CS3204 Assignment 5

Jack Hallissey

In this report I will describe how I explored Docker containerization technology by running a performance-sensitive application in a Docker container and comparing the performance with a virtual machine.

Overview

Containers give the impression of a separate operating system, however, they differ from virtual machines in that they share the kernel of the host operating system.

This gives containers a number of advantages:

- They are lightweight and provide fast performance.
- Fast provisioning – containers can be instantiated in seconds.
- They can contain fully packaged software with all dependencies included.
- Multiple containers can be started from the same image.
- They are extremely portable.
- It is easy to manage the orchestration and scaling of multiple containers.

However, it is important to note that virtual machines provide better isolation and security than containers.

Docker is a platform that allows you to create, run, manage and share containers.

Steps Followed

First, I installed and configured Docker on my PC and ensured it was functioning correctly by testing it with a basic “Hello World” container.

I then pulled the image for Debian from Docker Hub.

```
docker pull debian
```

I created a container using this image.

```
docker run -it --name my-debian-cont e11072c1614c
```

Once the container was created, I updated the list of available packages and installed Python.

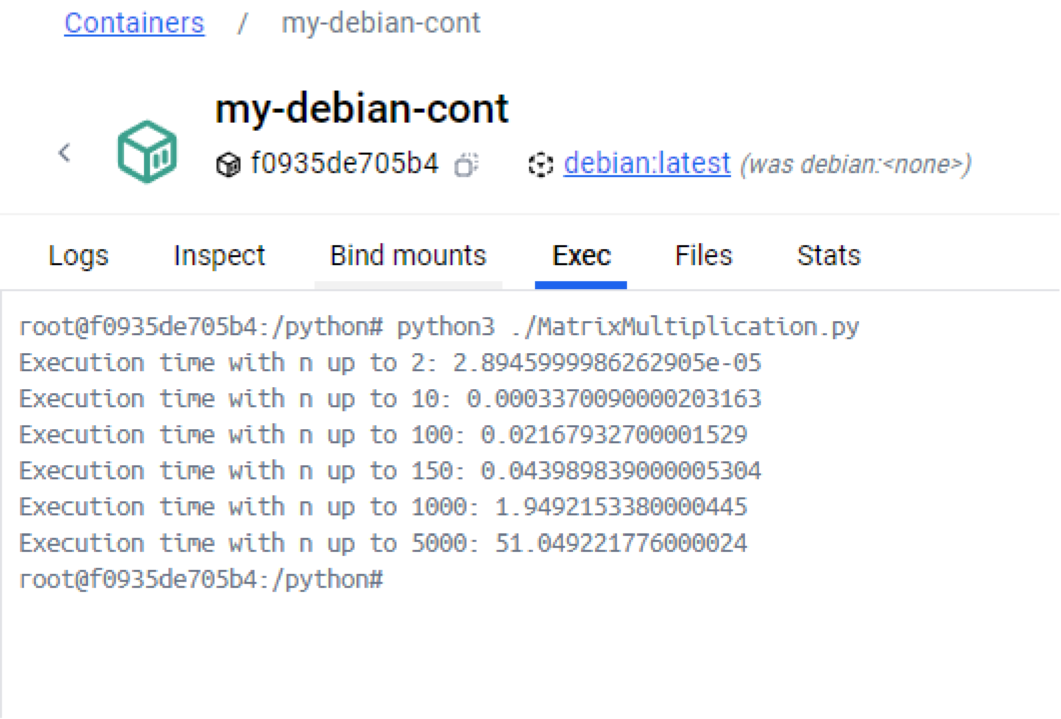
```
apt update

apt install python3
```

Finally, I copied my Python program into the container’s file system using the Docker Desktop application, and executed the program.

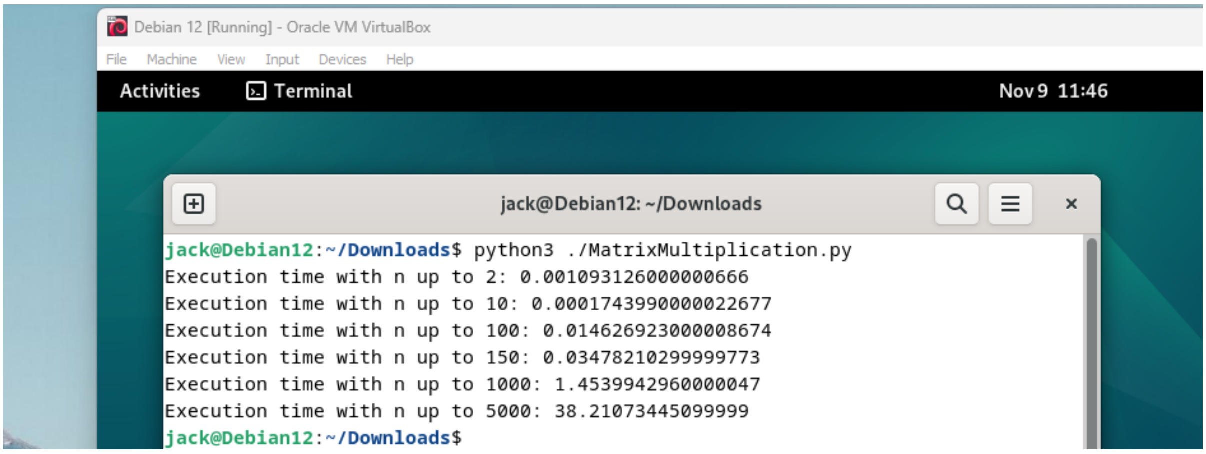
Results

Docker Container



```
root@f0935de705b4:/python# python3 ./MatrixMultiplication.py
Execution time with n up to 2: 2.8945999986262905e-05
Execution time with n up to 10: 0.0003370090000203163
Execution time with n up to 100: 0.021679327000001529
Execution time with n up to 150: 0.0439898390000005304
Execution time with n up to 1000: 1.9492153380000445
Execution time with n up to 5000: 51.049221776000024
root@f0935de705b4:/python#
```

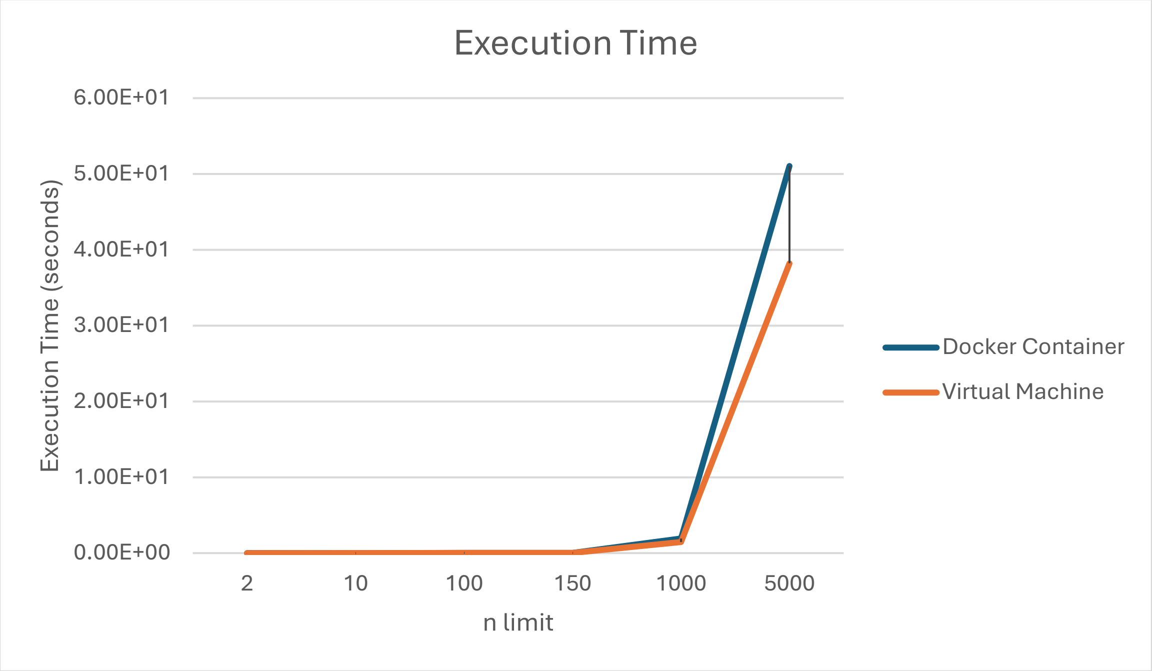
Virtual Machine

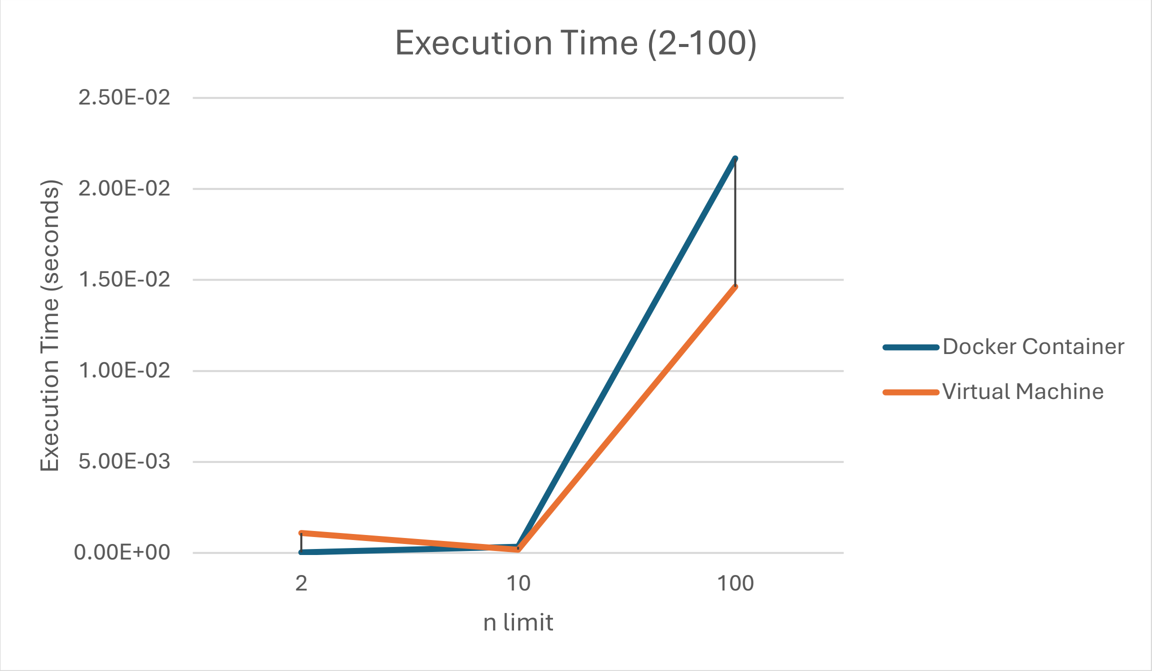

Debian 12 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Activities
Terminal
Nov 9 11:46
jack@Debian12: ~/Downloads

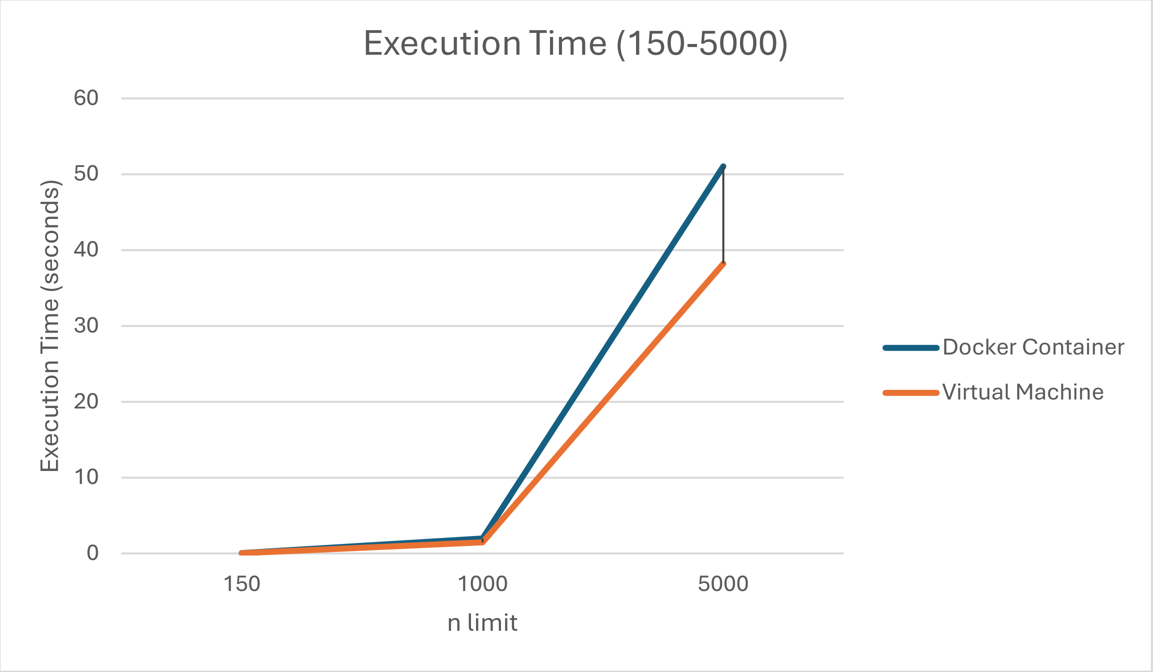
```
jack@Debian12:~/Downloads$ python3 ./MatrixMultiplication.py
Execution time with n up to 2: 0.001093126000000666
Execution time with n up to 10: 0.00017439900000022677
Execution time with n up to 100: 0.014626923000008674
Execution time with n up to 150: 0.03478210299999773
Execution time with n up to 1000: 1.4539942960000047
Execution time with n up to 5000: 38.21073445099999
jack@Debian12:~/Downloads$
```

Execution Time Comparison

n up to	Docker Container	Virtual Machine
2	2.8945999986262905e-05	0.001093126000000666
10	0.00033700900002031 63	0.0001743990000022677
100	0.02167932700001529	0.01462692300008674
150	0.043989839000005304	0.034782102999999773
1000	1.9492153380000445	1.4539942960000047
5000	51.049221776000024	38.21073445099999






Execution Time (150-5000)
60
50
40
30
20
10
0
Execution Time (seconds)
150
1000
5000
n limit
Docker Container
Virtual Machine

Analysis

Interestingly, the execution time was generally lower on the virtual machine than on the Docker container. This could be due to a variety of factors, such as differences in the configuration of the two environments, background processes running on the host system while measuring the performance of the Docker container, or differences in how resources are allocated to a particular process in the two environments.

If the execution time had been lower on the Docker container, as expected, this could have been attributed to the fact that the container shares the kernel of the host operating system, while the virtual machine does not. This means that better performance is generally observed on containers compared to virtual machines.

Reflections

I initially had some trouble with creating the container using the Docker Desktop application, but this was resolved when I created the container from the command line. I also experienced issues with installing Python and running the program on the container, but this was resolved relatively quickly.

As with the previous assignment, I repeated my test a number of times after observing that the execution time was generally lower on the virtual machine, however, this continued to be the case.

I learned a lot about Docker and containerization technology from completing this task. Setting up the container was a valuable experience, and I was surprised at how fast I was able to get the container ready, especially compared to virtual machines.