Cloud and Virtualization Systems Engineering
Final Project Report
Group 2

Leena Singhal
Qiyuan Zhou
Surabhi Agrawal

Aim:

Type a command on host only and it should execute commands on 3 different guest OS VMs. Show 3 different interesting scenarios. The guest OS can be Linux (any flavor) or Windows XP, still the commands should execute. Linux commands like “ls” for the common command interface should get executed as “dir” on Windows type machines.

Purpose:

The primary purpose of this project is to understand host guest communication and inter-OS translation of commands. In the real world scenario, this could be useful in situations where we need to perform workload balancing based on performance monitoring of different VMs on the system. We could run a single command on the host and see which processes among all VMs are taking maximum CPU time.

System maintenance and software upgrades is another area where executing a single command on the host would be useful and more efficient as opposed to running the same command separately on all the VMs. Eg: When all the VMs need to be restarted or formatted for use by a different customer, running a single command on the host can save a lot of time.

Host Guest Communication Infrastructure:

Oracle VM VirtualBox

VirtualBox is an extremely feature rich, high performance product for customers. It is installed on an existing host operating system as a hypervisor. Within this application, additional guest operating systems, each known as a Guest OS, can be loaded and run, each with its own virtual environment. User of VirtualBox can load multiple guest OSs under a single host OS. Each guest can be started, paused, resumed and terminated independently within its own VM. The host OS and guest OSs and applications can communicate with each other through a number of mechanisms. For our project, we have used VBoxManage to control virtual machines from the host.

VBoxManage

VBoxManage is the command line interface for VirtualBox. Using VBoxManage, we can completely control VirtualBox’s virtual machines from the command line of our host operating system. We have used the command ‘**guestcontrol**’ along with VBoxManage to control

things inside the virtual machine and to get commands to run on the virtual machine from the host.

Implementation:

We have implemented 4 scenarios in this project to demonstrate host-guest communication and inter-OS command translation.

The following Operating Systems have been used:
Host OS: Mac
Guest OS: Windows 7, Ubuntu, Mint

We created a bash script (approx 200 lines) which would be executed on the host and it would perform functions on the virtual machines. It also saves some information obtained from the virtual machines into a file on the host. Our script implements 4 interesting scenarios which would be of use in the real cloud computing environment.

1) **Taking a snapshot of all Virtual Machines**

This scenario is very useful in the case where a cloud service provider might want to take a snapshot of a machine before provisioning it to a tenant. They might want to restore the machine to that original snapshot when the machine has been freed by the tenant.

Taking a snapshot of a virtual machine saves its current state and enables us to return to the same state at any time. When we take a snapshot, VirtualBox captures the entire state of the virtual machine including disk, memory and settings. When we revert to that snapshot, all changes made after that snapshot are discarded.

The function executes as follows:
i) The function asks the user to enter a snapshot name
ii) Then for every running virtual machine it executes the command below:

```
$VBoxMngPath/VBoxManage snapshot $i take $snapshot
```

where `$VBoxMngPath` is a variable defined in the config file and `$i` is the name of the virtual machine on which the command is being executed.

2) **Obtaining Resource Utilization data from all machines and storing it in a file (ResourceReport.txt) on the host**

This functionality obtains the CPU utilization, RAM usage, Disk Utilization and I/O statistics for each of the virtual machines (Windows, Ubuntu and Mint) and stores it in a file on the host for quick review. This scenario is useful for monitoring the performance of all Virtual Machines. By saving information in a text file, it would be easy to lookup virtual machine statistics and identify malfunctioning behaviors.

The function executes as follows:

i) The function check for the OS running on that virtual machine and runs the corresponding block of code.

Eg: If it is a Windows, a command such as below would be executed

```
$VBoxMngPath/VBoxManage guestcontrol Windows exec --image "cmd.exe"
--username $WinUsername --password $WinPassword --wait-stdout -- "/C"  "cd
$WinDir & cscript //nologo util.vbs" | tee -a
$ResUtilFilePath/ResourceReport.txt
```

and if it is linux,

```
$VBoxMngPath/VBoxManage guestcontrol $i exec --image
"$CommandPath1/vmstat" --username $Username --password $Password
--wait-stdout | tee -a $ResUtilFilePath/ResourceReport.txt
```

3) **Creating text files on all Virtual Machines**

This scenario might be useful when a file is required to be populated on all Virtual Machines. For example, if you want to populate a README file to all VMs for its users.

This functionality creates ten text files under the directory specified in config.txt. File names are test#.txt, sample#.txt and trial.txt based on random generated numbers from 1 to 3.

The commands for creating text files for Windows and linux respectively are:

```
$VBoxMngPath/VBoxManage guestcontrol $i exec --image "cmd.exe"
--username $WinUsername --password $WinPassword --wait-stdout -- "/C"
"echo.>$WinDir\temp\test$j.txt"
```

```
$VBoxMngPath/VBoxManage guestcontrol $i exec --image
"$CommandPath1/touch" --username $Username --password $Password
--wait-stdout -- $Path/test$j.txt
```

where j is between 1 and 10. This function creates 10 files, but the number of test.txt , sample.txt and trial.txt are created depend on the modulo 3 of a randomly generated number.

4) **Deleting created text files from all Virtual Machines**

This deletes all files from a directory in which the earlier text files were created. This scenario could be useful when all user created files need to be cleared from the system.

The Windows command for this operation is:

```
	$VBoxMngPath/VBoxManage guestcontrol $i exec --image "cmd.exe"
--username $WinUsername --password $WinPassword --wait-stdout -- "/C"  "cd
$WinDir\temp & del *.txt"
```

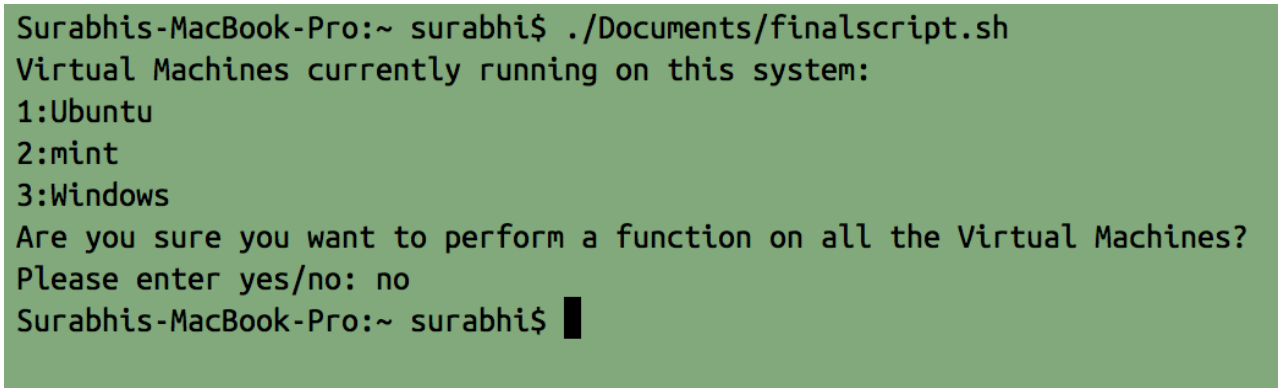
This command deletes all files with a .txt extension from the temp folder

The linux command is:

```
	$VBoxMngPath/VBoxManage guestcontrol $i exec --image
"$CommandPath2/rm" --username $Username --password $Password -- "-r"
"$CommandPath4/"
	$VBoxMngPath/VBoxManage guestcontrol $i exec --image
"$CommandPath2/mkdir" --username $Username --password $Password --
"$CommandPath4"
```

Screenshots:

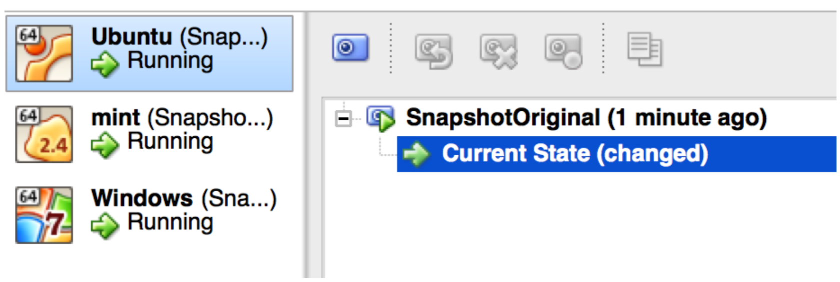
Our script handles the scenario in which the user executes the script even though he/she may not want to execute the same function on all the virtual machines. The script prompts the user and he/she can select 'no' and the script will exit. Below is a screenshot for the case.

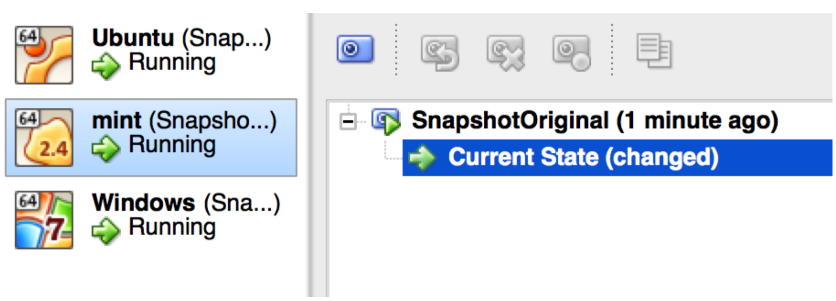


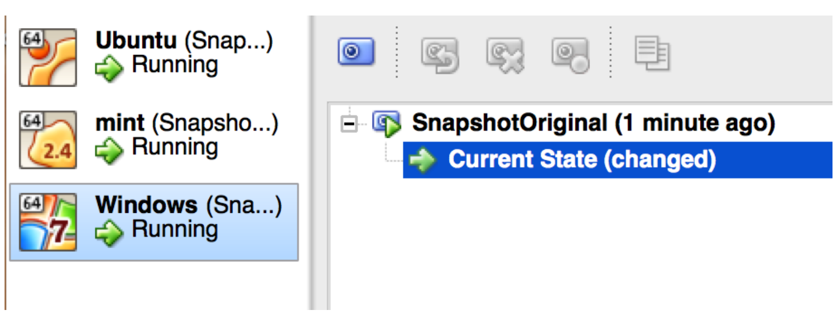
1) Taking a snapshot of all Virtual Machines

```
Surabhis-MacBook-Pro:~ surabhi$ ./Documents/finalscript.sh
Virtual Machines currently running on this system:
1:Ubuntu
2:mint
3:Windows
Are you sure you want to perform a function on all the Virtual Machines?
Please enter yes/no: yes

----Select the operation you want to perform---
1) Taking snapshot of all VMs
2) Obtain current resource utilization of all VMs
3) Create text files
4) Delete all .txt files in a folder
----Select Operation ID: 1
Please enter the name of the Snapshots:
SnapshotOriginal
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Snapshots successfully saved..
Surabhis-MacBook-Pro:~ surabhi$
```







2) **Obtaining Resource Utilization data from all machines and storing it in a file on the host (ResourceReport.txt)**

```
Surabhis-MacBook-Pro:~ surabhi$ ./Documents/finalscript.sh
Virtual Machines currently running on this system:
1:Ubuntu
2:mint
3:Windows
Are you sure you want to perform a function on all the Virtual Machines?
Please enter yes/no: yes

----Select the operation you want to perform---
1) Taking snapshot of all VMs
2) Obtain current resource utilization of all VMs
3) Create text files
4) Delete all .txt files in a folder
----Select Operation ID: 2


<----Ubuntu---->


<-----Key Resource Utilization Data----->

---Memory,process and CPU related displays:---

procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 244860  53160 316112    0    0   814    16   65  580  4  5 91  0  0


---RAM Usage:---

             total       used       free     shared    buffers     cached
Mem:       1017364     772504     244860       4004      53160     316164
-/+ buffers/cache:     403180     614184
Swap:      1046524          0    1046524


---Disk Usage:---

Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       6.8G  3.8G  2.7G  59% /
none            4.0K     0  4.0K   0% /sys/fs/cgroup
udev            486M  4.0K  486M   1% /dev
tmpfs           100M  860K   99M   1% /run
```

```
<----Ubuntu---->

<-----Key Resource Utilization Data----->

---Memory,process and CPU related displays:---

procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 244860  53160 316112    0    0   814    16   65  580  4  5 91  0  0


---RAM Usage:---

             total       used       free     shared    buffers     cached
Mem:       1017364     772504     244860       4004      53160     316164
-/+ buffers/cache:     403180     614184
Swap:      1046524          0    1046524


---Disk Usage:---

Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       6.8G  3.8G  2.7G  59% /
none            4.0K     0  4.0K   0% /sys/fs/cgroup
udev            486M  4.0K  486M   1% /dev
tmpfs           100M  860K   99M   1% /run
none            5.0M  4.0K  5.0M   1% /run/lock
none            497M   76K  497M   1% /run/shm
none            100M   40K  100M   1% /run/user
DownlSh         112G   76G   36G  68% /media/sf_DownlSh
/dev/sr0         56M   56M     0 100% /media/surabhi/VBOXADDITIONS_4.3.26_98988


---I/O Statistics:---

Linux 3.16.0-23-generic (surabhi-VirtualBox)    06/02/15        _x86_64_        (1 CPU)
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           3.94    0.47    4.67    0.36    0.00   90.55

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
scd0              0.05         0.15         0.00         70          0
sda              20.89       781.83        15.91     365419       7436
```

```
<----mint---->

<-----Key Resource Utilization Data----->

---Memory,process and CPU related displays:---

procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0   1104  22312  12968 169448    0    4  1440    48   90  398  3  2 94  1  0


---RAM Usage:---

             total       used       free     shared    buffers     cached
Mem:        501764     479708      22056       4664      12968     169648
-/+ buffers/cache:     297092     204672
Swap:      1328124       1104    1327020


---Disk Usage:---

Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       8.5G  4.4G  3.7G  55% /
none            4.0K     0  4.0K   0% /sys/fs/cgroup
udev            230M  4.0K  230M   1% /dev
tmpfs            50M  1.1M   48M   3% /run
none            5.0M     0  5.0M   0% /run/lock
none            245M 1000K  245M   1% /run/shm
none            100M   16K  100M   1% /run/user
/dev/sr0         56M   56M     0 100% /media/ujjwal/VBOXADDITIONS_4.3.26_989881


---I/O Statistics:---

Linux 3.13.0-37-generic (ujjwal-VirtualBox)     06/02/15        _x86_64_        (1 CPU)
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           3.15    0.00    1.78    0.85    0.00   94.22

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
scd0              0.11         0.40         0.00        118          0
sda              48.88      1399.95        48.09     418011      14360
```

<----Windows---->

<-----Key Resource Utilization Data----->

---CPU Utilization:---

System   0
smss   0
csrss   0
wininit   0
csrss#1   0
winlogon   0
services   0
lsass   0
lsm   0
svchost   0
VBoxService   0
svchost#1   0
svchost#2   0
svchost#3   0
svchost#4   0
audiodg   0
svchost#5   0
svchost#6   0
spoolsv   0
svchost#7   0
svchost#8   0
taskhost   0
dwm   0
explorer   0
VBoxTray   0
SearchIndexer   0
wmpnetwk   0
svchost#9   0
WmiPrvSE   0
sppsvc   0
svchost#10   31
dllhost   0
VBoxService#1   18
conhost   0
cmd   0
cscript   0
WmiApSrv   0
WmiPrvSE#1   0
WmiPrvSE#2   0

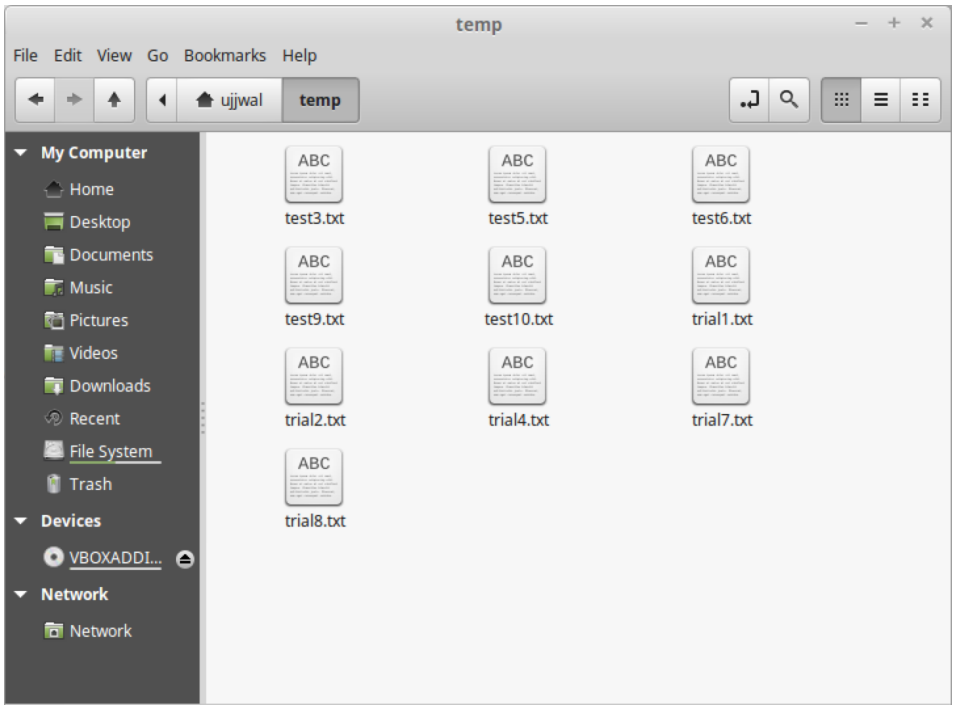
---Memory Usage:---

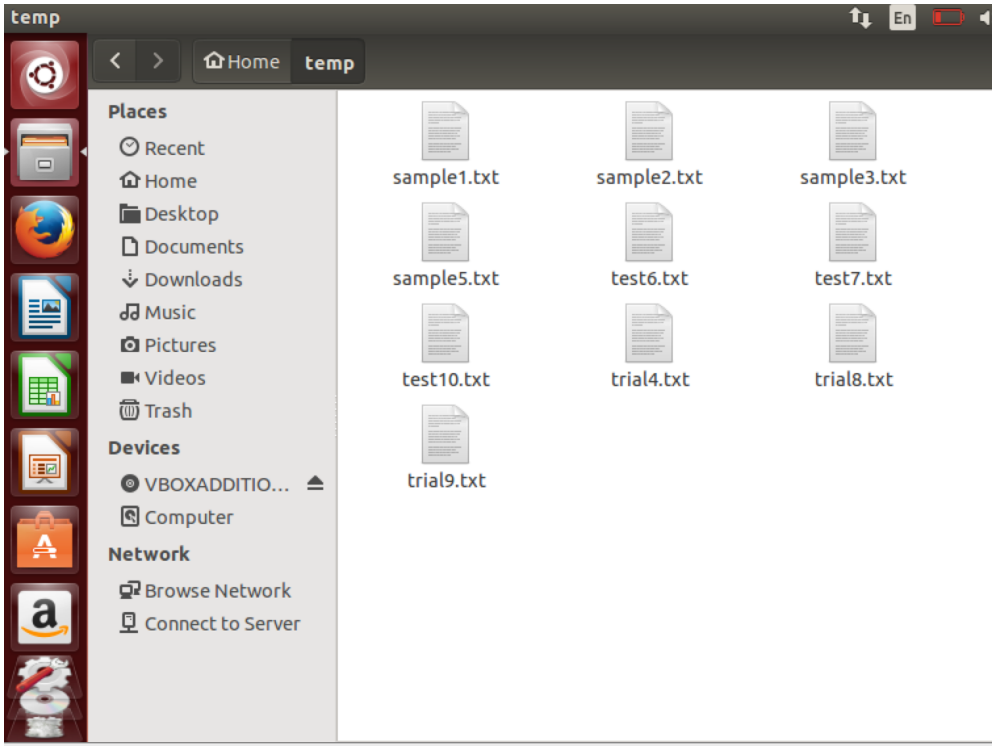
Total Physical Memory: 512 MB
Available Physical Memory: 191 MB
Virtual Memory: Max Size: 1,536 MB
Virtual Memory: Available: 1,000 MB
Virtual Memory: In Use: 536 MB

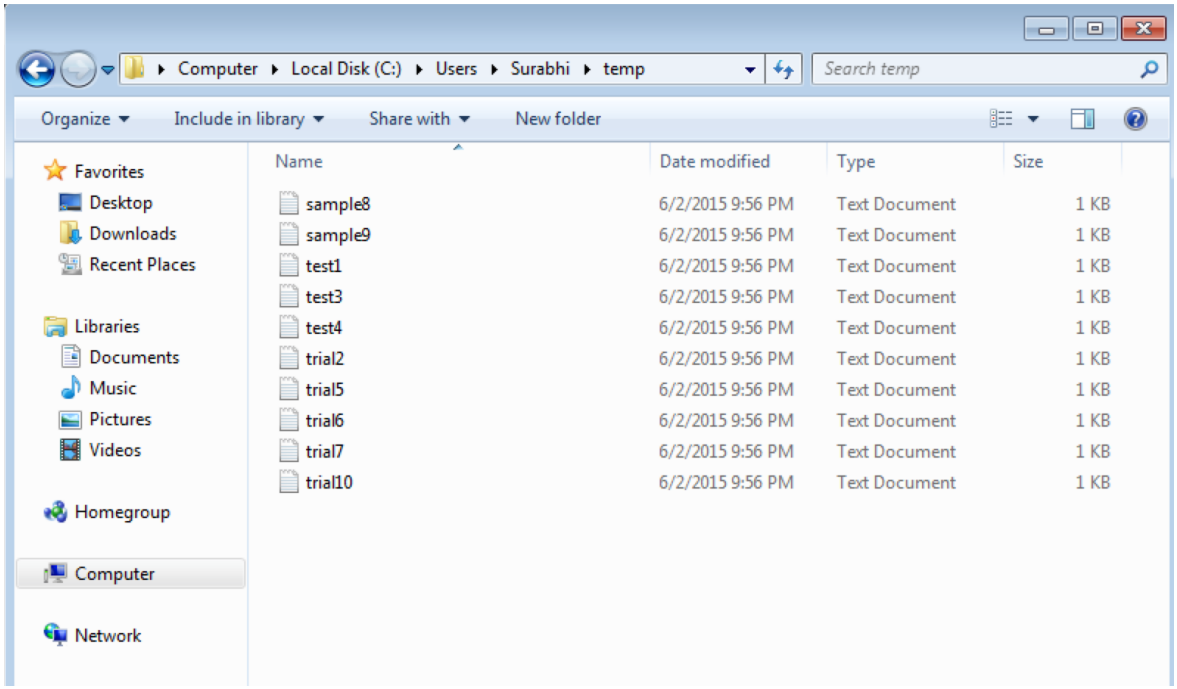
3) Creating text files on all Virtual Machines

```
Surabhis-MacBook-Pro:~ surabhi$ ./Documents/finalscript.sh
Virtual Machines currently running on this system:
1:Ubuntu
2:mint
3:Windows
Are you sure you want to perform a function on all the Virtual Machines?
Please enter yes/no: yes

----Select the operation you want to perform---
1) Taking snapshot of all VMs
2) Obtain current resource utilization of all VMs
3) Create text files
4) Delete all .txt files in a folder
----Select Operation ID: 3
Surabhis-MacBook-Pro:~ surabhi$
```



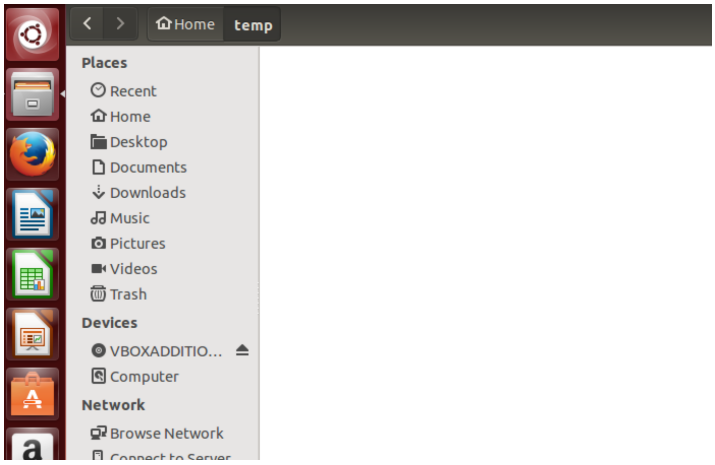

temp
En
Home
temp
Places
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Devices
VBOXADDITIO...
Computer
Network
Browse Network
Connect to Server
sample1.txt
sample2.txt
sample3.txt
sample5.txt
test6.txt
test7.txt
test10.txt
trial4.txt
trial8.txt
trial9.txt

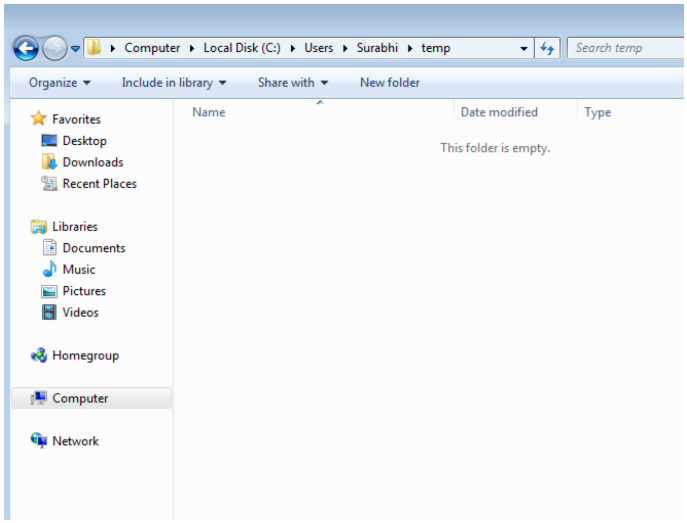

Computer ▸ Local Disk (C:) ▸ Users ▸ Surabhi ▸ temp
Search temp
Organize
Include in library
Share with
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Homegroup
Computer
Network
Name
Date modified
Type
Size
sample8 6/2/2015 9:56 PM Text Document 1 KB
sample9 6/2/2015 9:56 PM Text Document 1 KB
test1 6/2/2015 9:56 PM Text Document 1 KB
test3 6/2/2015 9:56 PM Text Document 1 KB
test4 6/2/2015 9:56 PM Text Document 1 KB
trial2 6/2/2015 9:56 PM Text Document 1 KB
trial5 6/2/2015 9:56 PM Text Document 1 KB
trial6 6/2/2015 9:56 PM Text Document 1 KB
trial7 6/2/2015 9:56 PM Text Document 1 KB
trial10 6/2/2015 9:56 PM Text Document 1 KB

4) Deleting created text files from all Virtual Machines

```
Surabhis-MacBook-Pro:~ surabhi$ ./Documents/finalscript.sh
Virtual Machines currently running on this system:
1:Ubuntu
2:mint
3:Windows
Are you sure you want to perform a function on all the Virtual Machines?
Please enter yes/no: yes

----Select the operation you want to perform---
1) Taking snapshot of all VMs
2) Obtain current resource utilization of all VMs
3) Create text files
4) Delete all .txt files in a folder
----Select Operation ID: 4
[2800 - Session 3]
[2805 - Session 4]
[2570 - Session 3]
[2575 - Session 4]
Surabhis-MacBook-Pro:~ surabhi$
```





Summary:

We have successfully implemented 4 scenarios to demonstrate inter-OS command translation through host-guest communication. We understand that in a real world scenario, VMs are most likely to be distributed among a cloud network. Then accessing those VMs may require extra connection through TCP/IP protocol, authentication and authorization. The command translation can only be applied to the functionalities we have implemented. Instead of directly enter commands from the host, we used a menu-based control to run the built-in functionalities. In the future, what we can do better is write a script that performs a function, translate it so that it is applicable for different OSes, and populate them onto VMs. We can then connect to the VMs and run the scripts locally to receive more powerful functionalities.