ORIGINAL RESEARCH PAPER

Computer Science

DESIGN & IMPLEMENTATION OF CLOUD COMPUTING SERVICES USING PYTHON

KEY WORDS: Cloud Computing, Cloud Server, Cloud Computing Services, Python, Encryption, Virtualization

Mr. Chandrakant R. Gujar Assistant Professor, W.K.B.S.Mandals Dr. Suryakanta R. Ajmera MCA College for Women, Dhule (MS) India

ABSTRACT

Cloud computing can be defined as a collection of hosted services provided to users over the Internet. It enables organizations to consume or even compute the resource, which includes Virtual Machines , storage, or an application as a utility. One of the most important advantages of design and implementation of cloud computing services using Python programming language is that it includes the ability to deploy applications virtually on any platform, which includes cloud as well. It implies that Python can be executed on cloud servers and can also be launched on handy devices such as desktop, tablet, or smart phone.

INTRODUCTION

The term **cloud** refers to a network or the internet. It is a technology that uses remote servers on the internet to store, manage, and access data online rather than local drives. The data can be anything such as files, images, documents, audio, video, and more.

There are the following operations that we can do using cloud computing:

- Developing new applications and services
- Storage, back up, and recovery of data
- Hosting blogs and websites
- Delivery of software on demand
- Analysis of data
- Streaming videos and audios

Cloud Server:

A cloud server is a virtual server (rather than a physical server) running in a cloud computing environment. It is built, hosted and delivered via a cloud computing platform via the internet, and can be accessed remotely. They are also known as virtual servers. Cloud servers have all the software they require to run and can function as independent units.

Why Cloud Computing?

Small as well as large IT companies, follow the traditional methods to provide the IT infrastructure. That means for any IT company, we need a Server Room that is the basic need of IT companies.

In that server room, there should be a database server, mail server, networking, firewalls, routers, modem, switches, QPS (Query Per Second means how much queries or load will be handled by the server), configurable system, high net speed, and the maintenance engineers.

To establish such IT infrastructure, we need to spend lots of money. To overcome all these problems and to reduce the IT infrastructure cost, Cloud Computing comes into existence.

Components of Cloud Computing Architecture

There are the following components of cloud computing architecture -

1. Client Infrastructure

Client Infrastructure is a Front end component. It provides GUI (Graphical User Interface) to interact with the cloud.

2. Application :

The application may be any software or platform that a client wants to access.

3. Service :

A Cloud Services manages that which type of service you access according to the client's requirement.

Cloud computing offers the following three type of services:

I. Software as a Service (SaaS) –

It is also known as **cloud application services**. Mostly, SaaS applications run directly through the web browser means we do not require to download and install these applications. Some important example of SaaS is given below –

Example: Google Apps, Sales force Drop box, Slack, Hub spot, Cisco Web Ex.

ii. Platform as a Service (PaaS) –

It is also known as **cloud platform services**. It is quite similar to SaaS, but the difference is that PaaS provides a platform for software creation, but using SaaS, we can access software over the internet without the need of any platform.

Example: Windows Azure, Force.com, Magento Commerce Cloud, Open Shift.

iii. Infrastructure as a Service (IaaS) –

It is also known as **cloud infrastructure services**. It is responsible for managing applications data, middleware, and runtime environments.

Example: Amazon Web Services (AWS) EC2, Google Compute Engine (GCE), Cisco Metapod.

4. Runtime Cloud :

Runtime Cloud provides the **execution and runtime environment** to the virtual machines.

5. Storage :

Storage is one of the most important components of cloud computing. It provides a huge amount of storage capacity in the cloud to store and manage data.

6. Infrastructure :

It provides services on the **host level**, **application level**, and **network level**. Cloud infrastructure includes hardware and software components such as servers, storage, network devices, virtualization software, and other storage resources that are needed to support the cloud computing model.

7. Management :

Management is used to manage components such as application, service, runtime cloud, storage, infrastructure, and other security issues in the backend and establish coordination between them.

8. Security :

Security is an in-built back end component of cloud computing. It implements a security mechanism in the back end.

9. Internet :

The Internet is medium through which front end and back end can interact and communicate with each other.

Python:
Python is a general-purpose programming language with easy, readable code that can be easily understood by both professional developers as well as novice programmers. Python comprises of many useful libraries that can be used with any stack framework. Many laboratories rely on Python to build basic models for predictions and to run experiments. It also helps to control critical operational systems.

Python has built-in capabilities to support digital investigation and protect the integrity of evidence during an investigation. In this tutorial, we will explain the fundamental concepts of applying Python in digital or computation forensics.

The codes written in Python look quite similar to the codes written in other conventional programming languages such as C or Pascal. It is also said that the syntax of Python is heavily borrowed from C. This includes many of the Python keywords which are similar to C language.

Cracking of Encryption Using Python:
A plain text in cryptography is some normal readable text, such as a message. A cipher text, on the other hand, is the output of an encryption algorithm fetched after you enter plain text.

Simple algorithm of how we turn a plain text message into a cipher text is the Caesar cipher, invented by Julius Caesar to keep the plain text secret from his enemies. This cipher involves shifting every letter in the message "forward" by three places in the alphabet.

Following is a demo illustration.
a →D
b → E
c →F
....
w→ Z
x →A
y→ B
z→ C

Example
A message entered when you run a Python script gives all the possibilities of characters, which is used for pattern evidence. The types of pattern evidences used are as follows −

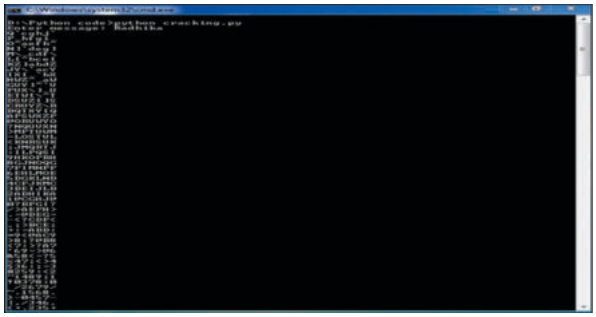
- Tire Tracks and Marks
- Impressions
- Fingerprints

Every biometric data comprises of vector data, which we need to crack to gather full-proof evidence.

The following Python code shows how you can produce a cipher text from plain text −

```
import sys

def decrypt(k,cipher):
  plaintext = ''

for each in cipher:
p = (ord(each)-k) % 126

if p < 32:
p+=95
plaintext += chr(p)
print plaintext

def main(argv):
if (len(sys.argv) != 1): sys.exit('Usage: cracking.py')
cipher = raw_input('Enter message: ')
for i in range(1,95,1):
decrypt(i,cipher)

if __name__ == "__main__":
  main(sys.argv[1:])
```

Output
Now, check the output of this code. When we enter a simple text "Radhika", the program will produce the following cipher text.



Virtualization in Cloud Computing
Virtualization is the "creation of a virtual (rather than actual) version of something, such as a server, a desktop, a storage device, an operating system or network resources".In other words, Virtualization is a technique, which allows to share a single physical instance of a resource or an application among multiple customers and organizations. It does by assigning a logical name to a physical storage and providing a pointer to that physical resource when demanded.

What is the concept behind the Virtualization?
Creation of a virtual machine over existing operating system and hardware is known as Hardware Virtualization. A Virtual machine provides an environment that is logically separated from the underlying hardware.

Types of Virtualization:
1. Hardware Virtualization.
2. Operating system Virtualization.
3. Server Virtualization.
4. Storage Virtualization.

1) Hardware Virtualization:
When the virtual machine software or virtual machine manager *(VMM) is directly installed on the hardware system* is known as hardware virtualization. The main job of hypervisor is to control and monitoring the processor, memory and other hardware resources.

After virtualization of hardware system we can install different operating system on it and run different applications on those OS.

Usage:
Hardware virtualization is mainly done for the server platforms, because controlling virtual machines is much easier than controlling a physical server.

2) Operating System Virtualization:
When the virtual machine software or virtual machine manager *(VMM) is installed on the Host operating system* instead of directly on the hardware system is known as operating system virtualization.

Usage:
Operating System Virtualization is mainly used for testing the applications on different platforms of OS.

3) Server Virtualization:
When the virtual machine software or virtual machine manager *(VMM) is directly installed on the Server system* is

known as server virtualization.

Usage:
Server virtualization is done because a single physical server can be divided into multiple servers on the demand basis and for balancing the load.

4) Storage Virtualization:
Storage virtualization is the *process of grouping the physical storage from multiple network storage devices so that it looks like a single storage device*.

Storage virtualization is also implemented by using software applications.

Usage:
Storage virtualization is mainly done for back-up and recovery purposes.

How does virtualization work in cloud computing?
Virtualization plays a very important role in the cloud computing technology, normally in the cloud computing, users share the data present in the clouds like application etc, but actually with the help of virtualization users shares the Infrastructure.

The **main usage of Virtualization Technology** is to provide the applications with the standard versions to their cloud users, suppose if the next version of that application is released, then cloud provider has to provide the latest version to their cloud users and practically it is possible because it is more expensive.

To overcome this problem we use basically virtualization technology, By using virtualization, all severs and the software application which are required by other cloud providers are maintained by the third party people, and the cloud providers has to pay the money on monthly or annual basis.

CONCLUSION
Mainly Virtualization means, running multiple operating systems on a single machine but sharing all the hardware resources. And it helps us to provide the pool of IT resources so that we can share these IT resources in order get benefits in the business.

The following example helps in understanding the creation of a virtual machine with the help of Python programming language.

Step 1 –Let the virtual machine be named 'dummy1'.
Every virtual machine must have 512 MB of memory in minimum capacity, expressed in bytes.
vm_memory = 512 * 1024 * 1024

Step 2 – The virtual machine must be attached to the default cluster, which has been calculated.
vm_cluster = api.clusters.get(name = "Default")

Step 3 – The virtual machine must boot from the virtual hard disk drive.
vm_os = params.OperatingSystem(boot = [params.Boot(dev = "hd")])

All the options are combined into a virtual machine parameter object, before using the add method of the vms collection to the virtual machine.

Example :
Following is the complete Python script for adding a virtual machine.

```
from ovirtsdk.api import API #importing API library
from ovirtsdk.xml import params

try: #Api credentials is required for virtual machine
api = API(url = "https://HOST",
username = "Radhika",
password = "a@123",
ca_file = "ca.crt")

vm_name = "dummy1"
vm_memory = 512 * 1024 * 1024 #calculating the memory in bytes
vm_cluster = api.clusters.get(name = "Default")
vm_template = api.templates.get(name = "Blank")

#assigning the parameters to operating system
vm_os = params.OperatingSystem(boot = [params.Boot(dev = "hd")])

vm_params = params.VM(name = vm_name,
memory = vm_memory,
cluster = vm_cluster,
template = vm_template
os = vm_os)

try:
api.vms.add(vm = vm_params)
print "Virtual machine '%s' added." % vm_name #output if it is successful.
except Exception as ex:
print "Adding virtual machine '%s' failed: %s" % (vm_name, ex)
api.disconnect()

except Exception as ex:
print "Unexpected error: %s" % ex
```

Output
Our code will produce the following output –

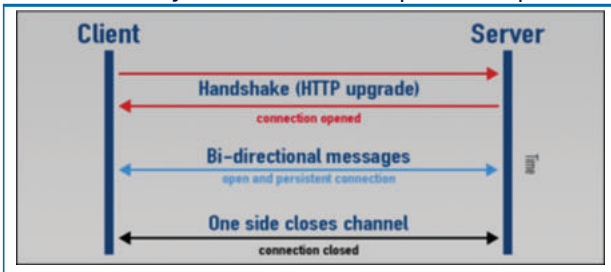
Concept of Network Programming Using Python
The following definitions are used in network programming.
Client –
Client is a part of client-server architecture of network programming which runs on a personal computer and workstation.

Server –
The server is a part of client-server architecture that provides services to other computer programs in the same or other computers.

WebSockets –
WebSockets provide a protocol between the client and the server, which runs over a persistent TCP connection. Through this, bi-directional messages can be sent between the TCP socket connection (simultaneously).

WebSockets come after many other technologies that allow the servers to send information to the client. Other than handshaking the Upgrade Header, WebSockets is independent from HTTP.

These protocols are used to validate the information which is sent or received by the third party users. As encryption is one of the methods used for securing messages, it is also important to secure the channel through which the messages have been transferred.



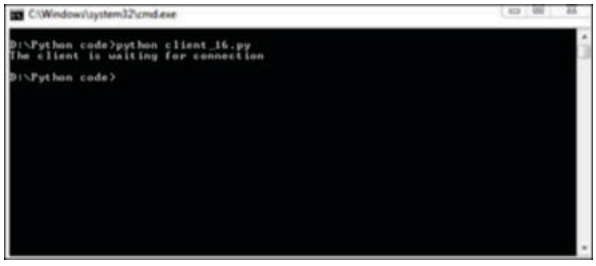
Consider the following Python program, which the client uses for **handshaking**.

Example

```
# client.py
import socket

# create a socket object
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# get local machine name
host = socket.gethostname()
port = 8080

# connection to hostname on the port.
s.connect((host, port))

# Receive no more than 1024 bytes
tm = s.recv(1024)
print("The client is waiting for connection")
s.close()
```

Output

It will produce the following output –



The server accepting the request for communication channel will include the following script.

```
# server.py
import socket
import time

# create a socket object
serversocket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

# get local machine name
host = socket.gethostname()
port = 8080

# bind to the port
serversocket.bind((host, port))

# queue up to 5 requests
serversocket.listen(5)

while True:
# establish a connection
clientsocket,addr = serversocket.accept()
print("Got a connection from %s" % str(addr))
currentTime = time.ctime(time.time()) + "\r\n"
clientsocket.send(currentTime.encode('ascii'))
clientsocket.close()
```

The client and server created with the help of Python programming listen to the host number.

Acknowledgments

We are thankful to our President West Khandesh Bhagini Seva Mandal's Mr. Atulji R. Ajmera, Mrs.MeghaV.Kamerkar OS, Director Dr. Rajeev B.Kharat, and all teaching staff /office staff of Dr. Suryakanta R. Ajmera MCA College for Women Deopur Dhule MS, India for motivation and encouragement which helped us to complete this research paper.

REFERENCES

[1] https://www.guru99.com/what-is-big-data.html#1
[2] https://www.tutorialspoint.com
[3] https://www.guru99.com/big-data-analytics-tools.html