole EA. Still the application administrators have to do their job manually. This paper proposes a framework that is suitable for backing up and restoring EAs deployed in a cloud environment. The proposed framework will automate the process of backing up EAs, monitor status of EAs, restore automatically a component to ensure the continuously operating of EAs. The paper, particularly focus on EAs developed based on J2EE³ and deployed in a OpenStack⁴ cloud environment. The paper will be structured as the following. The next section will discuss more about the challenges in backing up and restoring EAs. The proposed framework for backing up and restoring EAs deployed in cloud environment will be presented in the third section. The fourth section will be our implementation and evaluation of the proposed framework. The last section is our conclusions and future works.

2. Challenges in Backing up and Restoring Enterprise Applications

Backing up an EA is a set of operations to create a copy of the application and store the copy in a secure storage device. Reversely, restoring an EA is a set of operations to reconstruct the execution of the application from a copy of the application. Executing of an EA can be constructed from six components such as application code, runtime environment, application database, database management system (DBMS), static resources and file service. Backing up an EA therefore requires that all components belonged to the application have to be backed up. Each component can be planned to be backed up at different points of time. A backup version of an EA therefore is a set of backup versions of its components.

Restoring a backup version of an EA can be done by firstly finding a set of backup versions of its component, then doing the restore of backup versions of each component. Backup and restore are essential operations to enhance the availability of an application in preventing hardware failure and ensure security of application data. Backing up and restoring EAs are complex operations taking a lot of time and effort of administrators. Many backup and restore solutions are proposed [6]-[9] to automate backup and restore operations. However, these solutions mainly focus on backing up application databases. The other components of the application have not been considered. Recently, cloud computing paradigm allows to deploy applications on VMs that can be backed up and restored easily and quickly. *Snapshot* mechanism supported by the cloud platform can be used to create an *image file* that backups the current state of machine including state of applications running on that machine. The image file can be used to restore the VM later. Although the snapshot is considered as a powerful mechanism to backup and restore applications deployed in a cloud environment, there are still issues that snapshot mechanism has not resolved. For application components that change rapidly and frequently such as database components or static resources, snapshot cannot be done every minute to prevent the risk of losing data that are created or modified between the two success snapshot. In addition, snapshot mechanism is not responsible for creating backup versions of an application that composes from snapshots of many VMs. Snapshot also lacks mechanisms that monitor the executing status of application components to detect failure of components needed to restore automatically. Therefore, the next section will propose a framework that automate operations of backing up and restoring EA deployed in a cloud environment.

³<http://www.oracle.com/technetwork/java/javaee/overview/index.html>

⁴<http://www.openstack.org/>

3. Framework for Automating Operations of Backing up and Restoring Enterprise Applications Deployed in Cloud Environment

This section will discuss a framework for automating operations of backing up and restoring EAs that are deployed in a cloud environment providing Infrastructure as a Service (IaaS), particularly for private cloud built from open source cloud management platforms such as OpenStack [8], OpenNebula⁵, Eucalyptus⁶ and CloudStack⁷. As discussed in previous section, an EA may have many backup versions. The proposed framework in this paper is limited to backup and restore the most recently version of an EA.

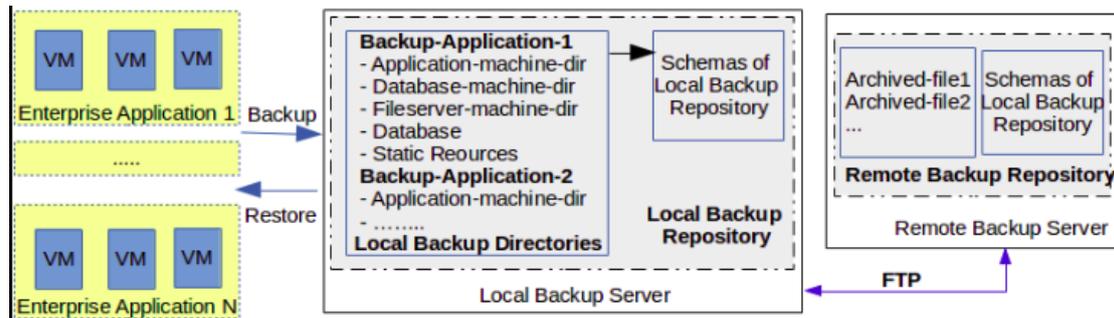


Fig. 1. Proposed framework for backing up and restoring EAs on cloud environment.

The proposed framework, as described in Fig. 1, requires that several VMs are reserved for each EA to deploy its components. A VM cannot be shared between two or more EAs. There are three types of VMs such as *Application machine*, *Database machine* and *File server machine* that host correspondingly application code components, application database components and static resource components of EAs. One machine (virtual or physical), so called *Local backup server*, located in data center hosting cloud environment where EAs are deployed is used to manage *Local backup repository*. Another machine located on the site that is a far way from the current data center will play the role of *Remote backup repository*.

3.1. Mechanism for Backing up Enterprise Applications

Local backup of an EA consists of five operations. *Operation 1*, Backup application machine, creates a snapshot of application machine and stores its image file into Local backup repository. This operation is responsible for backing up application code component and its runtime environment (operating system, middleware, libraries,..). *Operation 2*, Backup database machine, creates a snapshot of database machine and stores its image file into the Local backup repository. This operation is responsible for backing up database management system and application database component. *Operation 3*, Backup file server machine, creates a snapshot of file server machine and stores its image file into Local backup repository. This operation is responsible for backing up file service component and application static resources component. *Operation 4*, Backup application database, creates a new revision for the application database component and stores it into Local backup repository. This operation should be performed more frequently than the first three above operations to prevent the risk of losing data. *Operation 5*, Backup application static resources, creates a new revision for application static resources component and store it into Local repository. This operation also should be performed more frequently than the first three above operations to prevent the risk of losing data.

⁵<http://opennebula.org/>

⁶<http://www8.hp.com/us/en/cloud/helion-eucalyptus-overview.html>

⁷<https://cloudstack.apache.org/>

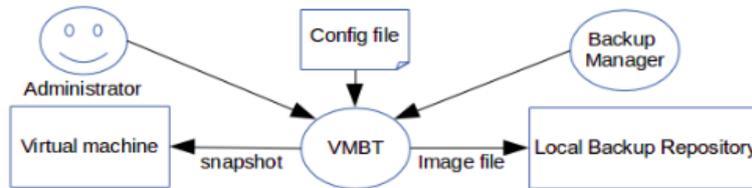


Fig. 2. Mechanism of virtual machine backup.

In terms of backup mechanism, the first three operations have the same mechanism. This is the snapshot mechanism supported by a cloud platform providing IaaS. However, a method for naming snapshot files in local repository should be defined to distinguish the backup of different machines and to distinguish different backup version of the same machine. We use notion *Virtual Machine Backup* to mention the backup mechanism for the first three operations.

Mechanism for backing up virtual machines: Fig. 2 presents a mechanism for backing up VMs. *Virtual Machine Backup Tool (VMBT)* maintains a list of identifiers of VMs that need to be backed up. *VMBT* can be called by application administrator or by *Backup Manager*. In fact, *VMBT* is only called by the application administrator whenever the application is upgraded or reconfigured.

Mechanism for backing up application database: Fig. 3 presents a mechanism for backing up application database. Usually, database management system creates log files to record all transactions occurred in its databases. *DB Backup Agent* uses these log files to create *Datalog files* that can be used to restore a database later. Datalog files are stored in a *DB cached directory* on the database server machine. In Local backup repository, a DB backup directory for application database is created. *RSync client* and *RSync Server* are configured to synchronize the content between DB cached directory and DB backup directory. The *Timer* is set to launch *RSync Client* periodically so that Datalog files in DB cached are backed up into DB backup directory.

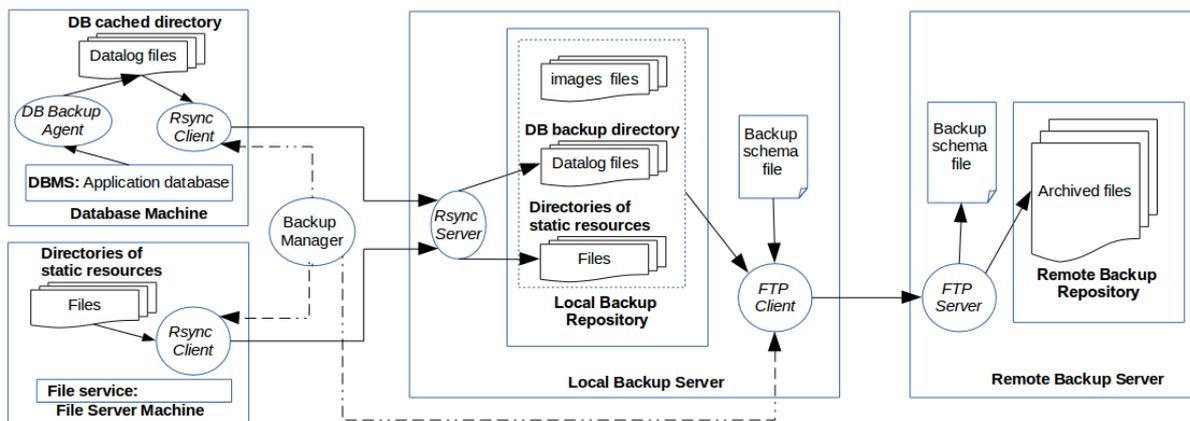


Fig. 3. Mechanism for backing up database and static resources of an EA.

Mechanism for backing up application static resources: When an application is running, it usually creates files in predefined directories, so-called *Directories of static resources*. *Rsync Client* and *Rsync Servers* are also configured to upload files in the *Directories of static resources* into *Local Backup Repository* as described in Fig. 3. *Backup Manager* should be configured so that operations of backing up the database and operations of backing up static resources are performed together.

Mechanism for remotely backing up enterprise applications: The whole EA is now backed up in *Local backup repository*. To prevent the failure of hardware or natural disaster, *Local backup repository* need to be backed up at a remote site far away from *Local backup repository* as described in Fig. 3. *Backup Manager*

will planned to launch *Remote Backup Client* to send files in Local backup repository to *Remote backup repository*. Each backup directory in Local backup repository will be compressed into a file which name presents an application component and time which is backed up to Remote backup repository. The remote backup client uses an FTP client to send compressed files to FTP server on Remote backup repository. In Local backup server, a *Backup schema file* is maintained to describe the actual organizational structure of Local backup repository and tells which set of backup directories belonged to an application. This Backup schema file has to be sent to Remote backup repository so that the administrator knows how to restore applications from Remote backup repository. Backup Manager should be planned to void process of local backup and process of remote backup for an application activated concurrently.

3.2. Mechanism for Restoring Enterprise Applications

Mechanism for local restoring enterprise applications: *Restore Manager* in Fig. 4 will monitor the executing status of application components. When Restore Manager detects that a machine belonged to the applications was fail, depending on the type of the machine, Restore Manager will launch an appropriate process of restoring.

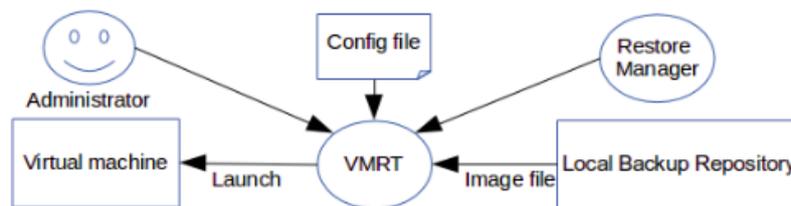


Fig. 4. Restore a virtual machine from Local backup repository.

This process is started when Restore Manager calls *Virtual Machine Restore Tool (VMRT)* with the identifier of the machine that need to be restored. VMRT will contact Local Backup Repository to find out the image file of the machine and launch an instance of this image file. If this is an application machine, the process of restoring is finished. In case of a database machine, the *Rsync Client* on the machine is set to be launched automatically to connect to Rsync Server to restore Datalog files from the *DB backup directory* to DB Cached Directory as described in Fig. 5. Then *DB Restore Agent* will update the application database with Datalog files in the DB Cached Directory. If this is a file server machine, the Rsync Client on the machine is set to be launched automatically to connect to Rsync Server to copy files from *Directories of static resources* to *Directories of static resources*.

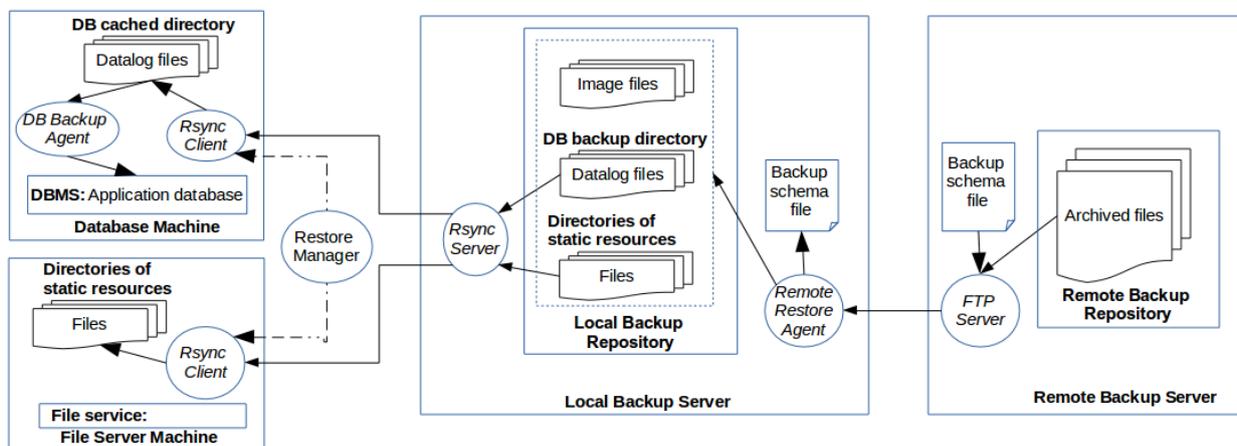


Fig. 5. Restore database and static resources of an EA from Local backup repository.

Mechanism for remotely restoring enterprise applications : Firstly, the administrator uses an FTP client to connect to FTP server on *Remote backup machine* and download image file of *Local backup server* to the cloud environment as described in Fig. 5. The administrator then launches Local backup machine from this image file. *Remote restore agent* will be executed automatically on Local backup machine. It connects to the FTP server on Remote backup machine and download *Backup schema file*. By analyzing the content of the file, Remote restore agent will download all backup files stored in *Remote backup repository*, decompress them to restore the content of Local backup repository. The administrator then executes *Application Restore Tool* to restore an EA from Local Repository, as described in Fig. 6. Based on the schema file, *Restore Application Tool* can know how many applications backed up in Local Backup Repository, how many VMs belonged to an application and which image file belongs to a VM. To restore an application, *Restore Application Tool* call *Virtual Machine Restore Tool (VMRT)* with identifiers of all machines belonged to that application.

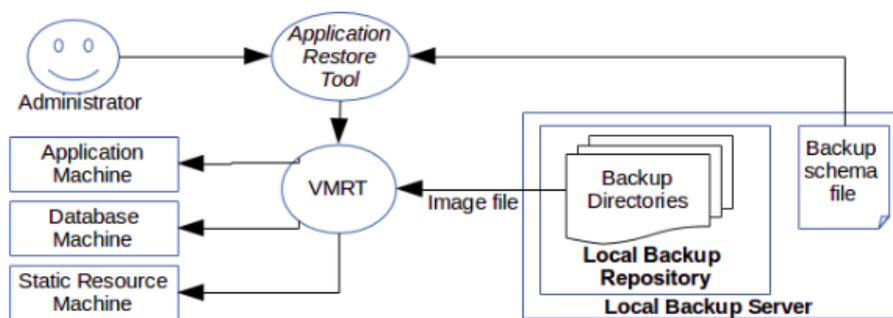


Fig. 6. Restoring an EA from Local backup repository.

4. Experimentation and Evaluation

We implemented the proposed framework for OpenStack cloud platform, version Liberty. A J2EE application developed on Liferay Platform was used to evaluate the framework. The application was deployed into 3 VMs. The application machine installed Tomcat 7.0. PostgreSQL was installed on the database machine. File server machine installed NFS server. On application machine, NFS Client was installed to mount directories of NFS server to directories that store static resource of the application. We used a VM and image service (Glance) of OpenStack to establish Local backup repository. Image files of VMs were stored in directories managed by image service. Backup directories for application and static resources component were managed by the filesystem of the VM. *Heat* and *Ceilometer*, two other services of OpenStack were exploited to create Backup Manager. *Backup WAL* supported by PostgreSQL was installed on the database machine to play the role of Data backup Agent. Backup WAL will create Datalog files with maximum size of 16 Mbytes. On Local backup server, *Rsync server* was installed. On database machine and file server machine, *Rsync client* was installed and configured to synchronize DB cached directories on database machine, static resource directories managed by the NFS server with backup directories on Local backup repository. The *VM backup tool* is a script file, named *vmbt.sh*, that receives in parameter a type of machine (application, database or fileserver) to be backed up, then it calls *nova client* command to create an image file of the corresponding machine. The image file as default is stored by Glance service. To evaluate the framework, firstly, the script *vmbt.sh* was called three times with the parameters for each time being application, database, fileserver. As a result, three images were created in Glance service. This proves that the application machines were backed up. Then, we accessed the testing application and used its functions. We found that Backup WAL began to create the first Datalog file. Because Rsync client on database machine was configured to check the directory of Datalog files every 15 minutes, the first Datalog file was found and

sent to Rsync Server by Rsync client. Rsync Server put the Datalog file in the database backup directory of Local backup server. We tried a function of the application that requires to upload a file into the application. After 15 minutes, we found this file in the static resource backup directory of Local backup server. These two test cases showed that the database and static resources of the application were backed up onto repository.

To evaluate the framework with the restore capabilities, script *vmrt.sh* was created. *Vmrt.sh* receives in its parameter the type of the machine (application, database or fileserver) that need to be launched. Then *vmrt.sh* calls *nova client* to launch an instance from a corresponding image file stored in Glance service. Heat and Ceilometer were configured to detect the failure or not running state of the three VMs deploying the application. When Ceilometer detects an application machine is not active, Heat will call *vmrt.sh* with application in parameter to relaunch the application. The same scenario was defined for database and file server machine.

Then we tried to terminate application machine. Verifying the running of the machine we found that the application machine was shut down and disappeared in several minutes. After that, it was relaunched. We accessed the application again and we found that the application was still functioning. This test case was also applied for database machine and static resource machine. Both machines were relaunched after several minutes. We verified the data that were put into the application and the file that were uploaded in previous test. We found that they were all existed in the application. Remember that, *vmrt.sh* launched data machine and static resource machine from images that did not have any data and files inputted or uploaded by us. So, the existing of data and uploaded file in the application proved that mechanisms of restoring the database and static resources of the application functioned well.

To evaluate the mechanism of restoring a whole EA, the script *art.sh* was created. *Art.sh* calls *vmrt.sh* to restore 3 VMs that host the application. We started the test by manually terminating the three running machines of the application. Waiting until three machines were well terminated, we executed *art.sh* script. We then verified the function of the application and found that the application were well restored.

5. Conclusion and Future Works

Backup and restore are essential operations to ensure the continuous running of an application and prevent the loss of data in case failure of hardware. This paper presented a framework for automating operations of backing up and restoring EAs that deployed in a cloud environment. The proposed framework is based on 3 principal mechanisms: mechanism of backing up and restoring VMs hosting EAs, mechanism of backing up and restoring a database of EAs and mechanism of backing up and restoring static resources of EAs. With proposed framework, all components of an EA are backed up in a consistent manner. Backed up data are not only stored on a local repository located in the same location with the application, but also stored in a remote repository located at a location far away from the application site. Restore mechanism proposed by the framework can detect and restore automatically an application component that was fail to ensure the application continue to function. The experiment proved that the proposed framework is feasible. However, the implement should be improved so that it is easier to integrate the backup and restore mechanism proposed by the framework into an EA.

References

- [1] Fowler, M., Rice, D., Foemmel, M., Hieatt, E., Mee, R., & Stafford, R. (2002). *Patterns of Enterprise Application Architecture*. Addison Wesley.
- [2] Badhel, S. P., & Chole, V. (2014). A review on data back-up techniques for cloud computing. *IJCSMC*, 3.
- [3] Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing, Recommendations of the National*

Institute of Standards and Technology. NIST Special Publication, 800-145.

- [4] Staimer, M. Storage snapshot technologies in data backup and recovery. From <http://searchdatabackup.techtarget.com/tip/Storage-snapshot-technologies-in-data-backup-and-recovery>
- [5] Preston, W. C. Using snapshot backups to replace your traditional data backup system. From <http://searchstorage.techtarget.co.uk/feature/Using-snapshot-backups-to-replace-your-traditional-data-backup-system>
- [6] Javaraiah, V. (2011). Backup for cloud and disaster recovery for consumers and SMBs. *Proceedings of IEEE 5th International Conference on Advanced Networks and Telecommunication Systems (ANTS)* (pp. 1-3).
- [7] Akkuş, I. E., & Goel, A. (2010). Data recovery for web applications. *Proceedings of IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*.
- [8] Sharma, K. & Singh, K. R. (2012). Online data back-up and disaster recovery techniques in cloud computing: A review. *International Journal of Engineering Research and Applications*, 2(5).
- [9] Vora, S. B., & Anandache, J. G. (2014). Data backup on cloud computing technology in digital libraries perspective. *Journal of Global Research in Computer Science*, 5(2).



Hung Ba Ngo is a lecturer of Cantho University, Vietnam, he is working at the College of Information and Communication Technology. He got a master's degree in informatics specializing in communication systems from IFI (Institut de la Francophonie pour L'informatique) in 1998. Then he got his PhD degree from INT (Institut National des Telecommunications), France in 2009 for his work on context-based model for personal file retrieval. He teaches and does the research mainly in the domain of computer networks, distributed systems, cloud and big data computing.



Cong Tru Huynh is a master student in information system at Cantho University. He is now preparing his master thesis, which subject is "Finding a framework for backing up and restoring Cantho City public administration applications deployed in cloud environment." He used to work as a network administrator for the Nguyen Ngoc Thach College at Cantho City, Vietnam.