LAB 1 REPORT

Gal Fajon, 123124084

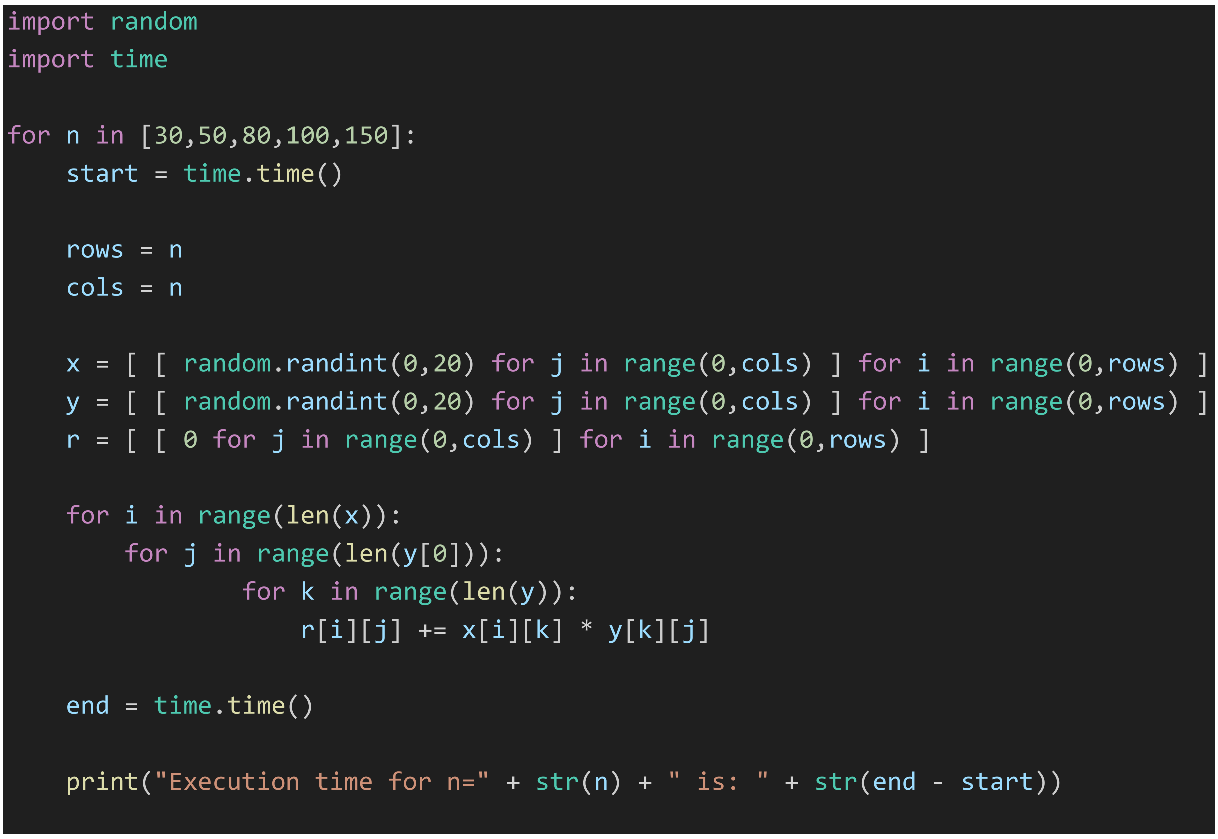
CS3204, Cloud infrastructure and services

Dr. John Herbert

1. WORK ENVIRONMENT AND APPLICATIONS USED

For the assignments of the lab exercises, I used my laptop with an installation of the Windows 10 operating system, an Oracle VirtualBox virtual machine with an installation of Lubuntu 22.04.3 (a lightweight flavour of Ubuntu), as well as the Docker Desktop app (using the WSL2 backend) and the official Ubuntu image, version 22.04.3.

To test execution times, I used the matrix multiplication program below. It is worth noting that both operating systems used the same version of Python for execution (3.10.12).



```
import random
import time

for n in [30,50,80,100,150]:
    start = time.time()

    rows = n
    cols = n

    x = [ [ random.randint(0,20) for j in range(0,cols) ] for i in range(0,rows) ]
    y = [ [ random.randint(0,20) for j in range(0,cols) ] for i in range(0,rows) ]
    r = [ [ 0 for j in range(0,cols) ] for i in range(0,rows) ]

    for i in range(len(x)):
        for j in range(len(y[0])):
                for k in range(len(y)):
                    r[i][j] += x[i][k] * y[k][j]

    end = time.time()

    print("Execution time for n=" + str(n) + " is: " + str(end - start))
```

2. VIRTUALIZATION AND CONTAINERS

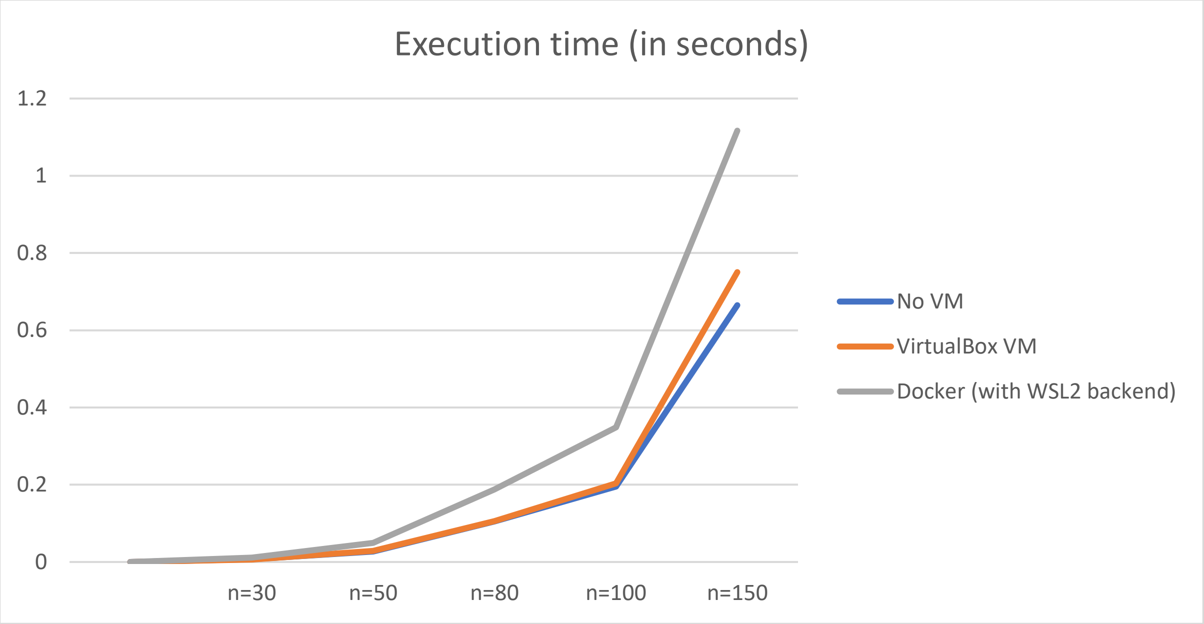
A virtual machine is a simulated machine running within a hypervisor, an environment that allows the host OS to imitate the functionality of whatever OS is installed on the visor as well as the hardware of the virtual machine. This allows a single host machine to potentially run several operating systems with different assigned hardware resources. The simulated machines do not have direct access to hardware resources, creating a large execution overhead.

Containers, on the other hand, do not virtualize the hardware of the machine and share the host operating system kernel and use it to run an application using the software libraries and environment variables that are necessary. They use features of the host operating system to isolate their processes and control the amount of resources they use. Resource limits for containers are soft, meaning they can expand the amount of resources they can use quickly if other containers are not using them.

3. EXECUTION RESULTS

When analysing the execution times of the program, I got the following results:

Execution time (in seconds)	No VM	VirtualBox VM	Docker container (with WSL2 backend)
n=30	0.006998777	0.006443024	0.011445761
n=50	0.027053356	0.028741837	0.049228907
n=80	0.105044365	0.105819225	0.187860966
n=100	0.194957256	0.203652859	0.348770618

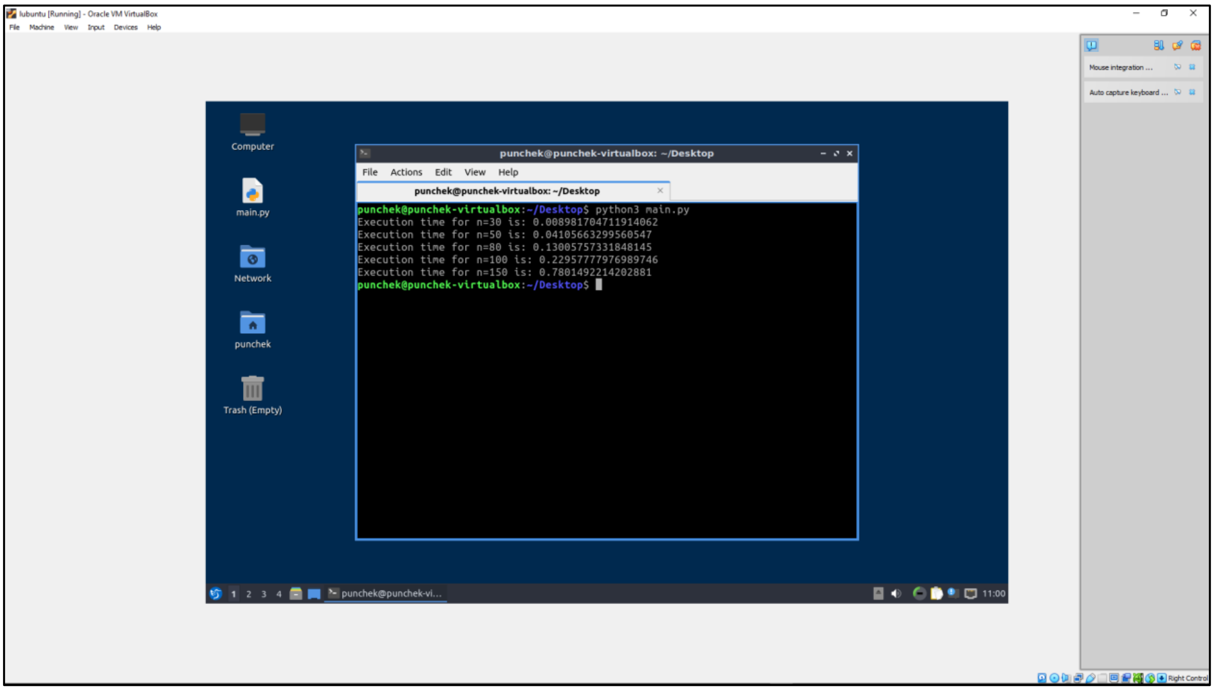


An interesting thing to observe here is that the execution time of the Docker container is marginally worse than that of the VM. This is because Docker containers on Windows operating systems work under an extra layer of virtualization, as Docker uses WSL2, a lightweight Linux virtual machine built into Windows to create and run containers. This causes additional execution overhead. If the same container were to be run on a native Linux installation, it is safe to assume its execution times would be better than those of the virtual machine, as the container could run natively on the Linux operating system. The best execution time came from running the program on the host machine without any virtualization.

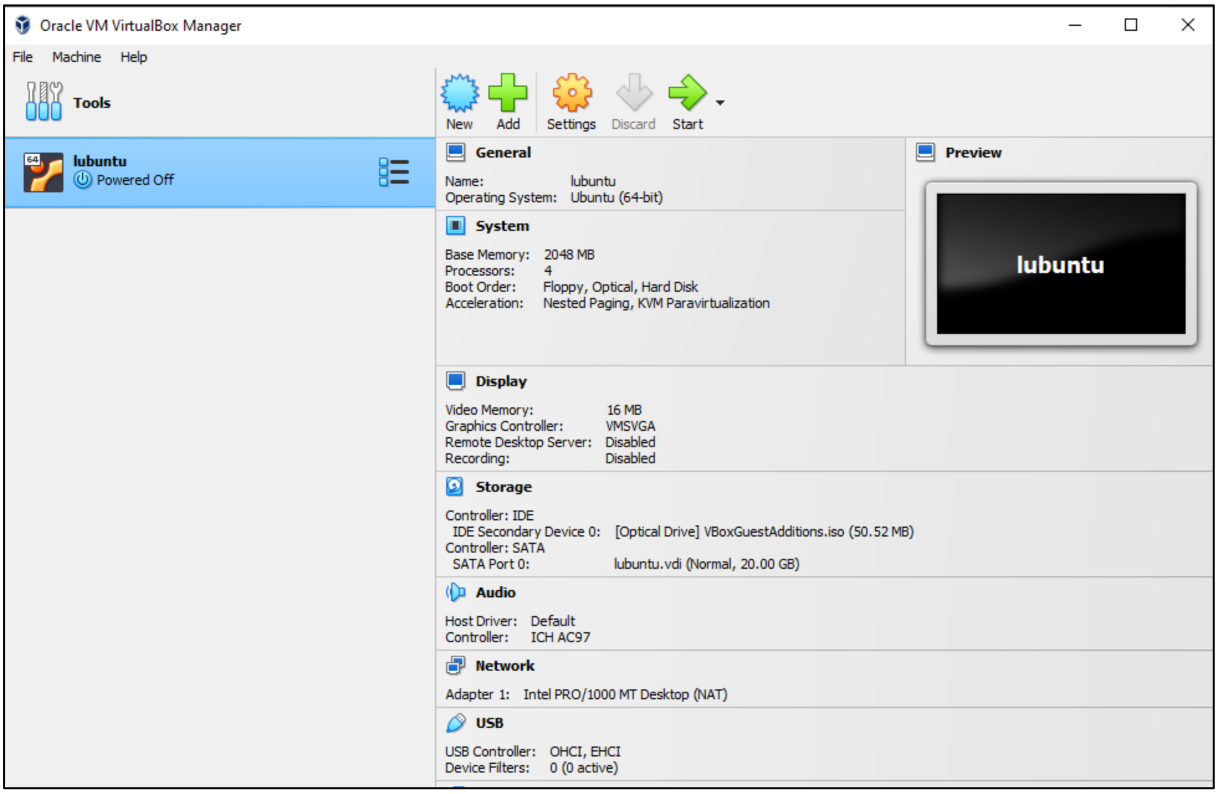
4. WHAT I LEARNED

In this lab, I learned how to configure a virtual machine on the windows operating system, as well as install images and run Docker containers using the Docker desktop app.

5. SCREENSHOTS


lubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Computer
main.py
Network
punchek
Trash (Empty)
punchek@punchek-virtualbox: ~/Desktop
File Actions Edit View Help
punchek@punchek-virtualbox: ~/Desktop
punchek@punchek-vi...
11:00
Mouse integration ...
Auto capture keyboard ...
Right Control

```
punchek@punchek-virtualbox:~/Desktop$ python3 main.py
Execution time for n=30 is: 0.008981704711914062
Execution time for n=50 is: 0.04105663299560547
Execution time for n=80 is: 0.13005757331848145
Execution time for n=100 is: 0.22957777976989746
Execution time for n=150 is: 0.7801492214202881
punchek@punchek-virtualbox:~/Desktop$
```



General

Name: lubuntu
Operating System: Ubuntu (64-bit)

System

Base Memory: 2048 MB
Processors: 4
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] VBoxGuestAdditions.iso (50.52 MB)
Controller: SATA
SATA Port 0: lubuntu.vdi (Normal, 20.00 GB)

Audio

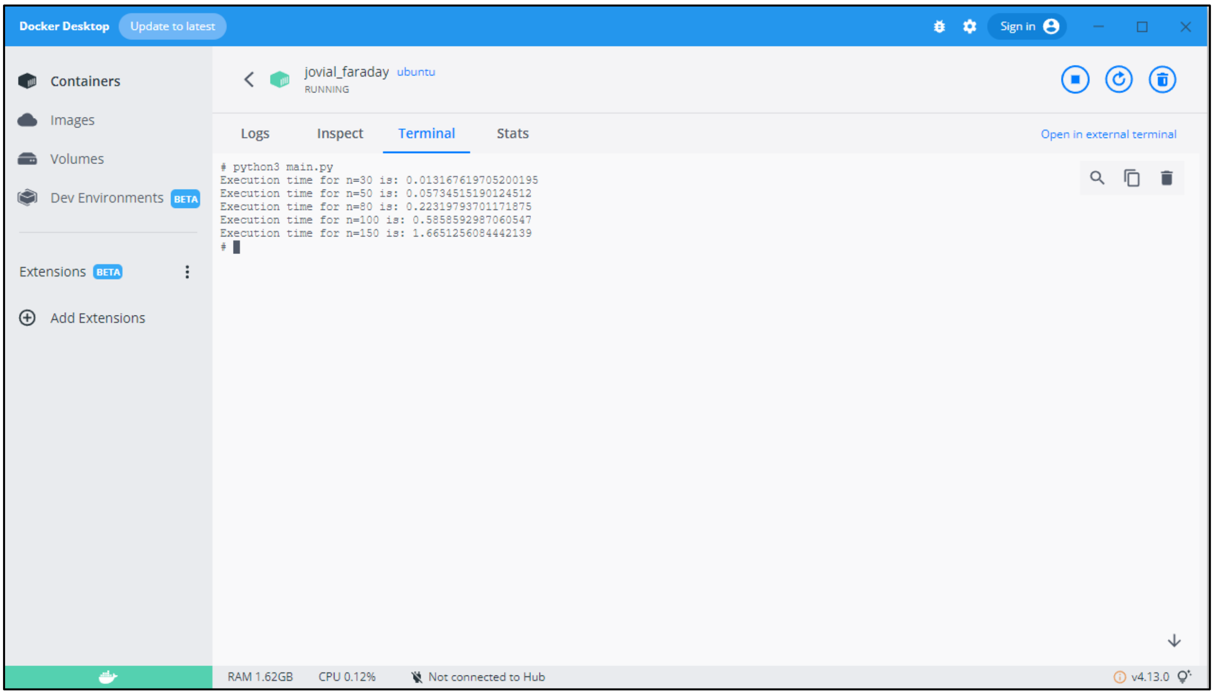
Host Driver: Default
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB

USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)



```
# python3 main.py
Execution time for n=30 is: 0.013167619705200195
Execution time for n=50 is: 0.05734515190124512
Execution time for n=80 is: 0.22319793701171875
Execution time for n=100 is: 0.5858592987060547
Execution time for n=150 is: 1.6651256084442139
# 
```

These screenshots were taken after already completing the table of execution times and were taken with several applications running in the background, their results differ from the table.

6. SOURCES

- Ubuntu Docker image: https://hub.docker.com/_/ubuntu
- Lubuntu .iso install and release notes: https://lubuntu.me/jammy-3-released/
- What is WSL: https://learn.microsoft.com/en-us/windows/wsl/about
- Explanation of the WSL backend for Docker: https://docs.docker.com/desktop/wsl/