

# JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY



## Remote Desktop Program Compilation

### SUBMITTED BY:-

ANANYAA AGARWALA - 13103606

NEETI KAVATHIA - 13103600

SANCHIT GARG - 13103609

CHETAN KANJANI - 13103618

BATCH: B6

### SUBMITTED TO:-

MRS. SANGEETA MITTAL

MRS. K. RAJALAXMI

MINOR LAB FACULTY

# **ABSTRACT**

With the increasing number of different platforms available in the market, it is very important for developers to test their application on multiple OS. Rather than buying different machines for different OS, we provide the developers with a virtual machine for testing their applications and also compare its performance on different OS. We also provide cloud storage.

# **OBJECTIVES**

- **Remote Compilation and Testing of different programming languages.**
- **Comparing their Performance on different OS (Virtual Ubuntu, Virtual Linux Mint, and Native Ubuntu).**
- **Storing files on Cloud (virtual machine).**

# **BACKGROUND STUDY AND FINDINGS**

**SSH**, also known as Secure Socket Shell, is a network protocol that provides administrators with a secure way to access a remote computer. Secure Shell provides strong authentication and secure encrypted data communications between two computers connecting over an insecure network such as the Internet. SSH is widely used by network administrators for managing systems and applications remotely, allowing them to log in to another computer over a network, execute commands and move files from one computer to another.

**Sockets** are interfaces that can "plug into" each other over a network. Once so "plugged in", the programs so connected communicate.

A "server" program is exposed via a socket connected to a certain /etc/services port number.

A "client" program can then connect its own socket to the server's socket, at which

time the client program's writes to the socket are read as stdin to the server program, and stout from the server program are read from the client's socket reads.

Before a user process can perform I/O operations, it calls Open to specify and obtain permissions for the file or device to be used.

Once an object has been opened, the user process makes one or more calls to Read or Write data.

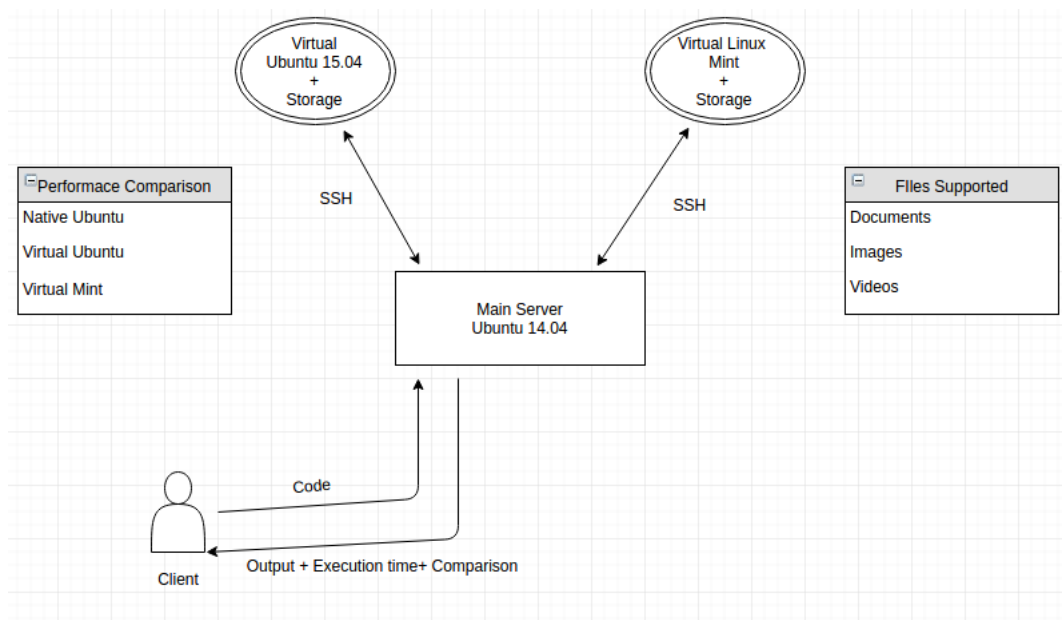
Read reads data from the object and transfers it to the user process, while Write transfers data from the user process to the object.

After all transfer operations are complete, the user process calls Close to inform the operating system that it has finished using that object

## **DESIGNING**

Our main server is based on Native Ubuntu which is hosting 2 virtual OS - Ubuntu and Linux Mint using Oracle VMware. The client makes a socket connection with our main server, selects the platforms it wants to test its program and sends the code to the main server. The main server then accesses the virtual servers using an SSH connection and forwards the code for compilation on the respective OS. The code is compiled and tested and the generated output and execution time are sent back to the main server. The output is then displayed to the user and a graph is plotted showing the execution times.

For using cloud storage, the users can browse any file on his machine and upload it to any of the virtual servers where a folder is created with his name containing all his files. Whenever he wants to download his files from the cloud, the main server returns all the user's files on different virtual servers which are displayed to him. The user can select whichever file he wants and it will be downloaded to his local machine.



## PARTIAL IMPLEMENTATION

```

# name=s.recv(1024)

import sys
from PySide.QtCore import *
from PySide.QtGui import *

class ExampleApp(QDialog):
    """ An example application for PyQt. Instantiate
    and call the run method to run. """
    def __init__(self):
        # create a Qt application --- every PyQt app needs one
        self.qt_app = QApplication(sys.argv)

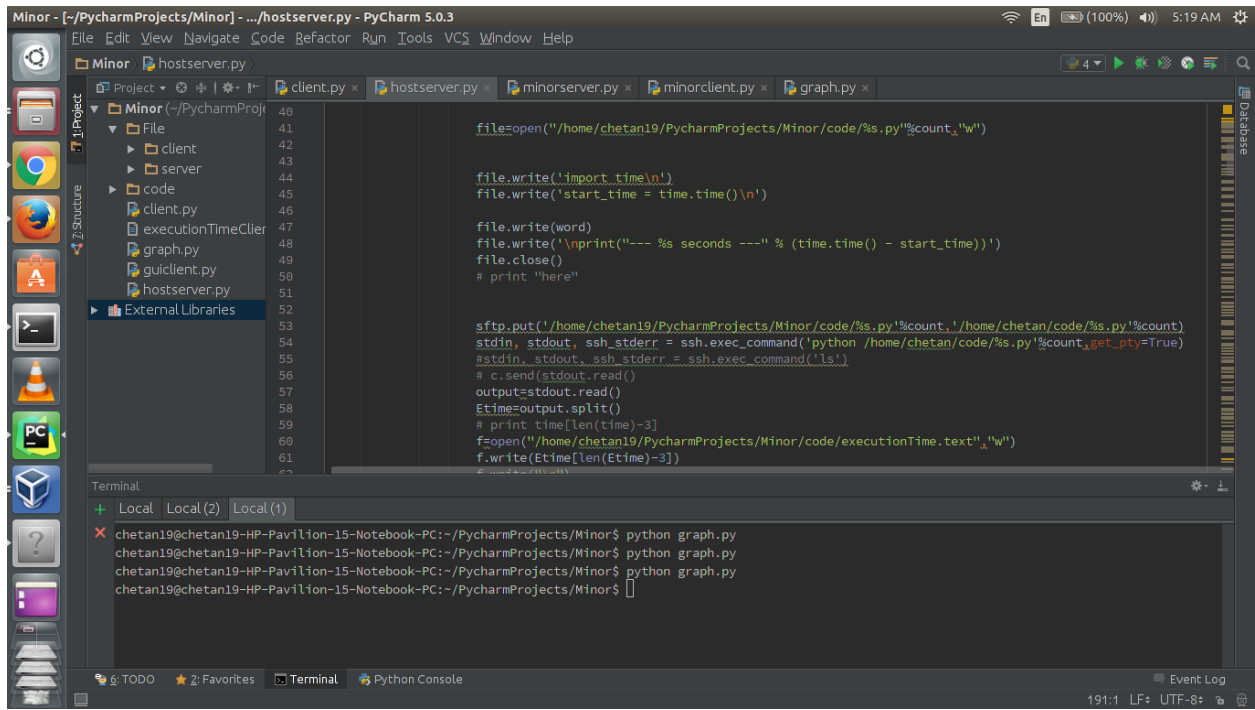
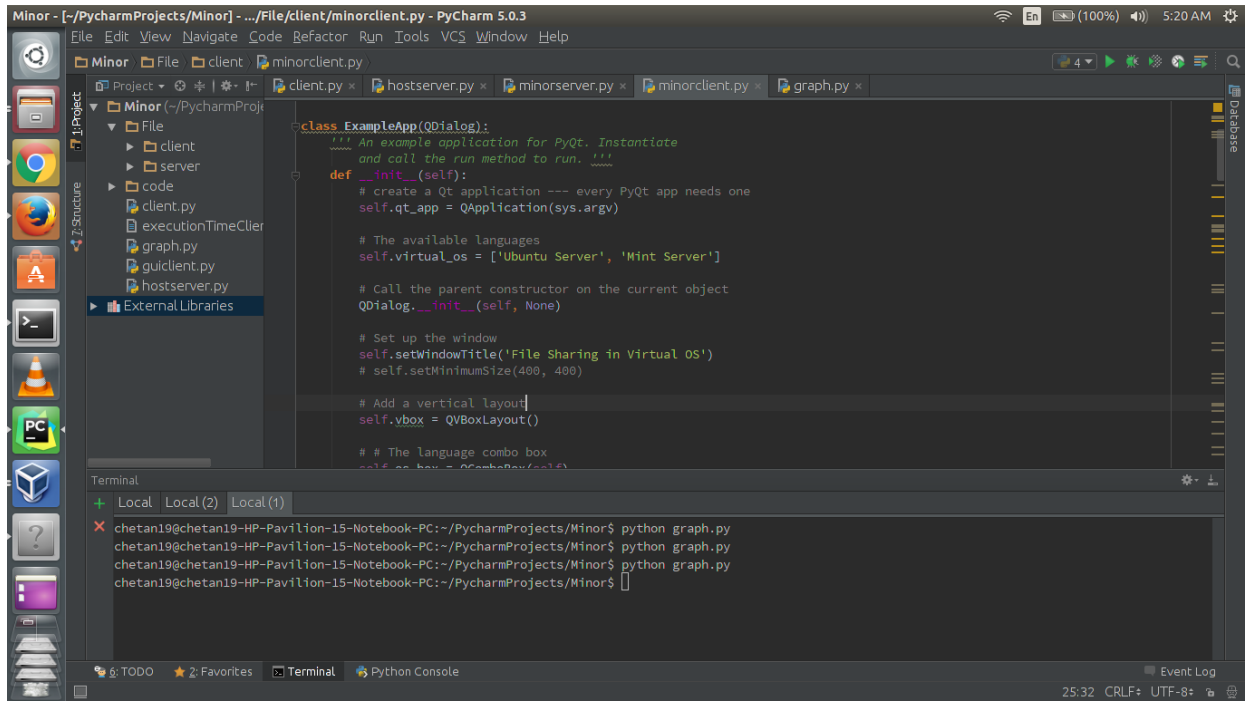
        # The available languages
        self.languages = ['C', 'C++', 'python']
        self.osOptions = ['ALL', 'Native Ubuntu', 'Virtual Ubuntu 15.04', 'Virtual Linux Mint']

        # Call the parent constructor on the current object
        QDialog.__init__(self, None)

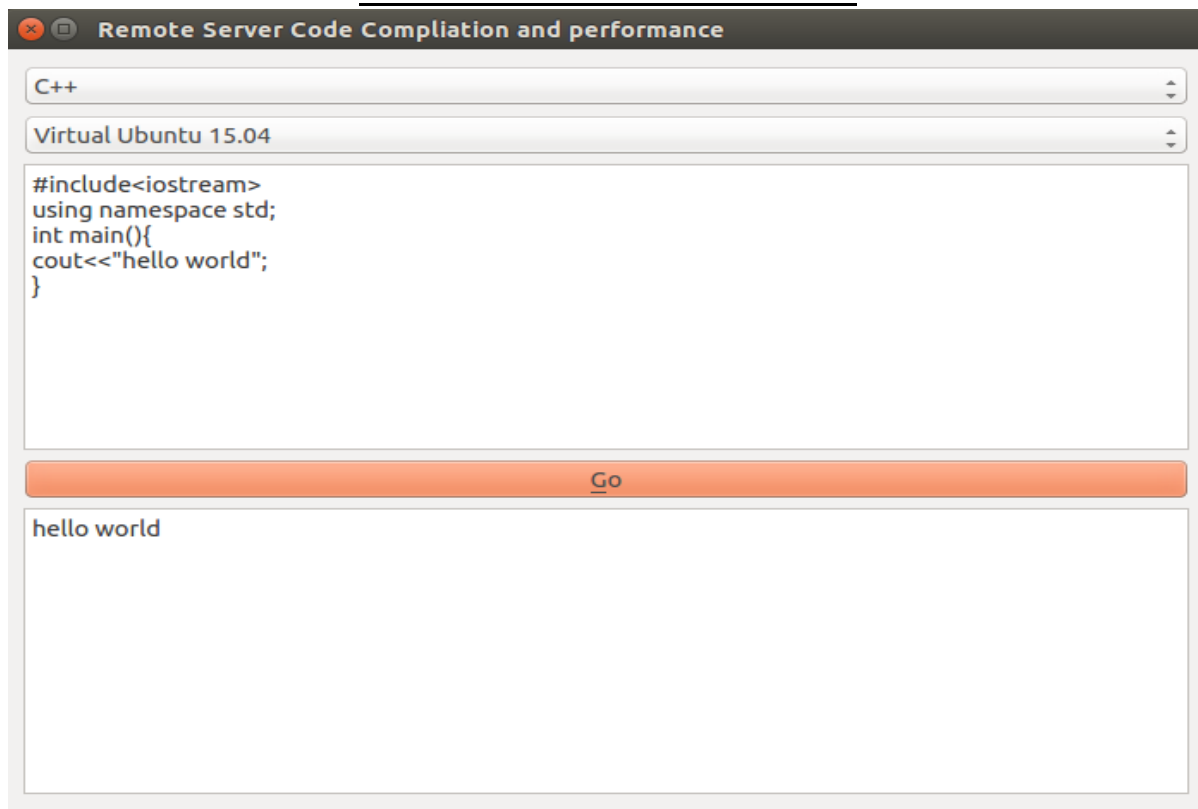
        # Set up the window
        self.setWindowTitle('Remote Server Code Compilation and performance')
        self.setGeometry(700, 400)

Terminal
+ Local Local(2) Local(1)
x chetan19@chetan19-HP-Pavilion-15-Notebook-PC:~/PycharmProjects/Minor$ python graph.py
chetan19@chetan19-HP-Pavilion-15-Notebook-PC:~/PycharmProjects/Minor$ python graph.py
chetan19@chetan19-HP-Pavilion-15-Notebook-PC:~/PycharmProjects/Minor$ python graph.py
chetan19@chetan19-HP-Pavilion-15-Notebook-PC:~/PycharmProjects/Minor$

```



# TESTING THE CODE



Remote Server Code Compilation and performance

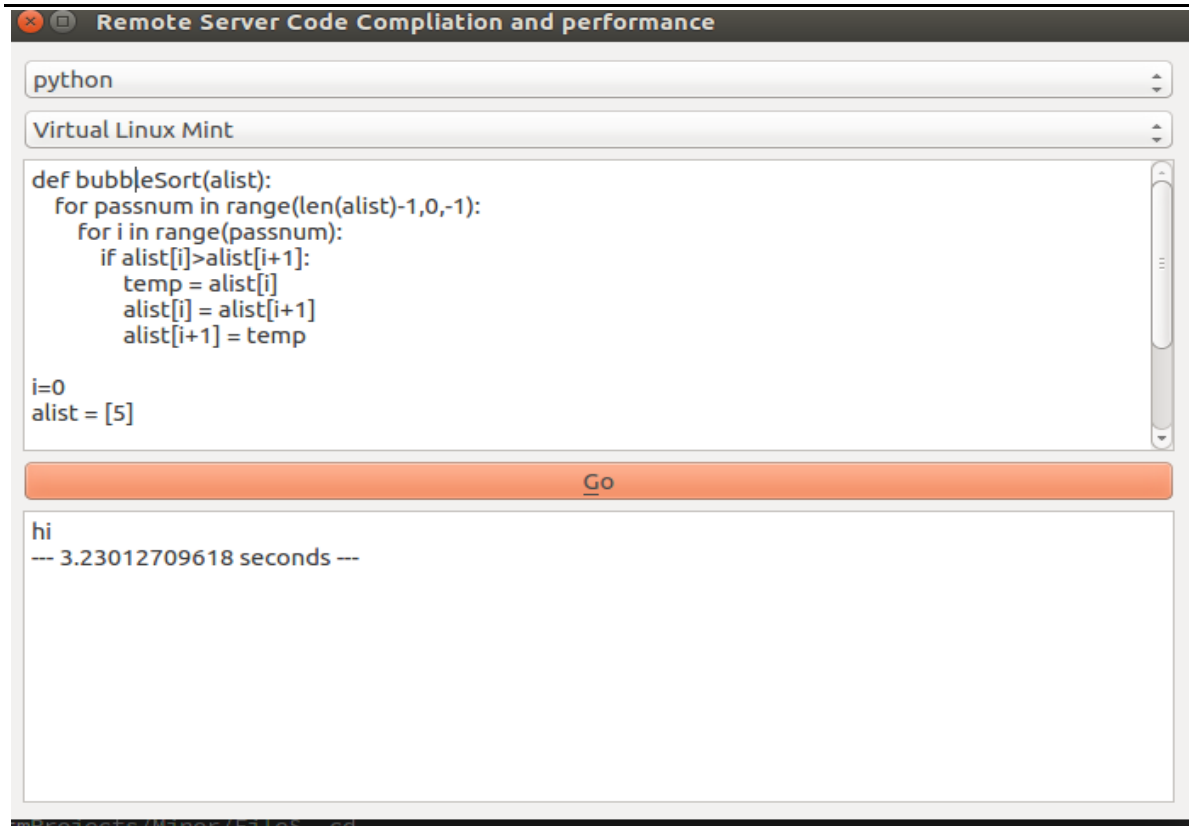
C++

Virtual Ubuntu 15.04

```
#include<iostream>
using namespace std;
int main(){
cout<<"hello world";
}
```

Go

hello world



Remote Server Code Compilation and performance

python

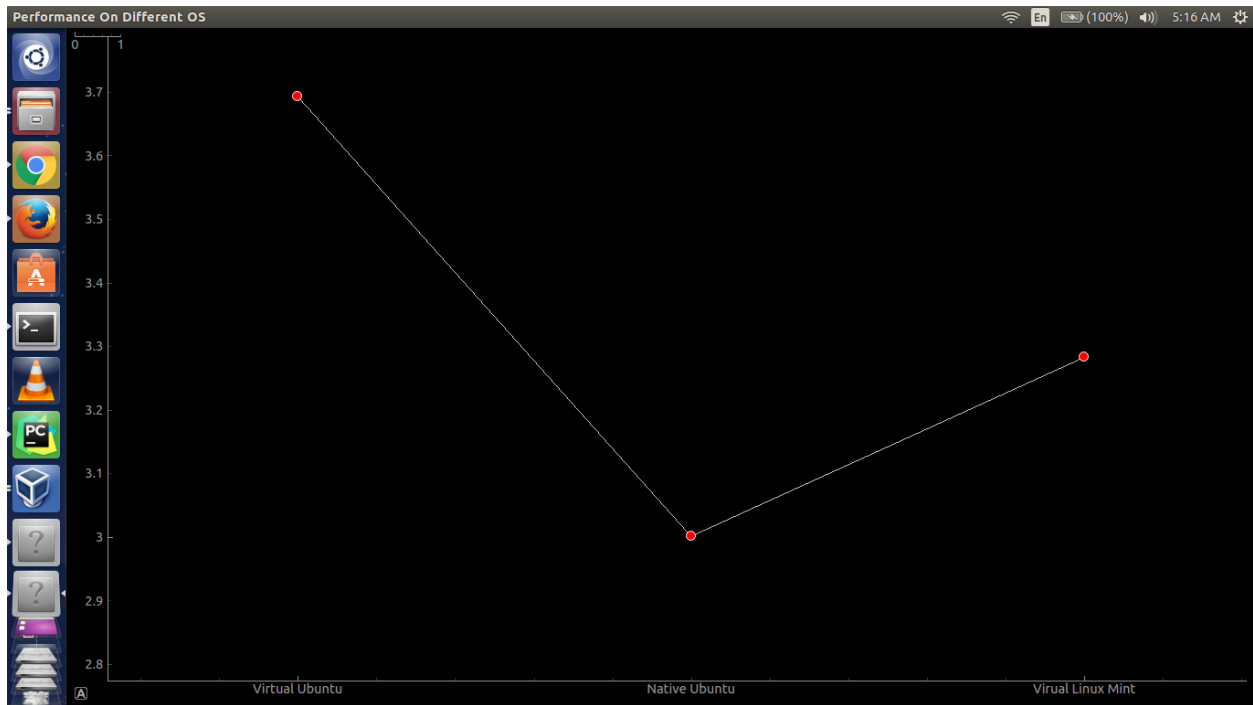
Virtual Linux Mint

```
def bubbleSort(alist):
    for passnum in range(len(alist)-1,0,-1):
        for i in range(passnum):
            if alist[i]>alist[i+1]:
                temp = alist[i]
                alist[i] = alist[i+1]
                alist[i+1] = temp

i=0
alist = [5]
```

Go

hi  
--- 3.23012709618 seconds ---



File Sharing in Virtual OS

Browse

Ubuntu Server

Upload

batdroid.png  
command.txt  
myfile.txt

Download

Quit

## REFERENCES

- GOOGLE
- [www.wikipedia.org](http://www.wikipedia.org)
- [www.stackoverflow.com](http://www.stackoverflow.com)
- [http://www.tutorialspoint.com/python/python\\_networking.htm](http://www.tutorialspoint.com/python/python_networking.htm)
- <http://docs.paramiko.org/en/1.16/>
- <https://wiki.qt.io/PySideDocumentation>