International Journal of Interdisciplinary Innovative Research &Development (IJIIRD)
ISSN: 2456-236X
Vol. 05 Issue 02 | 2021

Implementing live Migration on Amazon S3

[1]Mr. Shashank Reddy S, [2]Dr Gangothri Rajaram

[1]MCA Scholar, School of CS & IT, Dept of MCA, Jain (Deemed-to-be) University, Bengaluru
[2]Asst Professor, School of CS & IT, Dept of MCA, Jain (Deemed-to-be) University, Bengaluru

ABSTRACT

Cloud computing has been used almost everywhere now, whereas the number of cloud users has increased. Cloud users may use live migrate their virtual machines from public cloud providers to another because of lower cost, it isn't possible to install another virtualization platform on a public cloud because nested and hardware virtualization is disabled by default. Virtual machines are tightly coupled to the cloud providers. [1] Live Cloud is mainly designed to live migration of VM in cloud infrastructure with slight changes to the services that are hosted in these VM. we are going to Implement Live Cloud in Amazon S3 where it can be optimized as a high-Performance processor. Where it mainly focuses on transferring files from cloud to another cloud, cloud to local, and local to cloud transferring of files.

Keywords: Live migration, Virtual Machine, Amazon S3

1. INTRODUCTION

Cloud computing provides many services and resources through the Internet these resources contain data storage, servers, software. [2]Whereas cloud-based storage makes it possible to save files to a remote database and use those files when it is in need. There are different types of cloud services: SaaS, IaaS, PaaS.

[3]Virtualization on a cloud allows the users to explore more resources and helps to secure the cloud infrastructure and the public cloud which even has a mixed environment with more vendors due to this the cloud providers have their own emulator. these cloud providers have their own hypervisor
Boto is one of the amazon services and an SDK for python that helps python users to create, configure, and even manage services like S3. Boto provides API and low-Level access to the services like AWS.

Live Cloud migration helps in migrating the already existing virtual machines without disconnecting the application. [4] The client can share their files or data that they need and monitor the total migration time. Migrating to local drives and then to the cloud, migrating cloud to local and local to the cloud.

1.1 Motivation

[5]The main reason for making this project is to review the VM migration scheme considering the several aspects in migrating the virtual machines which can show the highlights of the total migration of an application and offline migration considering these several applications can be migrated without stopping the VM from one VM to another using amazon S3 this scheme helps in fast processing and it is not time consuming

1.2 Literature Review

[4]In this paper, the author writes about the techniques that have been implemented in cloud migration and data migration. In this, they have discussed about decouple of virtual machines from hypervisor a migrate those VM with low-cost services. The user criteria for live clod migration are to have flexibility, performance, and security. Live Cloud has Certain limitations and in live Cloud, we have User Datagram Protocol- based transfer (UDP). Software-defined network (SND).

2. REQUIREMENTS

This project contains both hardware and software requirements.
Hardware requirements:

- Processor above 500 MHz
- RAM: 4 GB
- Hard Disk: 4 GB
- Input Device: Standard Keyboard and mouse
- Output Device: High Resolution monitor

Software requirements:

- Anaconda
- Python 3.9
- AWS

3. METHODOLOGIES

The proposed work is implemented in Python 3.6.4 with libraries Tkinter, boto3, matplotlib and other mandatory libraries. We created credential with Amazon S3. The data can be migrated from one cloud to other cloud, local drive to cloud, cloud to local drive depending on user requirements. We have migrated any type of file from cloud or from local drive. This is facilitated by boto3 client.

3.1 Authentication and instantiation

Amazon web service (AWS) is used for our proposed work. We have created the account and extracted access_id and access_key from the dashboard. The access_id and access_key are used for creating connection to aws using boto3 library. The following code is used to create the authentication and gets all available buckets from the resource.

```
filename=d+"/"+filename

s3.Bucket(BUCKET_NAME).put_object(Key=filename, Body=data)
```

The following lines of code created and upload file to cloud. Here boto3's put_object is used to download data object to cloud.

```
path='sourcedownloads/'+FILE_NAME
s3.Bucket(BUCKET NAME).download file(FILE NAME, path);
```

3.2 Individual File migration

Individual file upload from local drive to cloud, file download from cloud to local drive and individual file migration from one bucket to the other bucket is handled. The following lines of code created and upload file to cloud. Here boto3's put_object is used to download data object to cloud.

```
filename=d+"/"+filename

s3.Bucket(BUCKET_NAME).put_object(Key=filename, Body=data)
path='sourcedownloads/'+FILE_NAME
s3.Bucket(BUCKET_NAME).download_file(FILE_NAME, path);
```

3.3 Migration using local disk

The following code is used for migration of data from cloud to local drive. Here we have created a folder "migrate" in local drive, to which all files are moved from cloud server.

```
path='migrate/'+FILE_NAME

s3.Bucket(BUCKET_NAME).download_file(FILE_NAME, path);

ACCESS_KEY_ID1 = sym4
ACCESS_SECRET_KEY1 = sym5
BUCKET_NAME1 = txt7.get()
s3 =
boto3.resource('s3',aws_access_key_id=ACCESS_KEY_ID1,aws_secret_access
_key=ACCESS_SECRET_KEY1,config=Config(signature_version='s3v4'))
```

3.4 Live migration

Live data migration from one bucket to another bucket is handled using the following codes.

```
s3.Bucket(BUCKET_NAME).download_file(FILE_NAME, path);
                    ACCESS_KEY_ID1 = sym4
                    ACCESS_SECRET_KEY1 = sym5
                    BUCKET_NAME1 = txt7.get()
                    s3 =
boto3.resource('s3',aws_access_key_id=ACCESS_KEY_ID1,aws_secret_access
_key=ACCESS_SECRET_KEY1,config=Config(signature_version='s3v4'))
                    data = open(path, 'rb')
    s3.Bucket(BUCKET NAME1).put object(Key=FILE NAME, Body=data)
```

Here we do, connect the resource1 and resource 2, then downloading path of Bucket1 is given and uploading path of Bucket 2 is given, live migration of data occurs.

4. PROPOSED SYSTEM

The proposed system should be able to migrate data from the cloud to another VM/cloud. It is designed in such a way that the live migration of virtual machines in IaaS doesn't cause much service interruption.
The overview of architecture design –

The user authenticates to the cloud server and the next user performs live migration or migrating to the local disk and from that local disk to the cloud

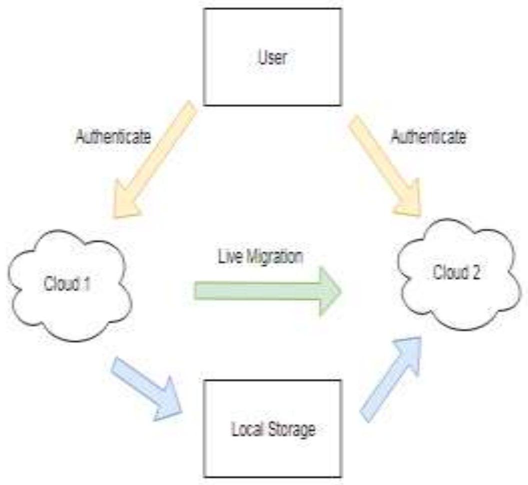


Fig 1.1 Architecture of the system

The flow of control in the system- The user has created two amazon S3 bucket where each bucket contains a certain amount of data. user have created an access key which is used to connect to the amazon services then after establishing the connection. The user will now access the VM's that are created and check the data that are present in them, now the user can share the data between two VM's and can transfer the files between then, can upload the data to the VM's using this the user can easily migrate the data that is present in the VM's, deletion of the data can also be done using API, the user can also check the time take to migrate the data from one cloud to another using Live and Local drives and migrating clou to cloud.

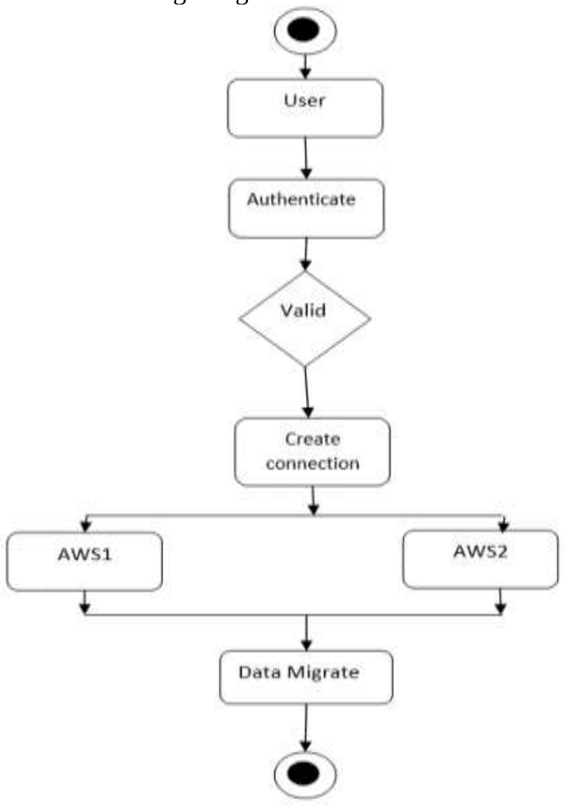
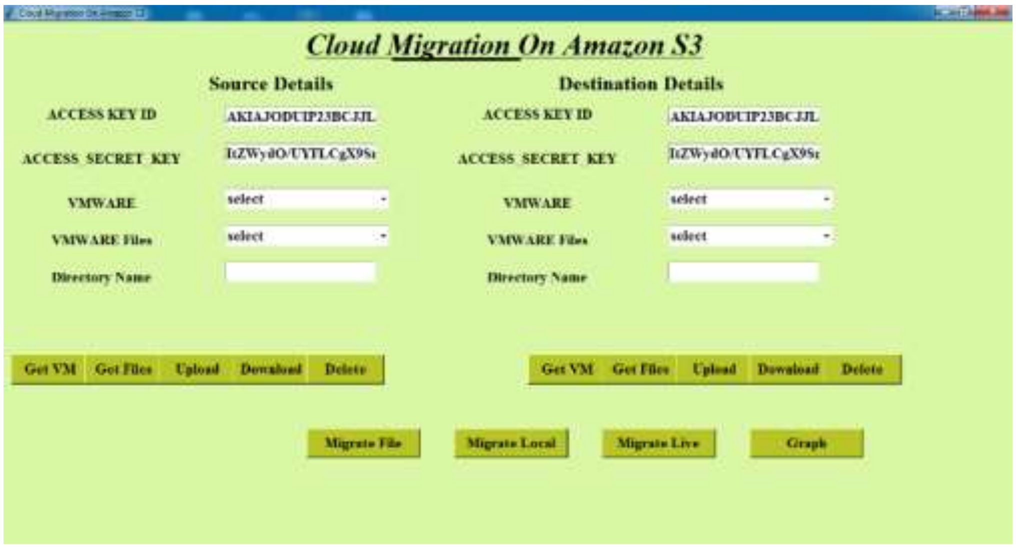


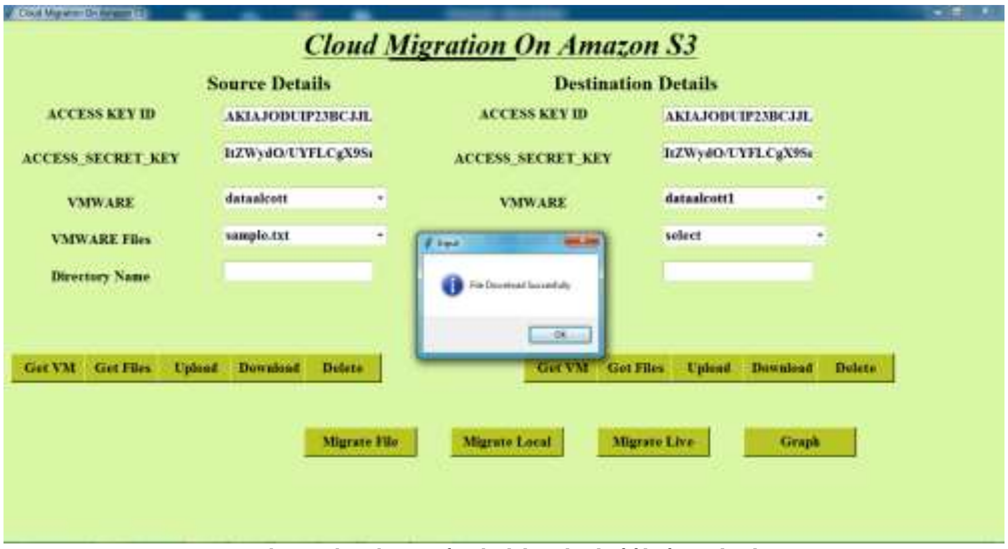
Fig 1.2 Flow of control in the system

5. RESULT


Cloud Migration On Amazon S3

Source Details		Destination Details	
ACCESS KEY ID	AKIAJODUIP23BCJJL	ACCESS KEY ID	AKIAJODUIP23BCJJL
ACCESS_SECRET_KEY	ltZWydO/UYFLCgX9St	ACCESS_SECRET_KEY	ltZWydO/UYFLCgX9St
VMWARE	select	VMWARE	select
VMWARE Files	select	VMWARE Files	select
Directory Name		Directory Name	

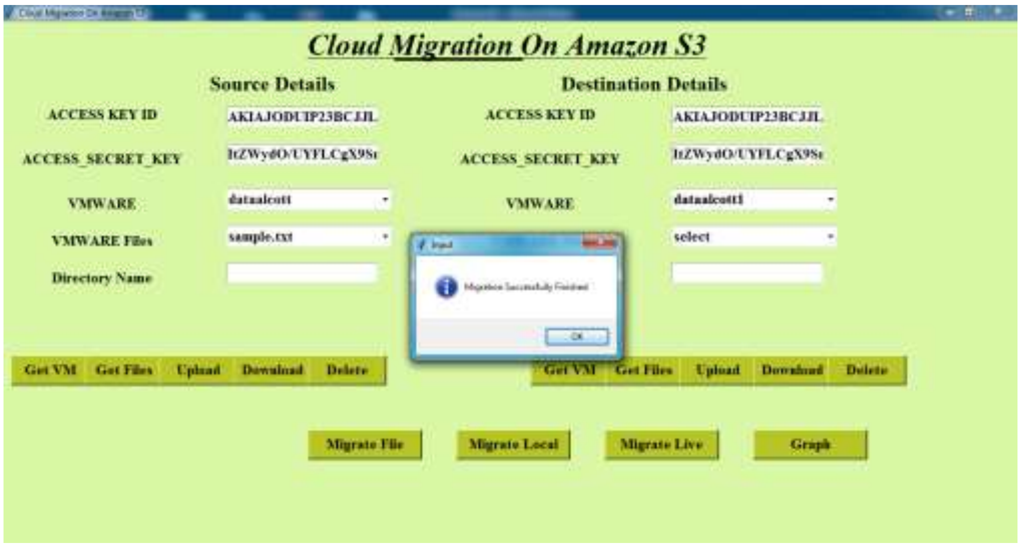
Fig 1.3 The following alert screen shown to user after successful Resource (Amazon Web Service) connection is made.


Cloud Migration On Amazon S3

Source Details		Destination Details	
ACCESS KEY ID	AKIAJODUIP23BCJJL	ACCESS KEY ID	AKIAJODUIP23BCJJL
ACCESS_SECRET_KEY	ltZWydO/UYFLCgX9St	ACCESS_SECRET_KEY	ltZWydO/UYFLCgX9St
VMWARE	dataalcott	VMWARE	dataalcott1
VMWARE Files	sample.txt		select
Directory Name			

Input
File Download Successfully
OK

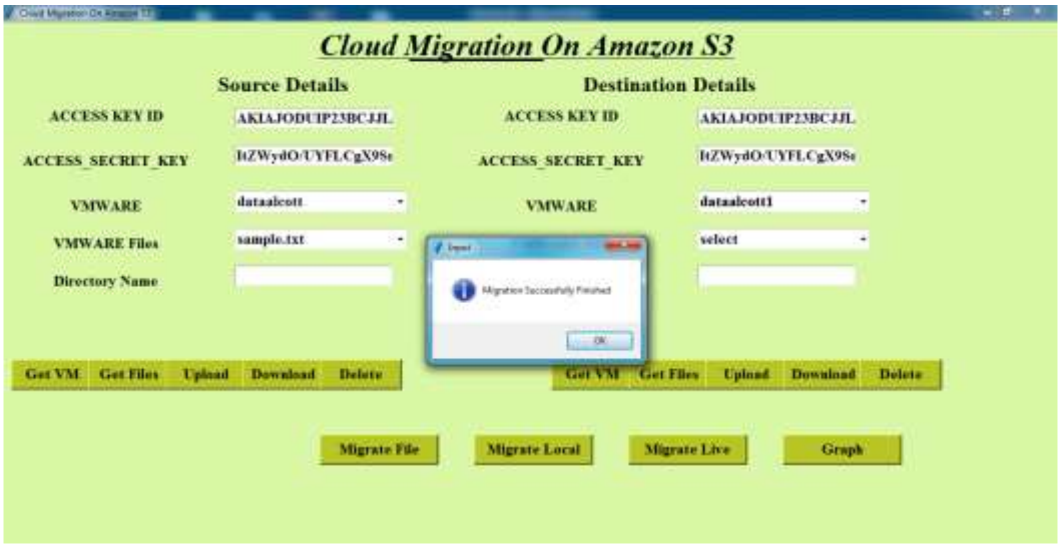
Fig 1.4 Shows when the user finished download of file from cloud server


Cloud Migration On Amazon S3

Source Details		Destination Details	
ACCESS KEY ID	AKIAJODUIP23BCJJL	ACCESS KEY ID	AKIAJODUIP23BCJJL
ACCESS_SECRET_KEY	ltZWydO/UYFLCgX9St	ACCESS_SECRET_KEY	ltZWydO/UYFLCgX9St
VMWARE	dataalcott	VMWARE	dataalcott1
VMWARE Files	sample.txt		select
Directory Name			

Input
Migration Successfully Finished
OK

Fig 1.5 The following screen shows the alert to user once the Migration of data is completed for (Cloud-local-cloud)



Cloud Migration On Amazon S3

Source Details

ACCESS KEY ID: AKIAJODUIP23BCJJL
ACCESS_SECRET_KEY: ItZWydO/UYFLCgX9Se
VMWARE: dataalcott
VMWARE Files: sample.txt
Directory Name:

Destination Details

ACCESS KEY ID: AKIAJODUIP23BCJJL
ACCESS_SECRET_KEY: ItZWydO/UYFLCgX9Se
VMWARE: dataalcott1
VMWARE Files: select

Migration Successfully Finished

Fig 1.5 screen shows the alert to user once the Live Migration of data is completed

5.1 Graph

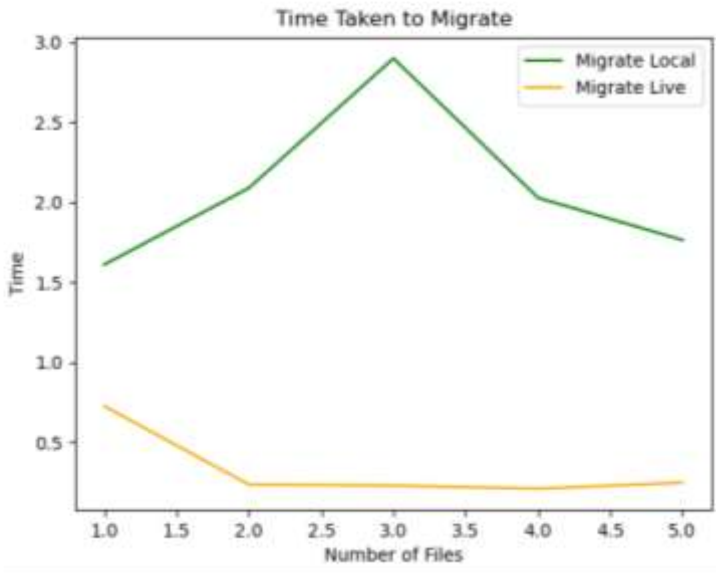


Fig 1.6 The following graph shows the time taken in seconds for data migration from one cloud to the other using Live and Local drive

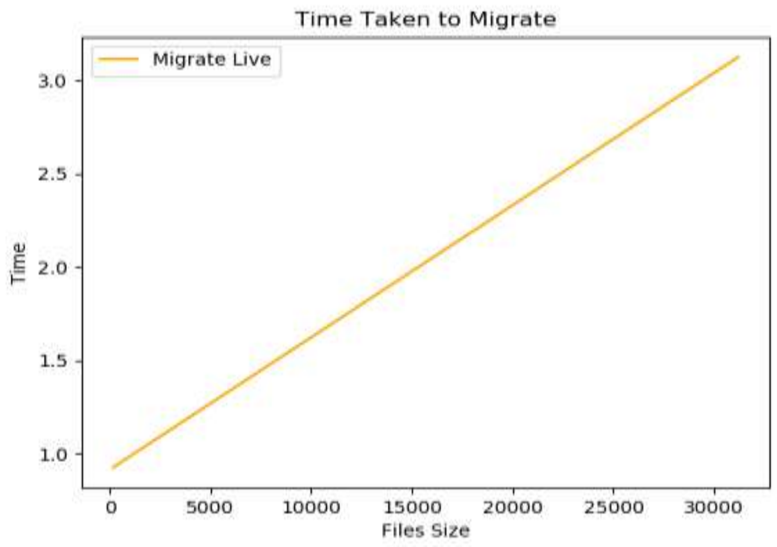


Fig 1.7 The following graph shows the time taken in seconds for Live data migration from one cloud to the other.

6. CONCLUSION

Given the current state of public cloud IaaS in terms of hardware-assisted virtualization features, VMs live migration is still challenging to cloud users. The proposed approach is introduced to help successfully live migrate cloud users' VMs without services disruption across different public cloud providers. The basic design stage of this approach is implemented and evaluated on Amazon S3 instance. The connectivity is securely maintained between Local-Host and Cloud-Host through boto3 library. The experiments are performed for migrating files live and from local disk. The experimental results are discussed.

7. REFERENCES

[1] I. E. A. Mansour, H. Bouchachia and K. Cooper, "Exploring Live Cloud Migration on Amazon EC2," 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), no. 10.1109/FiCloud.2017.20, p. 8, 2017.

[2] A. V. E. Toby Velte, "Cloud Computing, A Practical Approach," no. 978-0-07-162694-1, p. 352, 2009.

[3] R. W. G. A. A. H. S. H. S. M. X. F. M. S. A. Ahmad, "Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues," The Journal of Supercomputing, vol. 71, no. 10.1007/s11227-015-1400-5, 2015.

[4] P. G. J. Leelipushpam and J. Sharmila, "Live VM migration techniques in cloud environment," 2013 IEEE Conference on Information & Communication Technologies, no. 10.1109/CICT.2013.6558130, 2013.

[5] A. A. P. Pooyan Jamshidi, "Cloud Migration Research: A Systematic Review," IEEE Transactions on Cloud Computin, vol. 1, no. 10.1109/TCC.2013.10, 2014.