FACULTY OF COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY

UNIVERSITY OF MALAYA

INDUSTRIAL TRAINING FINAL REPORT

SEMESTER 1 SESSION 2022 - 2023

MEHRAN GHAROONI KHOSHKEHBAR

S2014607/1

SUPERVISOR FROM THE FACULTY: DR. AZNUL QALID MD SABRI

Table of Contents

i. Abstract

I did my internship at the Technical Department of Nottingham University. When I started, the university had launched a new system called HPC, but it was in its early stages and had many flaws. HPC is a system for sharing the resources of a supercomputer among users, Reuther et al (2013)[1], since this supercomputer has LinuxOS (Ubuntu), I had to learn everything about LinuxOS, which was a very valuable lesson for me. So, after that, I was able to interact with the HPC system, I got a limited HPC account like students, and I tried to challenge the system.

During my internship, I did many tests on HPC, such as running servers and accessing them outside the network, running heavy artificial intelligence codes like image processing, testing the code differences in running time between a GPU and a CPU, etc. I also installed a lot of software on HPC such as Gromax, Paraview, MATLAB, etc. While during each tasks, I learned a lot about Linux, Python, servers, sites, databases, and other stuffs that are valuable to me.

In this internship, the most important subject I learned was how to deal with problems and how solve various issues.

ii. Acknowledgment

I would like to express my most profound appreciation to all those who provided me assistance to pass this internship successfully. A special gratitude I give to my supervisor in the workplace, Mr. Ellias Ismail, whose contribution in stimulating suggestions and encouragement, helped me to pass my internship with a lot of valuable new knowledge.

Furthermore, I would also like to acknowledge with much appreciation of my colleagues Ms. Izyan, Mr. Saiful, Mr. Hamdi, Mr. Kumar, and others colleagues who established such a friendly environment that made me happy to work here.

Last but not least, many thanks go to the head of the project, Dr. Aznul Qalid Md Sabri whose have invested his full effort in guiding me in achieving my goals. I have to appreciate the guidance given by other supervisors as well as Dr. Lim Chee Kau and Dr. Hoo Wai Lam which have improved my internship due to their comments and advices.

iii. List of Tables

iv. List of Figures

1 Overview of Company/Organization

I spent the whole 24 weeks of my internship at the University of Nottingham Malaysia. A private campus of the University of Nottingham, and the University of Nottingham Malaysia is one of its great branches (generally there are 3 branches for the University of Nottingham, in the UK, China, and Malaysia). In Semenyih, Selangor, the university is located. The Malaysian Ministry of Higher Education has rated the university as "excellent" or tier 5 on a scale of tier 1-6 and categorized it as a private institution (SETARA 2009) [2].

1.1 History:

In 1998, a collaboration between Boustead Holdings Berhad, YTL Corporation Berhad, and the University of Nottingham was publicly launched in response to a request by the UK Ministry of Education to create an overseas campus.

This project marked the establishment of a British University's first-ever branch campus outside of the country.

In Kuala Lumpur, the University of Nottingham enrolled its first 89 students in September 2000. The university moved to a 118-acre campus in Semenyih, in the Broga Hill valley, in 2005. For the purpose of instructing MBA students, the campus reopened a branch in Kuala Lumpur in Chulan Tower on Jalan Conlay in 2006 (University of Nottingham, n.d.) [3].

1.2 Vision:

Their vision is to be a university without boundaries where they welcome the opportunities given by a changing world and where ambitious individuals and creative culture will empower them to improve a lot of humanity (University of Nottingham 2022) [4].

1.3 Business:

The University of Nottingham offers courses from the foundation level all the way up to doctoral degrees through its specialist schools and departments under the Faculties of Arts and Social Sciences, and Engineering and Science, and is ranked 114 in the world and (18 in the UK) out of more than 1,000 universities in the QS World University Rankings 2023.

A support team of more than 600 people helps our student population of 5000 students, who represent 75 different nations, and ensures that operations at the university are carried out properly and efficiently even in the event of a pandemic. (University of Nottingham, n.d.) [5].

1.4 Organization Structure:

Council and Senate are the two principal entities that oversee the University. The Chancellor and Vice-Chancellor, Professor Shearer West, are in charge of the University.

The administrative and support departments of the university assist in its management and contribute to its expansion and success on a national and worldwide level (University of Nottingham, n.d.) [6].

The technical department where I began working is under the direction of the Faculty of Science and Engineering (FOSE), and Mr. Ellias Ismail, the department head, had a special plan for me from the start so that as a user I could check and test some systems (like HPC), but with the existence of the abilities that I had learned from the university and learn during the internship, the plans set for me changed and became far more complex, more important, and higher level.

2 Work Experience

I accomplished a variety of tasks throughout the course of the 6-month industrial training period, from understanding Linux environments to operating SLURM on the server. In the report that follows, I will go into depth about every task I completed for this internship duration. It should also be noted that during this training course, there was no team in the work environment, and I completed all the tasks individually.

2.1 The Industrial Training Schedule

	1	2	3	4	5	6	7	8	9	10	11	12
Linux Environment	■	■										
Learning the HPC		■	■	■								
Installing Jupyter			■	■								
Difference between the HPC and Local Jupyter				■	■							
VNC				■	■							
HPC troubleshooting					■	■	■					
Installing Gromacs						■	■					
AI codes in Local Host							■	■	■			
Matlab environment								■	■			
Matlab Job Allocation									■	■	■	■
Image Processing on HPC									■	■	■	■
GPU vs CPU in HPC											■	■

Table 1 - Weekly schedule from week 1 to week 12

	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Paraview	▨	▨	▨	▨										
HPC Script		▨	▨	▨										
Creating a new HPC				▨	▨	▨	▨	▨	▨					
Chemical Team's Database								▨	▨	▨	▨	▨		
HPC Portal											▨	▨	▨	▨

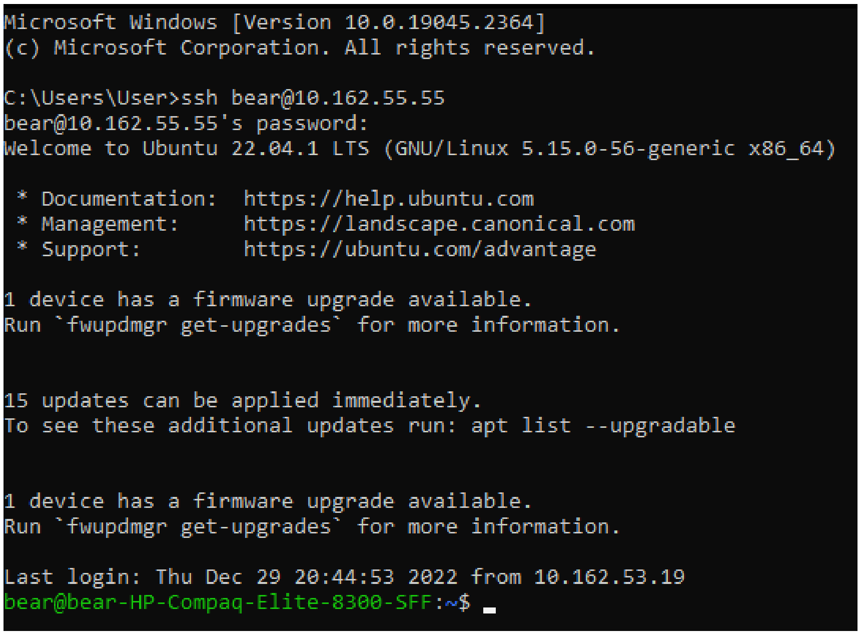
Table 2 - Weekly schedule from week 12 to week 15

2.2 Linux Environment:

- Devices or systems used for performing the tasks: Ubuntu 22.04.
- Given timeframe to complete the tasks: One week, from week 1 until week 2.
- Problems faced during the execution of the tasks: A completely new environment, the lack of information, and different thinking ways.
- How problems were solved: Learning the new environment and practicing a lot.
- Relation of this task to passed courses at the university: So much related to the Operating Systems course (WIA2004) and understanding the concept of UNIX systems.

How to operate in a Linux environment was one of the most crucial subjects I learned from my internship. At first, it was difficult to comprehend, yet after some time, I understood how vital it is and I made me an effort to study it. Following a 3-hour YouTube course where I acquired the fundamentals of Linux and gained an understanding of broad ideas, I started studying the "Linux Commands Handbook." This valuable resource has 135 pages. and explains every crucial commands that are needed. These taught me how to operate in a Linux environment.

In response to my supervisor's request, I started using Linux properly. The first thing I did was establish a connection to the server. I searched through the SSH code and these linking components because there were so many new terms. I was successful in connecting my laptop to the server.



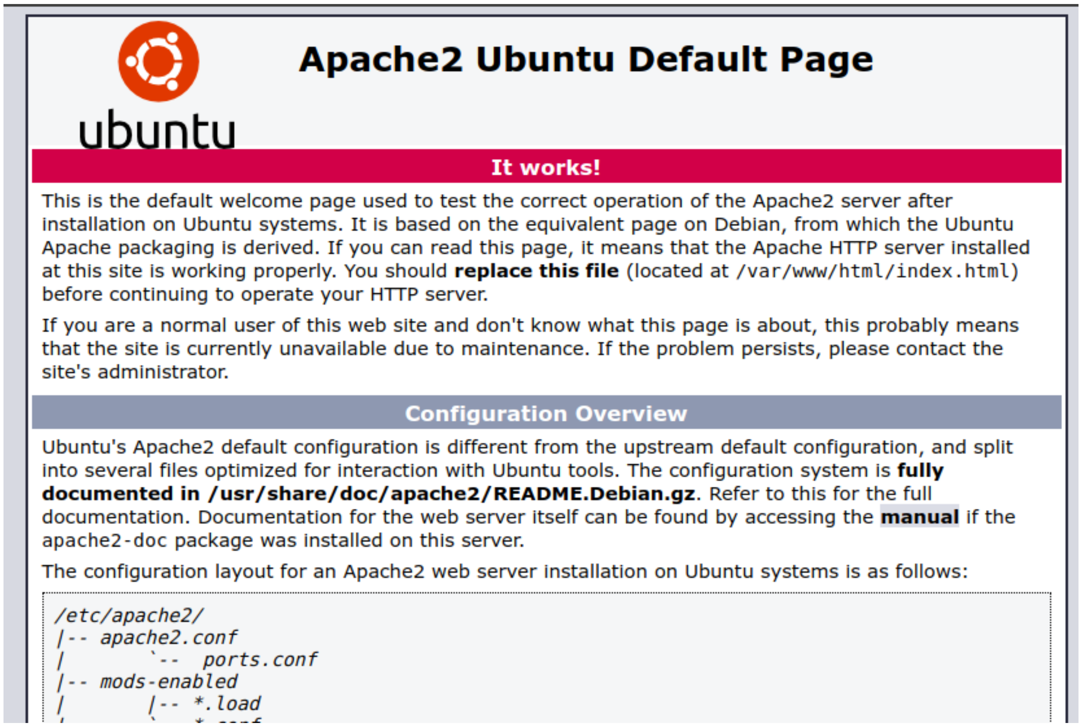
```
Microsoft Windows [Version 10.0.19045.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>ssh bear@10.162.55.55
bear@10.162.55.55's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-56-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 device has a firmware upgrade available.
Run `fwupdmgr get-upgrades` for more information.


15 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


1 device has a firmware upgrade available.
Run `fwupdmgr get-upgrades` for more information.

Last login: Thu Dec 29 20:44:53 2022 from 10.162.53.19
bear@bear-HP-Compaq-Elite-8300-SFF:~$
```

Figure 1 - Connect to Ubuntu via ssh from Windows

My supervisor asked me to install APACHE on the server. I was initially a little bewildered since I knew nothing about APACHE, but as I learned more about it, I began to see how valuable it is. I, nonetheless, firstly attempted to install it on my PC because it would be preferable to run some tests before trying it out on the server.

Same as I mentioned about SSH, APACHE also had a lot of new features, so I started learning about them. I tried to alter the localhost port and IP and add a domain name before installing APACHE on the server PC.

However, the localhost page that appeared was rather standard, so I made an effort to alter it by adding some HTML to the localhost folder. As a result, the localhost page was modified.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|        `--  ports.conf
|-- mods-enabled
|        |-- *.load
|        `-- *.conf
```

Figure 2 - The default page of Apache2

Additionally, I learned how to "Totally" uninstall a program from Linux. Since apps contain data and cache, completely uninstalling them requires a process the same as uninstalling installed files.

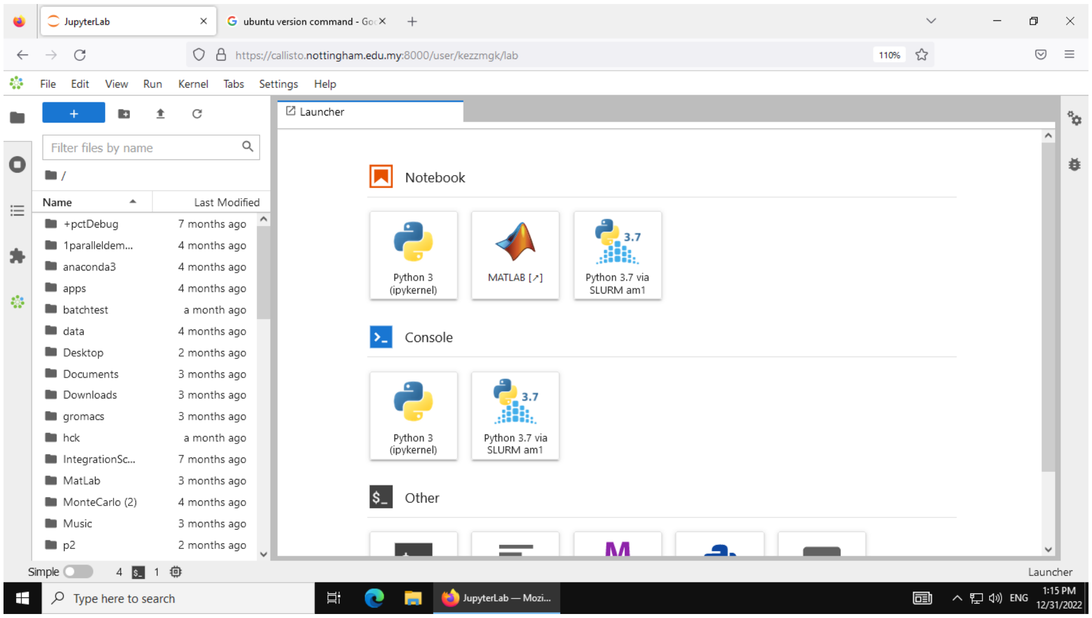
2.3 Learning the High-Performance Computing (HPC):

- Devices or systems used for performing the tasks: Ubuntu 20.04.
- Given timeframe to complete the tasks: Two weeks, from week 2 until week 4.

- Problems faced during the execution of the tasks: A new system with a lot of ambiguous points, job submission, and kernels using a different kind of Linux.
- How problems were solved: spending a whole week trying everything on HPC, following supervisor instructions to learn it efficiently.
- Relation of this task to passed courses at the university: There are several connections to the OS course (WIA2004) and working with IP review the Network Technology course (WIA1005) for me.

I comprehend the HPC system's idea and my area of responsibility, which is system testing from the perspective of a user, at a highly significant meeting.

I was able to grasp the idea of the HPC and how it functions during this meeting. I also realized that the University of Nottingham's HPC system makes use of Jupyter, which means I must learn how to use it. This program resembles execute scripts; with Jupyter, anybody may run their code from anywhere without needing to install additional software or an IDE.



Name	Last Modified
+pctDebug	7 months ago
1paralleldem...	4 months ago
anaconda3	4 months ago
apps	4 months ago
batchtest	a month ago
data	4 months ago
Desktop	2 months ago
Documents	3 months ago
Downloads	3 months ago
gromacs	3 months ago
hck	a month ago
IntegrationSc...	7 months ago
MatLab	3 months ago
MonteCarlo (2)	4 months ago
Music	3 months ago
p2	2 months ago

Figure 3 - The main page of Jupyter Lab for using HPC

Additionally, I attempted to use Jupyter to better understand how the system operated. I should installed the HPC version of Jupyter on my PC at that point so that I could test things on my computer (not a real server)

After completing the task, I had a thorough understanding of how the HPC system operates. I also learned about Jupyter and attempted to install it on the PC. I began to behave as a user that wanted to utilize the HPC system.

2.4 Installing Jupyter on the Test Server:

- Devices or systems used for performing the tasks: Ubuntu 22.04.
- Given timeframe to complete the tasks: one week, from week 3 until week 4.
- Problems faced during the execution of the tasks: Different approaches for installing Jupyter on the server, and also there are different versions of Jupyter.
- How problems were solved: Investigating Jupyter and learning how to install Jupyter using Python PIP
- Relation of this task to passed courses at the university: Because of running Jupyter on the server I had a lot of interactions with IP and Networks so it was related to the Network Technology course (WIA1005), and also the Information System course (WIA1001).

One of the tasks I did was running Jupyter on the test server, recording the procedure on a video, and making an installation code list because the code was not typical.

First off all, I attempted to utilize the "Anaconda" software to install Jupyter, however, this software downloads a tonne of unnecessary side programs that only fill up RAM. So I attempted a direct installation, which required installing Python Venv on my computer. After that, I produced Venv files and used them as a source.

By doing this, I was able to install Jupyter Notebook, but the HPC system requires Jupyter Lab instead, so I installed that instead and it successfully operated on the local host; in order to execute this, the system also needed Apache2 and Jupyter Lab.

After completing all of these steps, I installed Jupyter Lab on the test server. However, there was a problem: Jupyter Lab ran on the local host and was not accessible on other PCs. I spent a lot of time looking for a solution, and as a result, UNM test server now has an accessible Jupyter Lab.

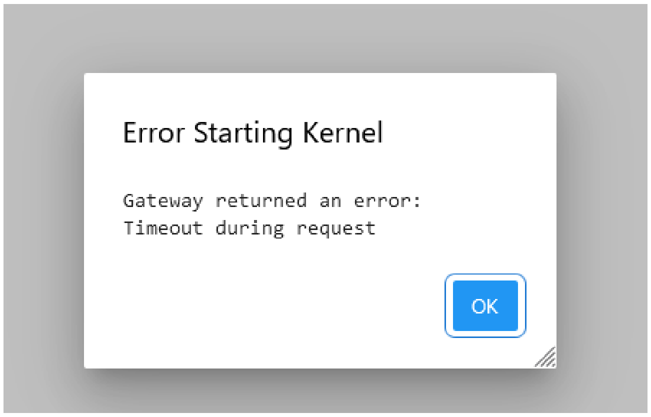
The full installation code that I provide for installing Jupyter Lab:

```
sudo apt update
sudo apt -y upgrade
sudo apt install -y python3-pip
sudo apt install build-essential libssl-dev libffi-dev python3-dev
sudo apt install -y python3-venv
cd /
sudo mkdir environments
cd environments
sudo python3 -m venv my_env
source my_env/bin/activate
sudo python3 -m pip install jupyter
sudo pip install jupyterlab
cd
jupyter lab --IP 0.0.0.0 --port 8888
```

2.5 Difference between the HPC's Jupyter and Local Jupyter:

- Devices or systems used for performing the tasks: Ubuntu 20.04.
- Given timeframe to complete the tasks: one week, from week 4 until week 5.
- Problems faced during the execution of the tasks: Not connecting to the HPC kernels and getting baffled because of differences between Jupyter Hub and Jupyter Lab.
- How problems were solved: Searching through books, the internet, and resources to find proper answers for the problem, test them, and find the solution for each problem.
- Relation of this task to passed courses at the university: Because of Linux environment connections to the OS course (WIA2004), and because of server working the Network Technology course (WIA1005), because of using python a lot of connections to the Logic Programing course (WID3001).

I attempted to utilize the HPC system with my account, but there was a significant issue that prevented my account from connecting to the server's kernel; as a result, I tried the system with the default kernel. I could compare the features of two different Jupyter types since I had access to the test server Jupyter (HPC and Local).



Error Starting Kernel

```
Gateway returned an error:
Timeout during request
```

Figure 4 - The HPC Kernel error

To test the features of the systems, I ran numerous Python programs; I specifically tried to run programs that require a lot of CPU, GPU, and Driver resources. The first distinction between Local Jupyter and HPC is that the former has access to PC drivers ("for presenting pictures, Display configuration, and Mouse Control") while the latter does not.

Another difference is about some prevention settings, in the default kernel there was some setting to prevent lags for showing outputs, but in the HPC one, there was not.

2.6 VNC (Virtual Network Computing):

- Devices or systems used for performing the tasks: Ubuntu 20.04.
- Given timeframe to complete the tasks: one week, from week 4 until week 5.
- Problems faced during the execution of the tasks: Accessibility to all users' files (for reading only) via this application.
- How problems were solved: By changing the permission of the “/home” directory in the HPC’s Ubuntu.
- Relation of this task to passed courses at the university: Because of Linux environment connections to the OS course (WIA2004), and because of being familiar system, the Information Systems course (WIA1001).

VNC (Virtual Network Computing) is an installed program, which is basically a GUI from Linux and is available on HPC's Jupyter platform. Students also have access to a fully functional computer running Ubuntu, but this VNC has some issues.

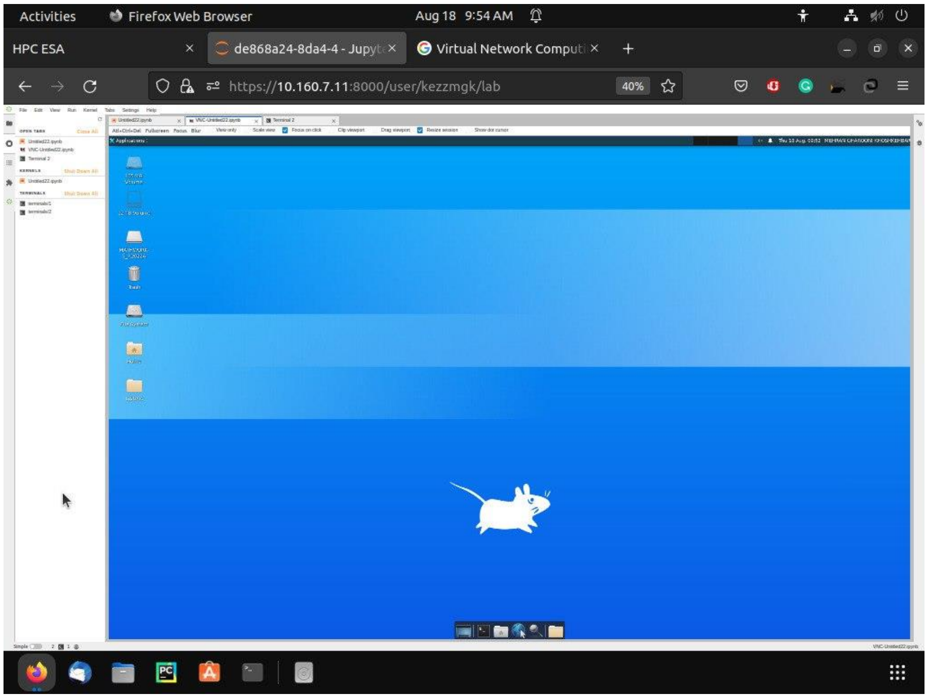
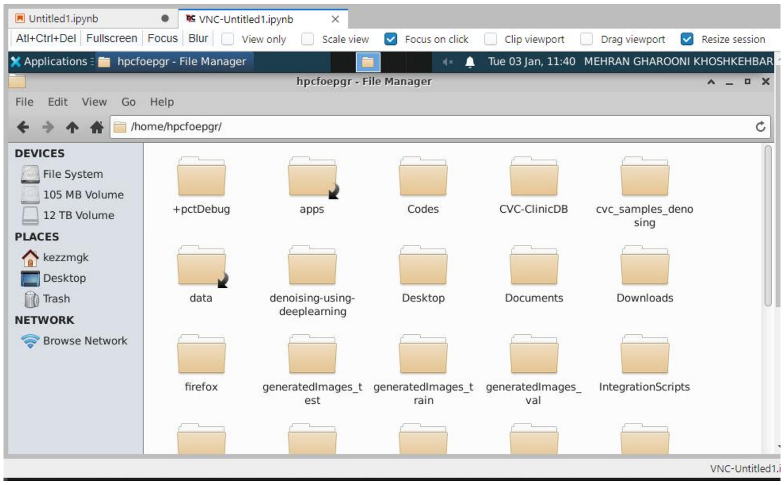


Figure 5 - The VNC interface for users

one of the VNC issues is that we have access to all users' files and can view all of their users' users. For instance, in the image below, we can see the "hpcfoepgr" account and we can also see all of its users' files and open them.



☐ View only ☐ Scale view ☑ Focus on click ☐ Clip viewport ☐ Drag viewport ☑ Resize session

/home/hpcfoepgr/

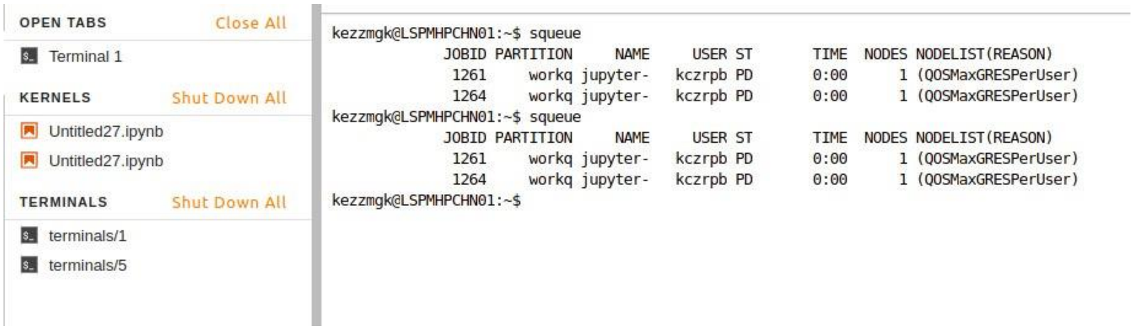
Figure 6 - The File Explorer of VNC on the HPC

By changing the permission of the "/home" directory in the HPC's Ubuntu we can protect users from other accounts. the code that we can use is *"chmod -rwx /home/*"* to remove permissions.

2.7 HPC troubleshooting:

- Devices or systems used for performing the tasks: Ubuntu 20.04.
- Given timeframe to complete the tasks: one week, from week 5 until week 7.
- Problems faced during the execution of the tasks: Several problems, kernels not working, HPC cannot recognize me as a user, and the jobs cannot submit. and the "*srun*" command which is for connecting to the Compute node is not working.
- How problems were solved: By investigating about Jupyter system, we found a very important point which is rebooting the JupyterHub Server to solve these kinds of problems.
- Relation of this task to passed courses at the university: Because of reasoning the issues and analyzing them to find the best answer, knowledge representing and reasoning course (WID2001), because of Linux environment connections to the OS course (WIA2004), because of using python a lot of connections to the Logic Programing course(WID3001).

I was given this task after my account unexpectedly ran into a major issue and all but stopped functioning. We could not turn off the two identical kernels that have been created. Additionally, no job has been assigned to them.

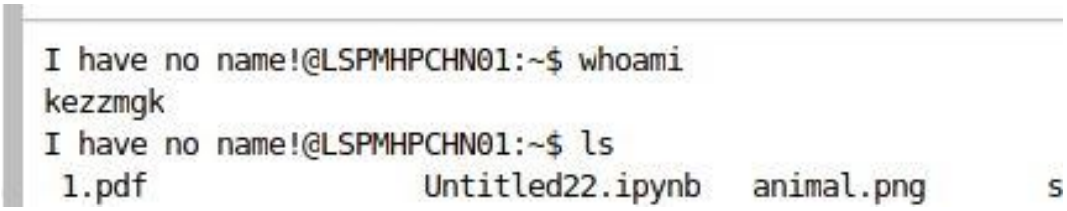


```
kezzmgk@LSPMHPCHN01:~$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
              1261     workq jupyter-   kczrpb PD       0:00      1 (QOSMaxGRESPerUser)
              1264     workq jupyter-   kczrpb PD       0:00      1 (QOSMaxGRESPerUser)
kezzmgk@LSPMHPCHN01:~$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
              1261     workq jupyter-   kczrpb PD       0:00      1 (QOSMaxGRESPerUser)
              1264     workq jupyter-   kczrpb PD       0:00      1 (QOSMaxGRESPerUser)
kezzmgk@LSPMHPCHN01:~$
```

Figure 7 - Two Kernels with the same name and no job for them

In addition to these, I was technically unable to use HPC to do anything, including developing a new kernel. After that, I double-checked and found that even "ipykernel," which is Jupyter's default, was not functioning.

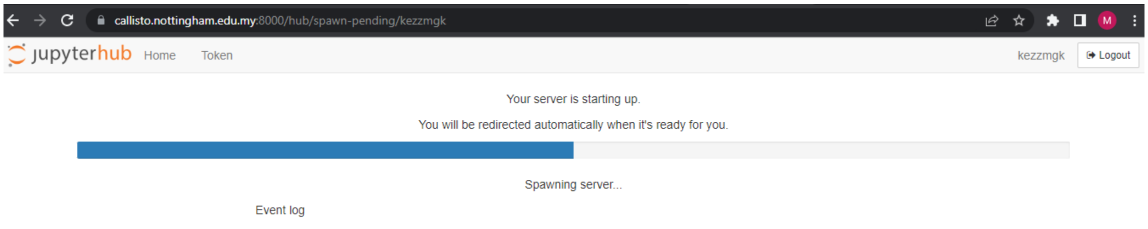
Also, when I entered my HPC account, I saw my username changed to “I have no name” in Terminal. But when I ran the “whoami” code in the Terminal, I could see my username.



```
I have no name!@LSPMHPCHN01:~$ whoami
kezzmgk
I have no name!@LSPMHPCHN01:~$ ls
 1.pdf                    Untitled22.ipynb    animal.png        s
```

Figure 8 - The HPC account domain name issue

I discovered that restarting the Jupyter-lab server is the answer to these kinds of problems.



Your server is starting up.

You will be redirected automatically when it's ready for you.

Spawning server...

Figure 9 - Restarting the Jupyter Hub Server

2.8 Installing Gromacs:

- Devices or systems used for performing the tasks: Ubuntu 20.04.
- Given timeframe to complete the tasks: one week, from week 6 until week 7.
- Problems faced during the execution of the tasks: I was unfamiliar with installing a terminal-based application, and also Gromacs needs a lot of side packages to install
- How problems were solved: Using official Gromacs Instructions and testing them on the test server before installing that on the HPC
- Relation of this task to passed courses at the university: Because of Linux environment connections to the OS course (WIA2004), and also because of working with applications and making the change on them to connect them to HPC, connections to the Software Modeling course(WIA2002).

Gromacs is a molecular dynamics package which is primarily made for simulating proteins, lipids, and nucleic acids, this research finds that the method chosen is similar to Van et al. (2005) [7].

My supervisor asked me to research GROMACS, try to install it on my PC, and then test it on an HPC. So I attempted to install it, but I ran into a variety of issues. For instance, when I searched on YouTube, there were very little material concerning GROMACS. So I browsed its website and began to gradually configure all the code. Nevertheless, one of the files I had to download did not function; in fact, the site from which I had to get the Regression testing file did not function (404 Error page). but after a day, it started to work properly.

Then I downloaded the Protein file and attempted to test it, but I noticed a lot of installation-related issues. So I tried a different way of installation that used FFTW and Cmake, which was challenging but after finishing, I received the same Error.

Finally, after installing Gromacs on the test server, I ran some files and calculations on the Gromacs. First I download the ".pdb" file from https://www.rcsb.org/, and after that, I visualize it on the "SPDBV" application which makes a 3D view of protein based on the input file.

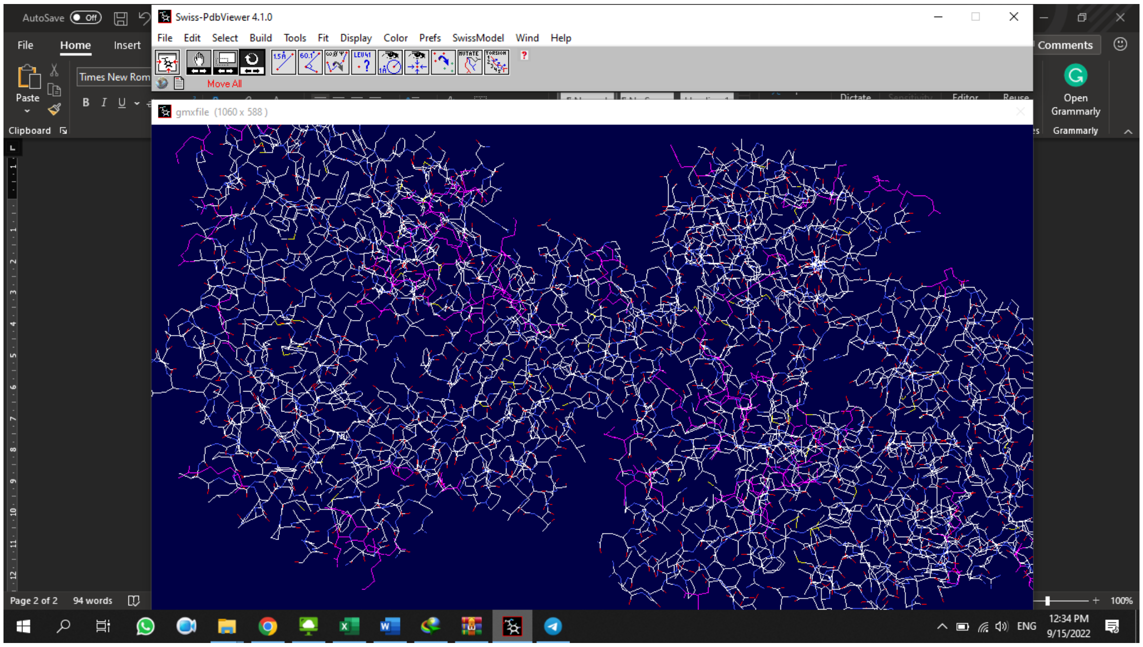
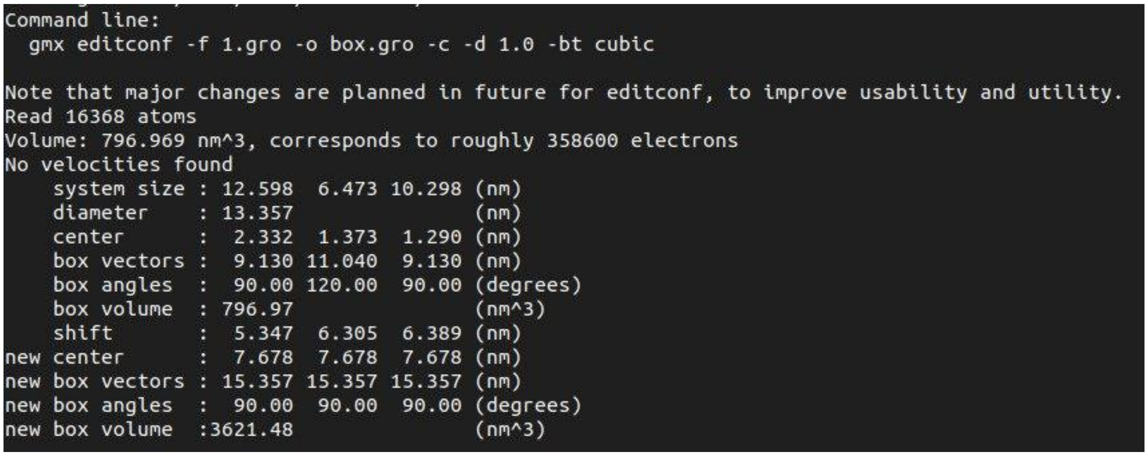


Figure 10 - 3D view of a Protein

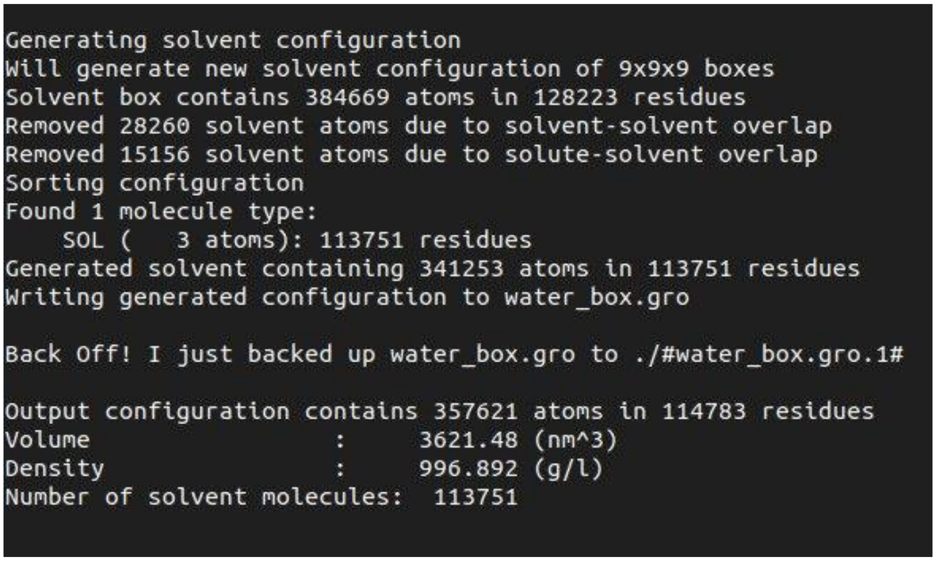
After that, I exported another ".pdb" file from this application and ran that file on the Gromacs; here are some outputs from Gromacs that shows us Gromacs is working properly.

```
$ gmx pdb2gmx -f gmxfile.pdb -o 1.gro -p 1.top -water spce
$ gmx editconf -f 1.gro -o box.gro -c -d 1.0 -bt cubic
$gmx solvate -cp box.gro -cs spc216.gro -o water_box.gro
```



```
Command line:
  gmx editconf -f 1.gro -o box.gro -c -d 1.0 -bt cubic

Note that major changes are planned in future for editconf, to improve usability and utility.
Read 16368 atoms
Volume: 796.969 nm^3, corresponds to roughly 358600 electrons
No velocities found
    system size : 12.598  6.473 10.298 (nm)
    diameter    : 13.357              (nm)
    center      :  2.332  1.373  1.290 (nm)
    box vectors :  9.130 11.040  9.130 (nm)
    box angles  :  90.00 120.00  90.00 (degrees)
    box volume  : 796.97              (nm^3)
    shift       :  5.347  6.305  6.389 (nm)
new center      :  7.678  7.678  7.678 (nm)
new box vectors : 15.357 15.357 15.357 (nm)
new box angles  :  90.00  90.00  90.00 (degrees)
new box volume  :3621.48              (nm^3)
```

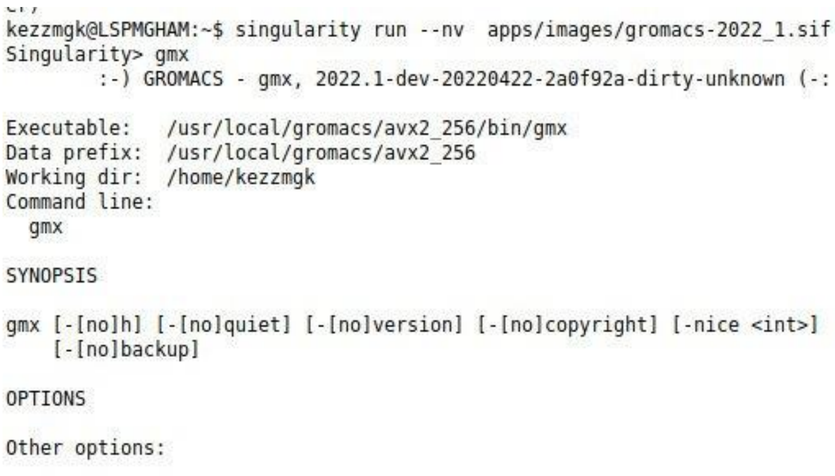
Figure 11 - Some information of Protein from Gromacs commands



```
Generating solvent configuration
Will generate new solvent configuration of 9x9x9 boxes
Solvent box contains 384669 atoms in 128223 residues
Removed 28260 solvent atoms due to solvent-solvent overlap
Removed 15156 solvent atoms due to solute-solvent overlap
Sorting configuration
Found 1 molecule type:
    SOL (   3 atoms): 113751 residues
Generated solvent containing 341253 atoms in 113751 residues
Writing generated configuration to water_box.gro

Back Off! I just backed up water_box.gro to ./#water_box.gro.1#

Output configuration contains 357621 atoms in 114783 residues
Volume                 :     3621.48 (nm^3)
Density                :     996.892 (g/l)
Number of solvent molecules:  113751
```

Figure 12 - Some other information of Protein from Gromacs commands

Finally, I tested the Gromacs on the HPC, and luckily, it was operating without a hitch. I tested it with a few files, and it functions really nicely and smoothly.



```
kezzmgk@LSPMGHAM:~$ singularity run --nv  apps/images/gromacs-2022_1.sif
Singularity> gmx
        :-) GROMACS - gmx, 2022.1-dev-20220422-2a0f92a-dirty-unknown (-:

Executable:   /usr/local/gromacs/avx2_256/bin/gmx
Data prefix:  /usr/local/gromacs/avx2_256
Working dir:  /home/kezzmgk
Command line:
  gmx

SYNOPSIS

gmx [-[no]h] [-[no]quiet] [-[no]version] [-[no]copyright] [-nice <int>]
    [-[no]backup]

OPTIONS

Other options:
```

Figure 13 - Prove of running Gromacs on HPC

2.9 AI codes in Local Host:

- Devices or systems used for performing the tasks: Windows 10.
- Given timeframe to complete the tasks: One week, from week 7 until week 9.
- Problems faced during the execution of the tasks: AI code concept was hard to understand and Running Python code in the Jupyter Lab environment was tricky.
- How problems were solved: Learning carefully the AI code comments to understand the way that they were used. Gain experience in running code in Jupyter Lab IDE.
- Relation of this task to passed courses at the university: Because running AI codes in Python is so much related to the Fundamental of Artificial Intelligence course (WIA1004), and Functional and Logic programing course (WID3001).

I tested some large, resource-intensive Python programs on my local PC before testing them on the HPC system.

I used the Jupyter Lab to run Artificial Intelligence Image Processing and saved all of the codes. For instance, I built a code that could find a similar object among given images or find the same object between 4 and 5 images.

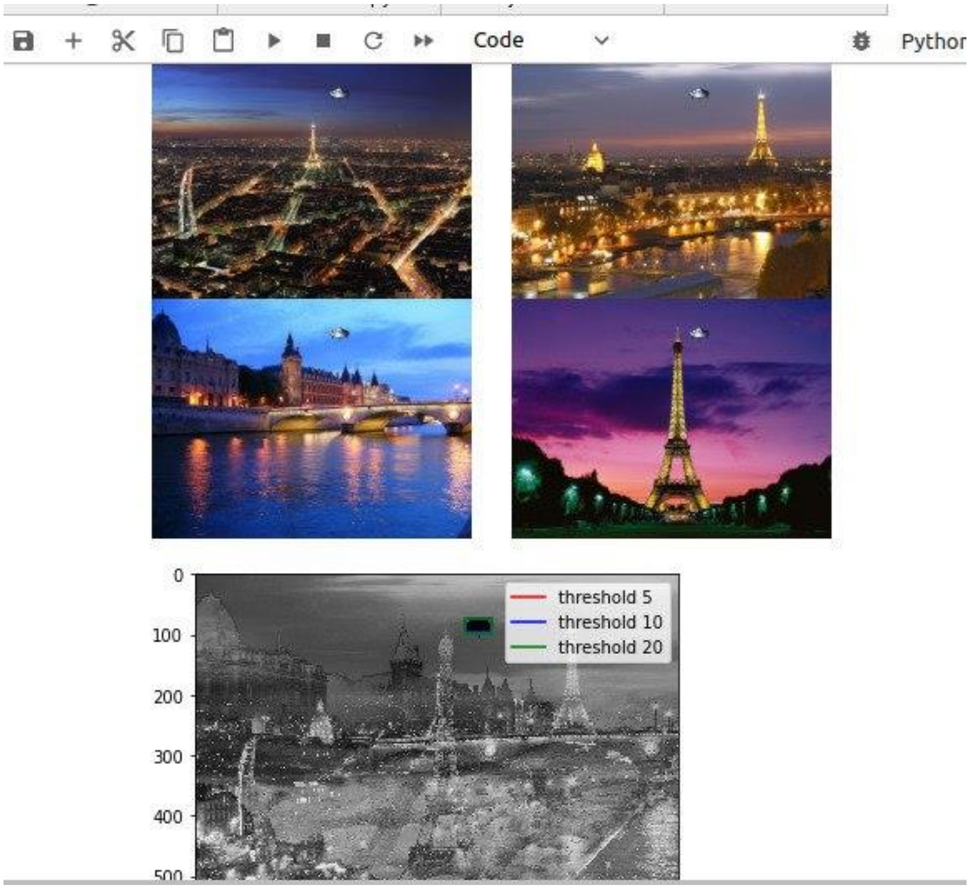
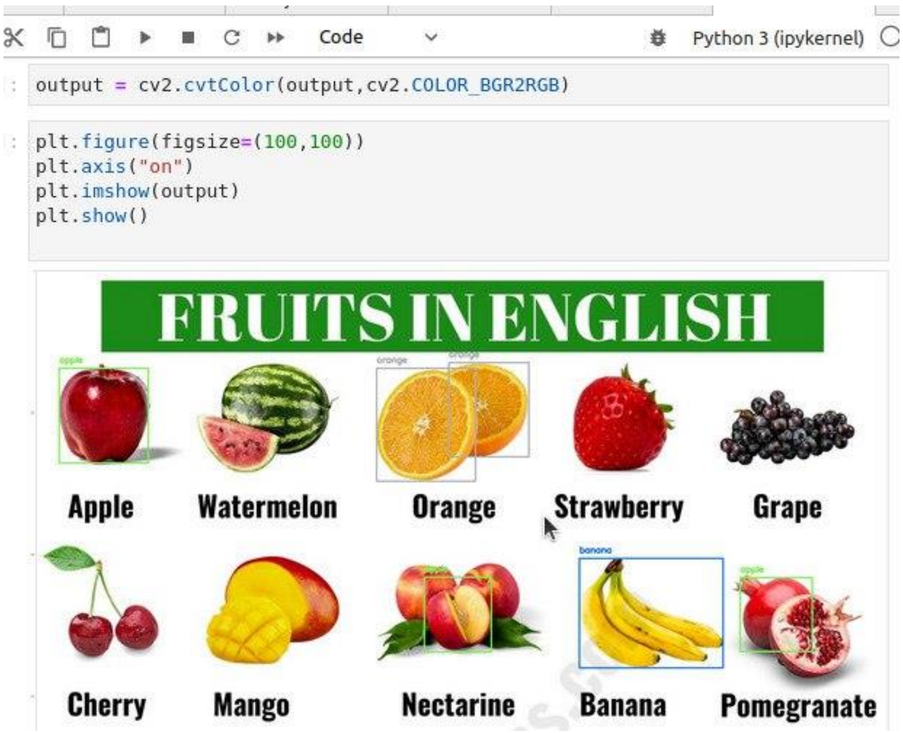


Figure 14 - Finding the same object form different images by Image Processing

And also, the two largest are TensorFlow and OpenCV, which I attempted to run on Jupyter and install; this application requires a lot of GPU and a lot of CPU, so it will be a good test for the HPC.



```
output = cv2.cvtColor(output,cv2.COLOR_BGR2RGB)
```

```
plt.figure(figsize=(100,100))
plt.axis("on")
plt.imshow(output)
plt.show()
```

Figure 15 - Finding objects from an image by Image Proccessing

Because the HPC servers were being repaired while this task was being completed, it was carried out on a local host. However, the same results were obtained when these codes were also run on the HPC after it had been started.

The astounding thing, though, was how quickly HPC ran these codes. In a separate task for the report's continuation, I looked into the speed of program execution on localhost versus HPC.

2.10 Matlab environment:

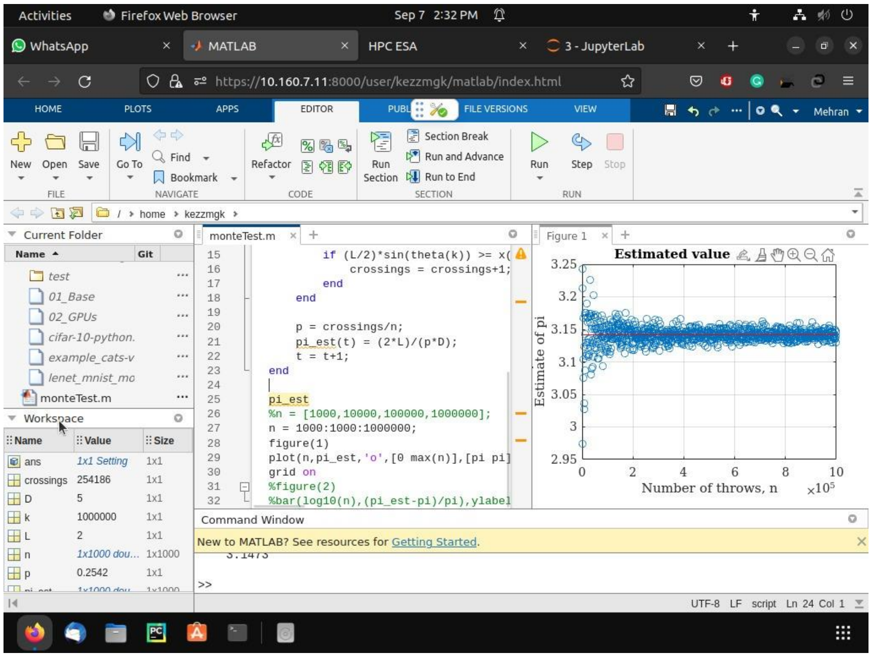
- Devices or systems used for performing the tasks: Web/Windows 10.
- Given timeframe to complete the tasks: one week, from week 8 until week 9.
- Problems faced during the execution of the tasks: Completely new environment and new programing language, and it was too slow for running on the web.

- How problems were solved: Learning the Matlab coding language and gaining experience.
- Relation of this task to passed courses at the university: Because Matlab works with plenty of matrixes connections to the Computing Mathematics course (WIX1001), and because Matlab had a new programing language it connected to and Functional and Logic programing course(WID3001).

In this task, I had to run some well-known algorithms on MatLab in both local and HPC systems and provide a report about them as part of my work.

I initially began working with Matlab in online mode. I signed up for a MatLab account under the Nottingham login "kezzmgk”

I used MatLab to run "Monte Carlo Simulation" and gained a lot of expertise using the environment, functions, and codes of the program. I also wrote a comprehensive report on the subject.

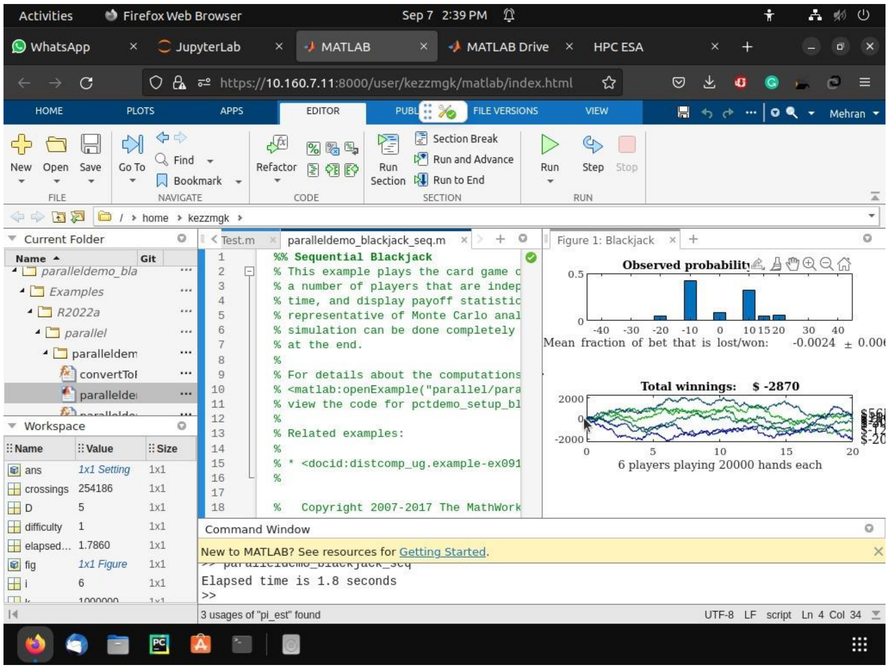


```
            if (L/2)*sin(theta(k)) >= x(
                crossings = crossings+1;
            end
        end

        p = crossings/n;
        pi_est(t) = (2*L)/(p*D);
        t = t+1;
    end

    pi_est
    %n = [1000,10000,100000,1000000];
    n = 1000:1000:1000000;
    figure(1)
    plot(n,pi_est,'o',[0 max(n)],[pi pi]
    grid on
    %figure(2)
    %bar(log10(n),(pi_est-pi)/pi),ylabel
```

Name	Value	Size
ans	1x1 Setting	1x1
crossings	254186	1x1
D	5	1x1
k	1000000	1x1
L	2	1x1
n	1x1000 dou…	1x1000
p	0.2542	1x1

```
New to MATLAB? See resources for Getting Started.
    3.1473

>>
```

Figure 16 - Running Monte Carlo Simulation on the MatLab



```
%% Sequential Blackjack
% This example plays the card game c
% a number of players that are indep
% time, and display payoff statistic
% representative of Monte Carlo anal
% simulation can be done completely
% at the end.
%
% For details about the computations
% <matlab:openExample("parallel/para
% view the code for pctdemo_setup_bl
%
% Related examples:
%
% * <docid:distcomp_ug.example-ex091
%


%   Copyright 2007-2017 The MathWork
```

Workspace

Name	Value	Size
ans	1x1 Setting	1x1
crossings	254186	1x1
D	5	1x1
difficulty	1	1x1
elapsed...	1.7860	1x1
fig	1x1 Figure	1x1
i	6	1x1

New to MATLAB? See resources for Getting Started.

```
>> paralleldemo_blackjack_seq
Elapsed time is 1.8 seconds
>>
```

3 usages of "pi_est" found

Figure 17 - Visualizing the output of Matlab calculation

I also used HPC's MatLab to create a few 3D graphs, which clearly demonstrate the program's capability.

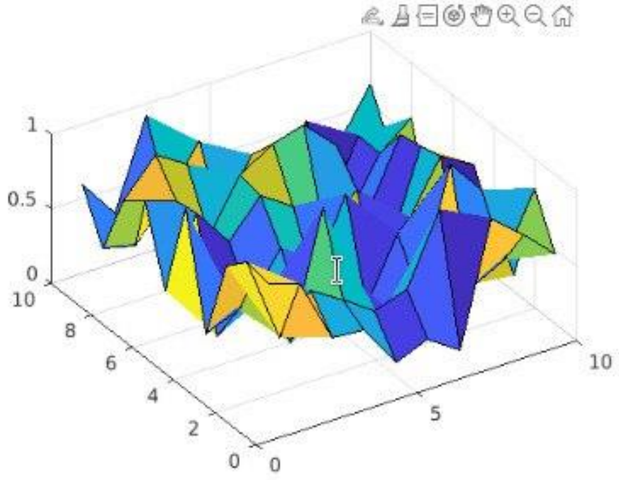


Figure 18 - 3D Graff created by MatLab

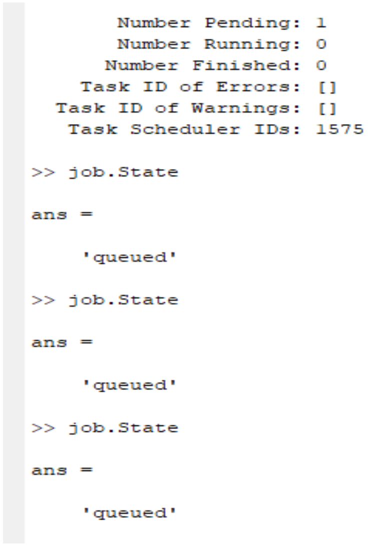
I created a whole complete paper on this subject which I've included as appendix B

2.11 Matlab Job Allocation:

- Devices or systems used for performing the tasks: Web/Windows 10.
- Given timeframe to complete the tasks: one week, from week 9 until week 12.
- Problems faced during the execution of the tasks: Matlab cluster cannot connect to the HPC compute node and jobs cannot allocate, the Matlab license is not working.
- How problems were solved: open a new physical port on the HPC server only for Matlab cluster usage and connecting to the Matlab administrator and ask them to send us a new license.
- Relation of this task to passed courses at the university: Because Matlab works with a lot of matrixes connections to the Computing Mathematics course (WIX1001), and because of the working with license, it is connected to the Software Modeling course (WIA2002), and because Matlab had a new programing language it connected to and Functional and Logic programing course(WID3001).

On this task, my activity was about “Batch”, “Cluster”, and “Job Allocating” on MatLab. The main problem was that Matlab could not use the HPC resources for job allocation. And my task was to figure out the reason for this problem and solve the problem.

I watched the recorded meeting between the MatLab administrator and UNM technical team several times, and I tried to follow the steps exactly the same as the Matlab administrator; I downloaded the provided files from the MatLab team and ran the “configCluster.m” exactly the same as the instructions; The Cluster was installed properly and I was able to connect to UNM Parallel Cluster “ispm R2022a”, but every time that I created a job and gave it some tasks, it confronted errors.

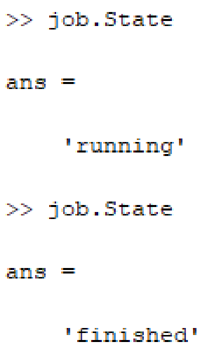


```
       Number Pending: 1
       Number Running: 0
      Number Finished: 0
    Task ID of Errors: []
  Task ID of Warnings: []
   Task Scheduler IDs: 1575

>> job.State

ans =

    'queued'

>> job.State

ans =

    'queued'

>> job.State

ans =

    'queued'
```

Figure 19 - The state of submitted jobs which did not work

The reason for this problem was the bug existed in the "ispm R2022a Cluster," which was offered by MatLab teams.

I performed each of the above steps on the "Local Cluster" to demonstrate that UNM codes were error-free.

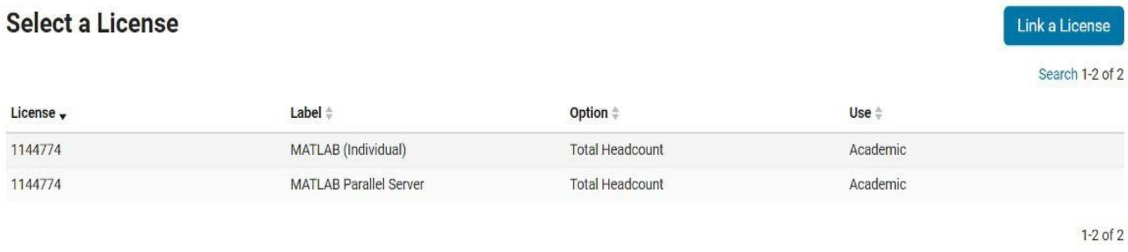


```
>> job.State

ans =

    'running'

>> job.State

ans =

    'finished'
```

Figure 20 - The state of submitted jobs which was finished

I installed Matlab on my computer so I could use it completely. Therefore, I had complete access to Matlab and could update and verify the license information.

Then, middle of installing Matlab, I Face a second Matlab License which was about “Matlab Parallel Server License”.



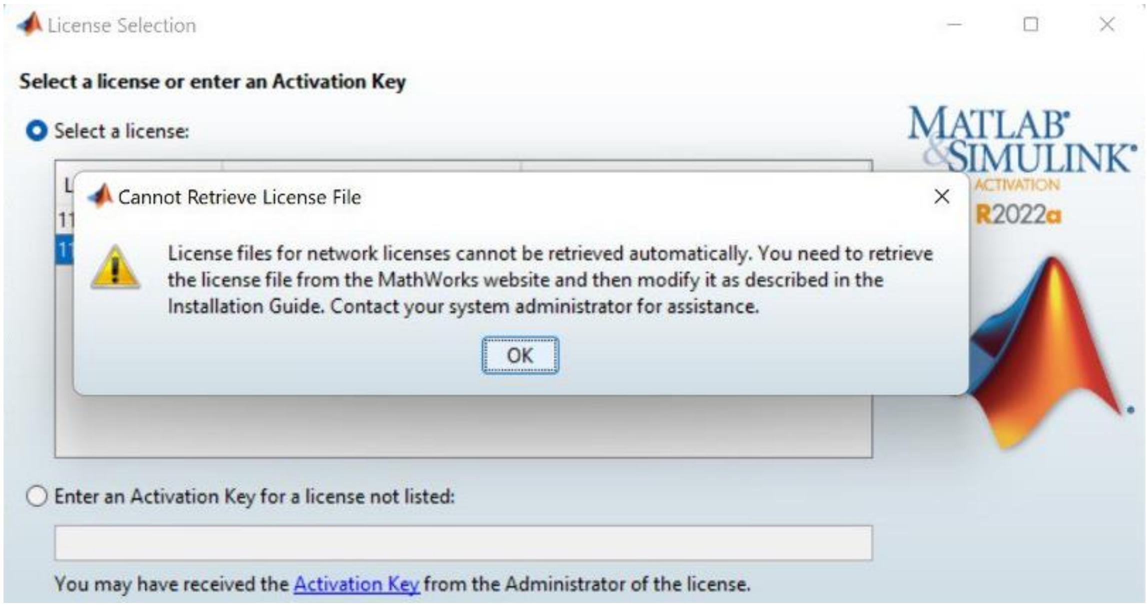
Select a License

License	Label	Option	Use
1144774	MATLAB (Individual)	Total Headcount	Academic
1144774	MATLAB Parallel Server	Total Headcount	Academic

Figure 21 - MatLab licenses for the University of Nottingham

and after that, I tried to Active that license because the given error from Matlab was about the license and so I found a relationship between these 2 reasons.

When I tried to Active that license, I faced a strange error:



License Selection

Select a license or enter an Activation Key

● Select a license:

Cannot Retrieve License File

License files for network licenses cannot be retrieved automatically. You need to retrieve the license file from the MathWorks website and then modify it as described in the Installation Guide. Contact your system administrator for assistance.

○ Enter an Activation Key for a license not listed:

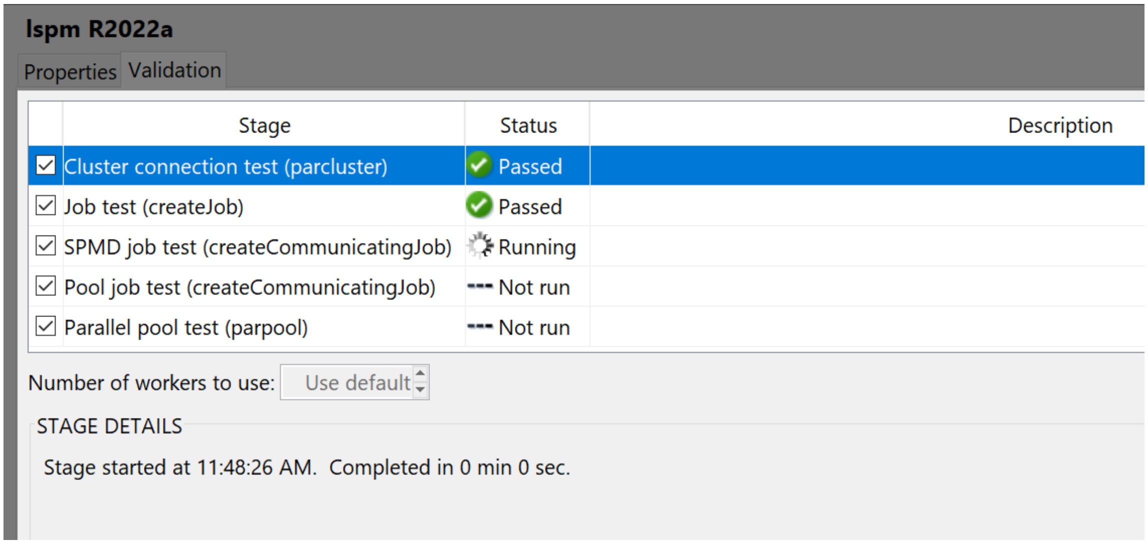
You may have received the Activation Key from the Administrator of the license.

Figure 22 - MatLab error for acitvating the license

After communicating with the Matlab administrator, and giving her some information, she checked our license and she realized that the Matlab license for parallel usage is expired, so she created a new license file for UNM.

So, by fixing the license everything was working except Cluster Validation, which would stop on the “SPMD job Test”.

However, I found a solution to this problem as well, after changing the number of CPU workers, I could submit jobs by using pools that work properly.



	Stage	Status	Description
☑	Cluster connection test (parcluster)	✓ Passed	
☑	Job test (createJob)	✓ Passed	
☑	SPMD job test (createCommunicatingJob)	Running	
☑	Pool job test (createCommunicatingJob)	--- Not run	
☑	Parallel pool test (parpool)	--- Not run	

Number of workers to use: Use default

STAGE DETAILS

Stage started at 11:48:26 AM. Completed in 0 min 0 sec.

Figure 23 - MatLab job test is passed

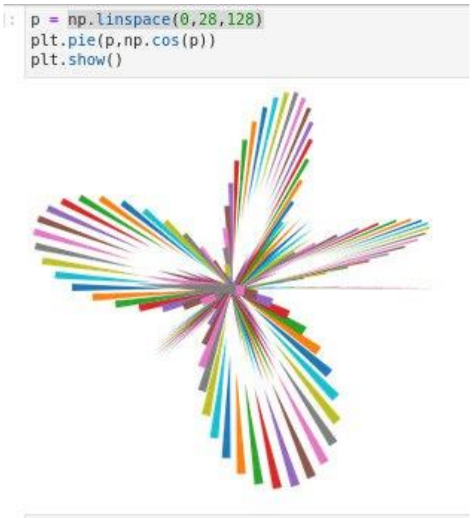
Therefore, the Matlab jobs were registered and processed in the HPC’s Compute node and this task got finished for me successfully.

2.12 Image Processing on HPC:

- Devices or systems used for performing the tasks: ubuntu 20.04.
- Given timeframe to complete the tasks: one week, from week 9 until week 12.
- Problems faced during the execution of the tasks: Using GPU for running AI Python codes was complicated, and several new packages were required for running image processing in HPC.
- How problems were solved: Learning the codes and solutions for finding the best option, testing them on the HPC, and submitting them to my supervisor
- Relation of this task to passed courses at the university: Because of using the AI algorithm, it is related to Algorithm Design and Analysis course (WIA2005), and because using python is connected to and Functional and Logic programing course(WID3001), and because running AI codes in Python is so much related to the Fundamental of Artificial Intelligence course (WIA1004).

My objective was to verify the HPC's functionality for executing AI programs, and image processing was one of those AI codes.

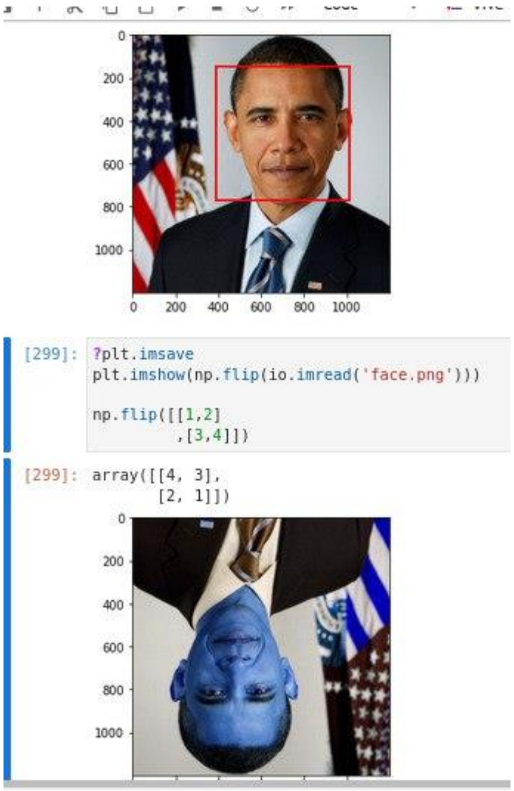
To demonstrate its use, I conducted some image processing using NumPy on HPC and even attempted to test a MatLab Graph Function on the Jupyter:



```
p = np.linspace(0,28,128)
plt.pie(p,np.cos(p))
plt.show()
```

Figure 24 - 3D graff created in Python and Jupyter Lab by using NumPy

Also, face recognition by image processing was tested in the HPC environment:



```
[299]: ?plt.imsave
       plt.imshow(np.flip(io.imread('face.png')))

       np.flip([[1,2]
               ,[3,4]])
```

```
[299]: array([[4, 3],
              [2, 1]])
```

Figure 25 - Face Recognition and changing images by NumPy

The previous image processing is independent of TensorFlow or OpenCV-2 and is only based on NumPy.

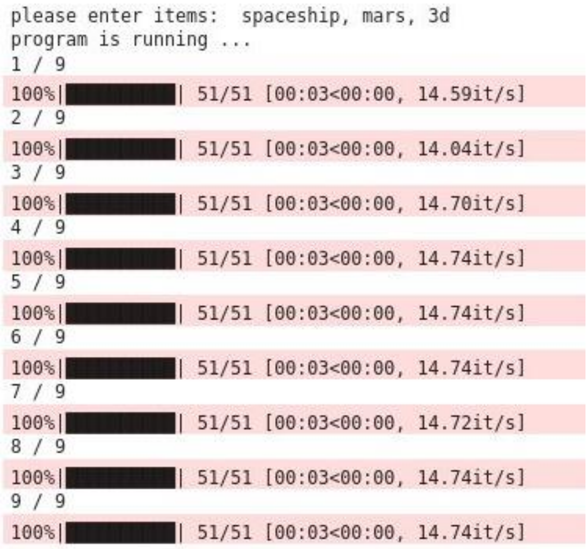
Also, I finished Image generation by using Artificial Intelligence which used a lot of Graphic cards, and the HPC system was able to easily handle that.
for this application, I used some python packages, like “Tkinter”, “Pillow”, “Troch”, and “Diffusers”.

Tkinter is for window creation which is not used on the HPC. And Pillow is used for saving and showing the image created by Torch and Diffusers.

Torch is a program for Deep learning and Diffusers also use Torch to generate an image.

In this program, a user provides some words to the program, and then the program based on that words by using Diffusers and Torch, the program generates some images. And 9 images will generate and usually one of them is the same as their expectation.
So, you can see some results of this program (the words were “spaceship”, “mars”, “3d” and “Jupiter”, “ice cream”).

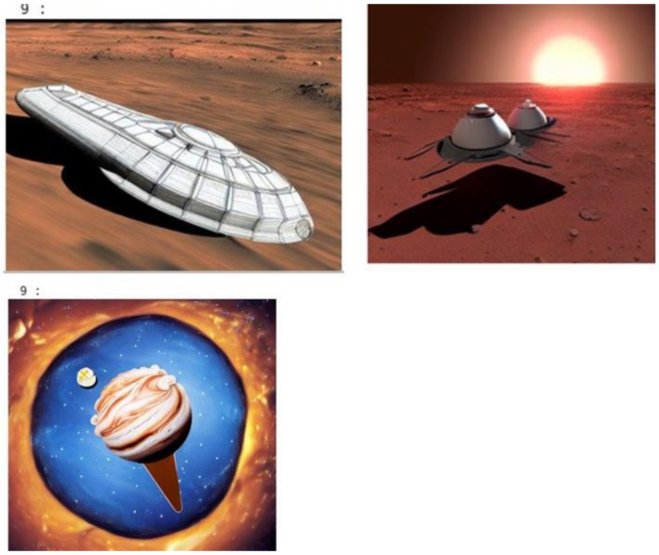
This image will be generated and after that, they will show it to the users:



```
please enter items:  spaceship, mars, 3d
program is running ...
1 / 9
100%|██████████| 51/51 [00:03<00:00, 14.59it/s]
2 / 9
100%|██████████| 51/51 [00:03<00:00, 14.04it/s]
3 / 9
100%|██████████| 51/51 [00:03<00:00, 14.70it/s]
4 / 9
100%|██████████| 51/51 [00:03<00:00, 14.74it/s]
5 / 9
100%|██████████| 51/51 [00:03<00:00, 14.74it/s]
6 / 9
100%|██████████| 51/51 [00:03<00:00, 14.74it/s]
7 / 9
100%|██████████| 51/51 [00:03<00:00, 14.72it/s]
8 / 9
100%|██████████| 51/51 [00:03<00:00, 14.74it/s]
9 / 9
100%|██████████| 51/51 [00:03<00:00, 14.74it/s]
```

Figure 26 – Creating nine images based on given words

And these are the result:

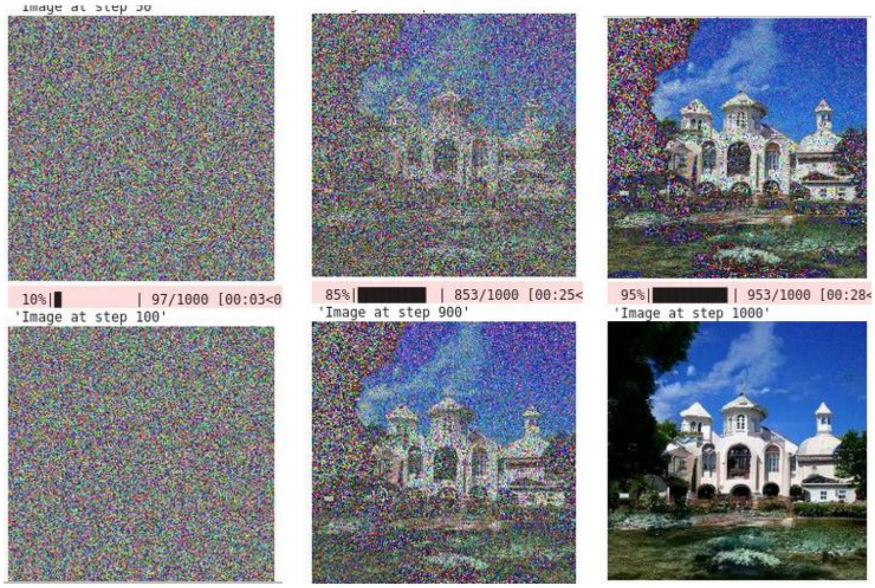


```
9 :
9 :
```

Figure 27 - Some generated image by AI

The approach to generating these images via Diffusers is by reordering the pixels to make sensible images.

In these pictures, you can see how Diffusers work



```
'Image at step 50'
10%|█          | 97/1000 [00:03<0
'Image at step 100'
85%|████████   | 853/1000 [00:25<
'Image at step 900'
95%|█████████  | 953/1000 [00:28<
'Image at step 1000'
```

Figure 28 - The process of image generating based on AI

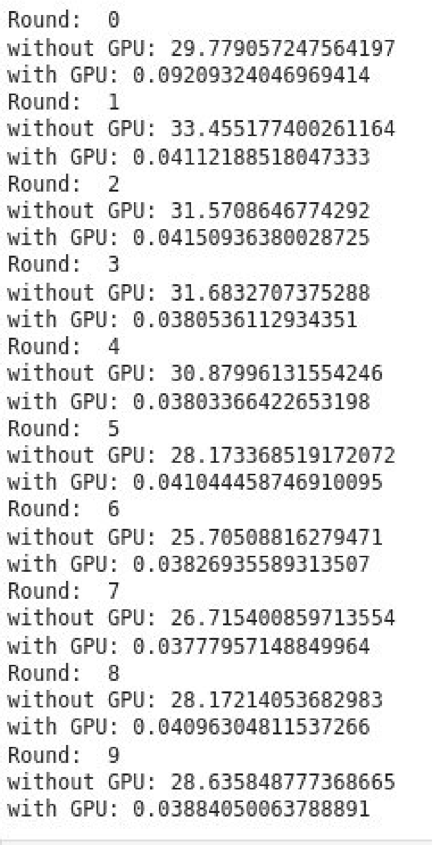
2.13 GPU vs CPU in HPC:

- Devices or systems used for performing the tasks: Ubuntu 20.04.
- Given timeframe to complete the tasks: One week, from week 11 until week 12.
- Problems faced during the execution of the tasks: Using GPU in the Python program on the IDE was needed CUDA which was tricky to install.
- How problems were solved: Follow Nvidia instructions for installing CUDA and run the codes.
- Relation of this task to passed courses at the university: Because running AI codes in Python is so much related to the Fundamental of Artificial Intelligence course (WIA1004), and Functional and Logic programing course (WID3001).

The launched HPC system has several CPUs and graphics cards that are designed to run jobs powerfully and at high speed, in this report, the real performance of the processors of this system is challenged, and the differences and the proper placement of CPU and GPU have been tested and significant results achieved.

An identical function that measures the CPU and GPU under the same circumstances is necessary to test these two components.

The method used in this experiment is a 100 million circulation loop whose function is to fill an array with the same value. This method runs twice, once with CPU components and again using GPU, then the time for these components for executing this method gets measured. So, the performance speed of these two components can be calculated with very high accuracy.

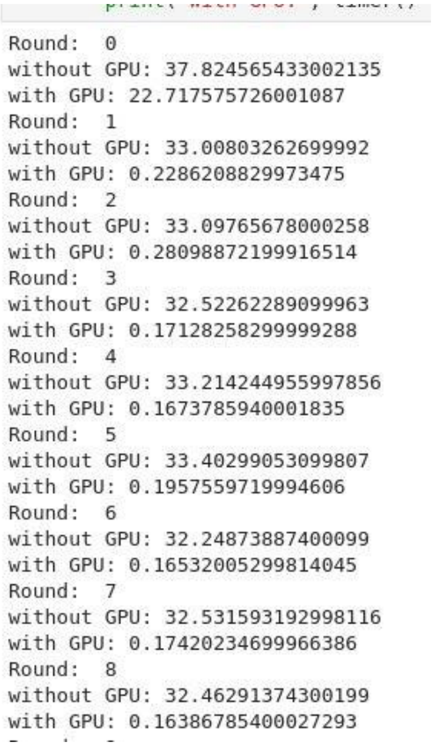


```
Round:  0
without GPU: 29.779057247564197
with GPU: 0.09209324046969414
Round:  1
without GPU: 33.455177400261164
with GPU: 0.04112188518047333
Round:  2
without GPU: 31.5708646774292
with GPU: 0.04150936380028725
Round:  3
without GPU: 31.6832707375288
with GPU: 0.0380536112934351
Round:  4
without GPU: 30.87996131554246
with GPU: 0.03803366422653198
Round:  5
without GPU: 28.173368519172072
with GPU: 0.041044458746910095
Round:  6
without GPU: 25.70508816279471
with GPU: 0.03826935589313507
Round:  7
without GPU: 26.715400859713554
with GPU: 0.03777957148849964
Round:  8
without GPU: 28.17214053682983
with GPU: 0.04096304811537266
Round:  9
without GPU: 28.635848777368665
with GPU: 0.03884050063788891
```

Figure 29 - The result of the CPU vs GPU test on the HPC

We understand for this test, GPU is 1000 times faster than the CPU.

The following results are for the local PC and show that HPC can be 20 times faster than the local pc:



```
Round:  0
without GPU: 37.824565433002135
with GPU: 22.717575726001087
Round:  1
without GPU: 33.00803262699992
with GPU: 0.2286208829973475
Round:  2
without GPU: 33.09765678000258
with GPU: 0.28098872199916514
Round:  3
without GPU: 32.52262289099963
with GPU: 0.17128258299999288
Round:  4
without GPU: 33.214244955997856
with GPU: 0.1673785940001835
Round:  5
without GPU: 33.40299053099807
with GPU: 0.1957559719994606
Round:  6
without GPU: 32.24873887400099
with GPU: 0.16532005299814045
Round:  7
without GPU: 32.531593192998116
with GPU: 0.17420234699966386
Round:  8
without GPU: 32.46291374300199
with GPU: 0.16386785400027293
```

Figure 30 - The result of the CPU vs GPU test on the local PC

I created a whole complete paper on this subject which I've included as appendix A

2.14 Paraview:

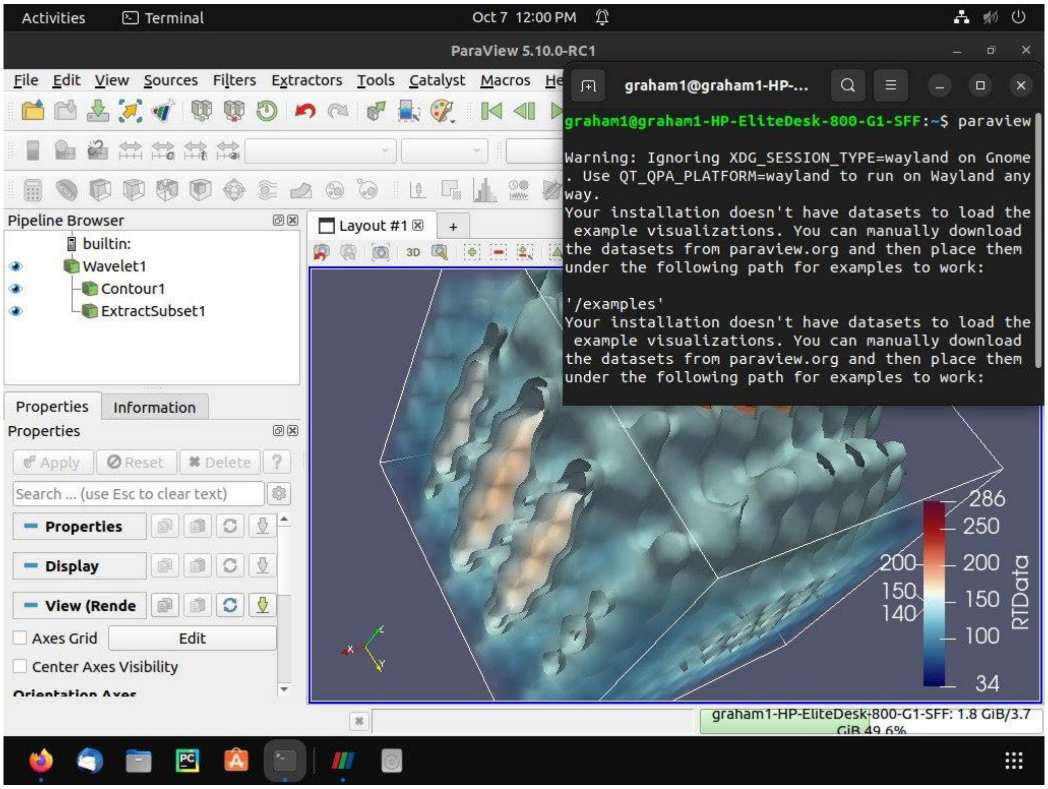
- Devices or systems used for performing the tasks: Ubuntu 20.04.
- Given timeframe to complete the tasks: one week, from week 12 until week 15.
- Problems faced during the execution of the tasks: Paraview needs to run as a server and this concept was new for me, also there were 3 versions of Paraview that make it challenging to understand and install
- How problems were solved: Investigate Paraview and learn the difference between versions and spend an online free course to learn how to install and use it.

- Relation of this task to passed courses at the university: Because of Linux environment connections to the OS course (WIA2004), and also because of working with applications and making the change on them to connect them to HPC, connections to the Software Modeling course(WIA2002).

My task was installing and running the Paraview on the test server and HPC server.

I did some research on ParaView and its use before attempting to install it on a local PC and a test server. there are three different versions of Paraview which are Paraview local, Paraview Web, and Paraview Server.

It only took one command to install the offline version of ParaView on the local PC, and as you can see, it worked perfectly and allowed me to run some simulations.

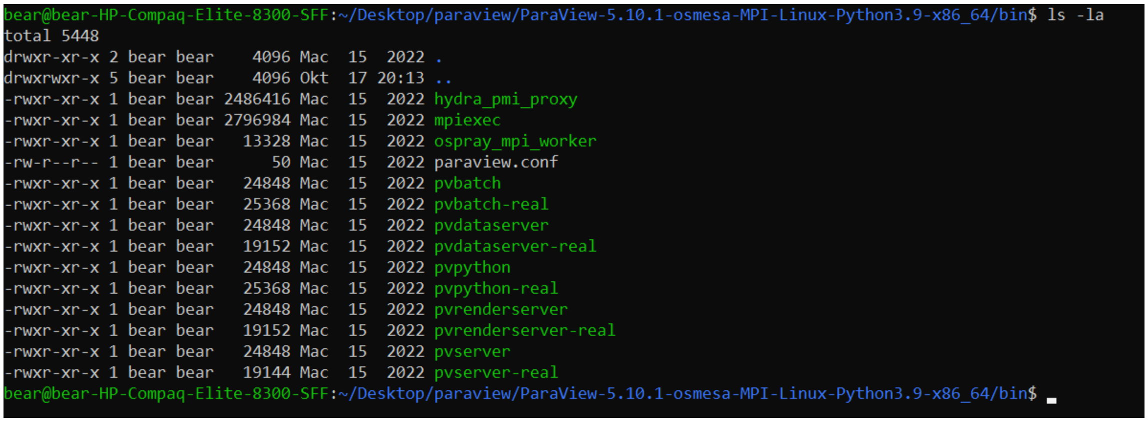


```
graham1@graham1-HP-EliteDesk-800-G1-SFF:~$ paraview

Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome
. Use QT_QPA_PLATFORM=wayland to run on Wayland any
way.
Your installation doesn't have datasets to load the
 example visualizations. You can manually download
the datasets from paraview.org and then place them
under the following path for examples to work:

'/examples'
Your installation doesn't have datasets to load the
 example visualizations. You can manually download
the datasets from paraview.org and then place them
under the following path for examples to work:
```

Figure 31 - Running the Paraview local and a simulation on it

After that, I tried to install it as a server and connect clients to it. The primary problem for me in running ParaView on the test server is that there is no effective tutorial or instruction to follow.

for running Paraview Server, I first execute the two server-related files after downloading the Paraview.tar.gz file from the official Paraview website.

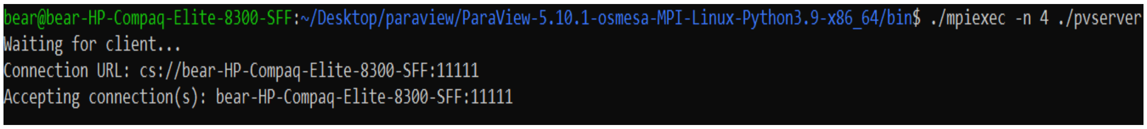


```
bear@bear-HP-Compaq-Elite-8300-SFF:~/Desktop/paraview/ParaView-5.10.1-osmesa-MPI-Linux-Python3.9-x86_64/bin$ ls -la
total 5448
drwxr-xr-x 2 bear bear    4096 Mac  15  2022 .
drwxrwxr-x 5 bear bear    4096 Okt  17 20:13 ..
-rwxr-xr-x 1 bear bear 2486416 Mac  15  2022 hydra_pmi_proxy
-rwxr-xr-x 1 bear bear 2796984 Mac  15  2022 mpiexec
-rwxr-xr-x 1 bear bear   13328 Mac  15  2022 ospray_mpi_worker
-rw-r--r-- 1 bear bear      50 Mac  15  2022 paraview.conf
-rwxr-xr-x 1 bear bear   24848 Mac  15  2022 pvbatch
-rwxr-xr-x 1 bear bear   25368 Mac  15  2022 pvbatch-real
-rwxr-xr-x 1 bear bear   24848 Mac  15  2022 pvdataserver
-rwxr-xr-x 1 bear bear   19152 Mac  15  2022 pvdataserver-real
-rwxr-xr-x 1 bear bear   24848 Mac  15  2022 pvpython
-rwxr-xr-x 1 bear bear   25368 Mac  15  2022 pvpython-real
-rwxr-xr-x 1 bear bear   24848 Mac  15  2022 pvrenderserver
-rwxr-xr-x 1 bear bear   19152 Mac  15  2022 pvrenderserver-real
-rwxr-xr-x 1 bear bear   24848 Mac  15  2022 pvserver
-rwxr-xr-x 1 bear bear   19144 Mac  15  2022 pvserver-real
bear@bear-HP-Compaq-Elite-8300-SFF:~/Desktop/paraview/ParaView-5.10.1-osmesa-MPI-Linux-Python3.9-x86_64/bin$
```

Figure 32 - The files in the Paraview package

The code that I use for running the Paraview server is:

“. /mpiexec -n 4 ./pvserver”

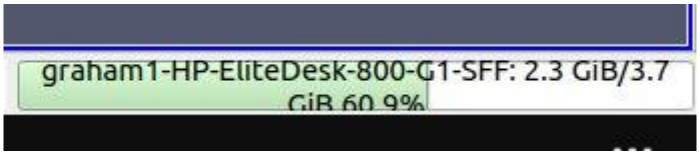


```
bear@bear-HP-Compaq-Elite-8300-SFF:~/Desktop/paraview/ParaView-5.10.1-osmesa-MPI-Linux-Python3.9-x86_64/bin$ ./mpiexec -n 4 ./pvserver
Waiting for client...
Connection URL: cs://bear-HP-Compaq-Elite-8300-SFF:11111
Accepting connection(s): bear-HP-Compaq-Elite-8300-SFF:11111
```

Figure 33 - Paraview server is ready for clients

The server is now available, and we can connect to it and utilize its resources by using the Paraview Client app.

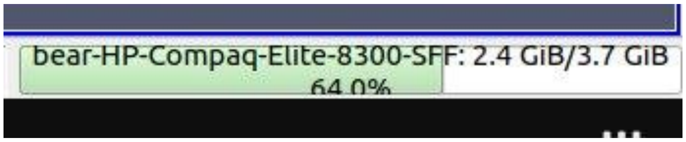
Paraview before connecting to the server (local):



graham1-HP-EliteDesk-800-G1-SFF: 2.3 GiB/3.7 GiB 60.9%

Figure 34 - Paraview status in local PC

Paraview after connecting to the server:



bear-HP-Compaq-Elite-8300-SFF: 2.4 GiB/3.7 GiB
64.0%

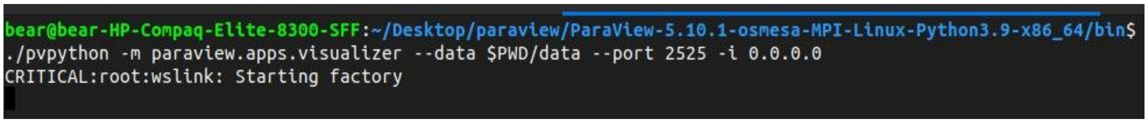
Figure 35 - Paraview status after connecting to the test server

As you can see, the Paraview is connected to the server and uses server resources for running the jobs.

I then launched Paraview Web on the test server and connected to it using a remote.

The code that I used is:

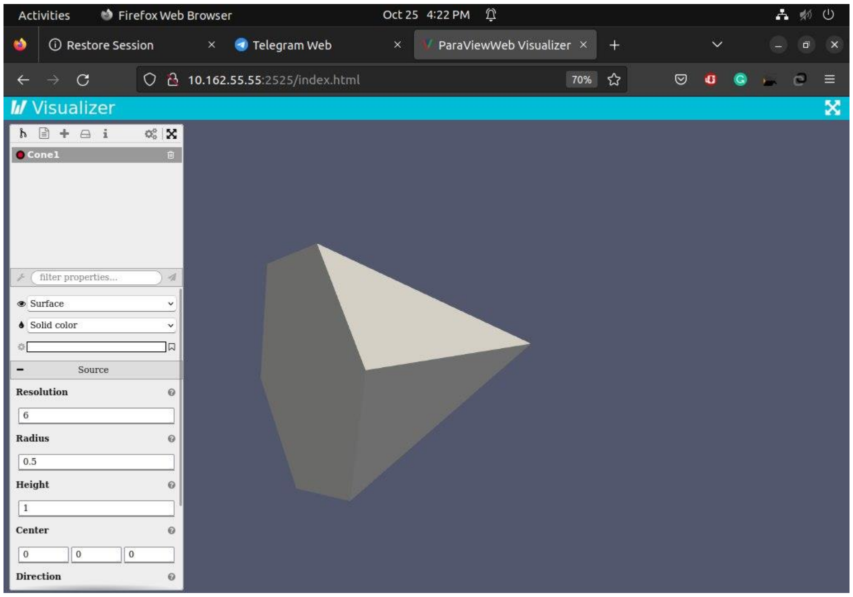
“. /pvpython -m paraview.apps.visualizer --data $PWD/data --port 2525 -i 0.0.0.0”



```
bear@bear-HP-Compaq-Elite-8300-SFF:~/Desktop/paraview/ParaView-5.10.1-osmesa-MPI-Linux-Python3.9-x86_64/bin$
./pvpython -m paraview.apps.visualizer --data $PWD/data --port 2525 -i 0.0.0.0
CRITICAL:root:wslink: Starting factory
```

Figure 36 - Paraview Web is ready for clients

Anyone may now visit the Paraview Web Server by using the test server IP address and port 2525.



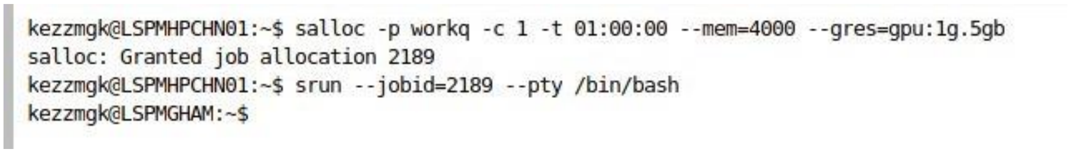
Resolution: 6
Radius: 0.5
Height: 1
Center: 0 0 0
Direction

Figure 37 - Paraview web interface

Thus, anyone can access Paraview without having to locally install it.

For running Paraview on the HPC, first, I upload Paraview files to the HPC file management similarly to a local computer. Next, I use the HPC to run Paraview Web and Paraview Server.

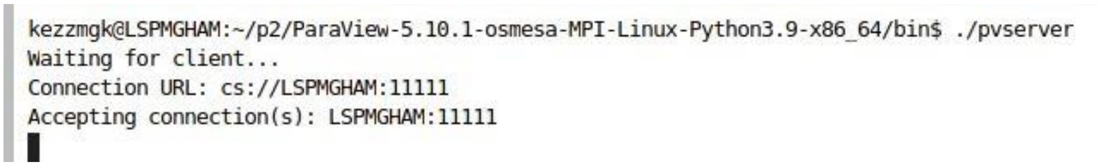
then, I make a server and have access to it by making a task and using "srun" to gain access to the compute node (via application or browser).



```
kezzmgk@LSPMHPCHN01:~$ salloc -p workq -c 1 -t 01:00:00 --mem=4000 --gres=gpu:1g.5gb
salloc: Granted job allocation 2189
kezzmgk@LSPMHPCHN01:~$ srun --jobid=2189 --pty /bin/bash
kezzmgk@LSPMGHAM:~$
```

Figure 38 - Connecting to the HPC Compute Node by using "srun"

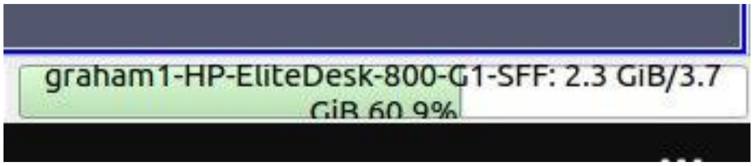
And then we start the Paraview server here



```
kezzmgk@LSPMGHAM:~/p2/ParaView-5.10.1-osmesa-MPI-Linux-Python3.9-x86_64/bin$ ./pvserver
Waiting for client...
Connection URL: cs://LSPMGHAM:11111
Accepting connection(s): LSPMGHAM:11111
```

Figure 39 - Paraview Server is running on the HPC

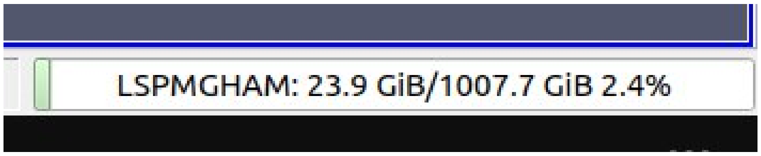
Paraview before connecting to the server (local):



graham1-HP-EliteDesk-800-G1-SFF: 2.3 GiB/3.7 GiB 60.9%

Figure 40 - Paraview status in local PC

Paraview after connecting to the HPC server:



LSPMGHAM: 23.9 GiB/1007.7 GiB 2.4%

Figure 41 - Paraview status after connecting to the HPC server

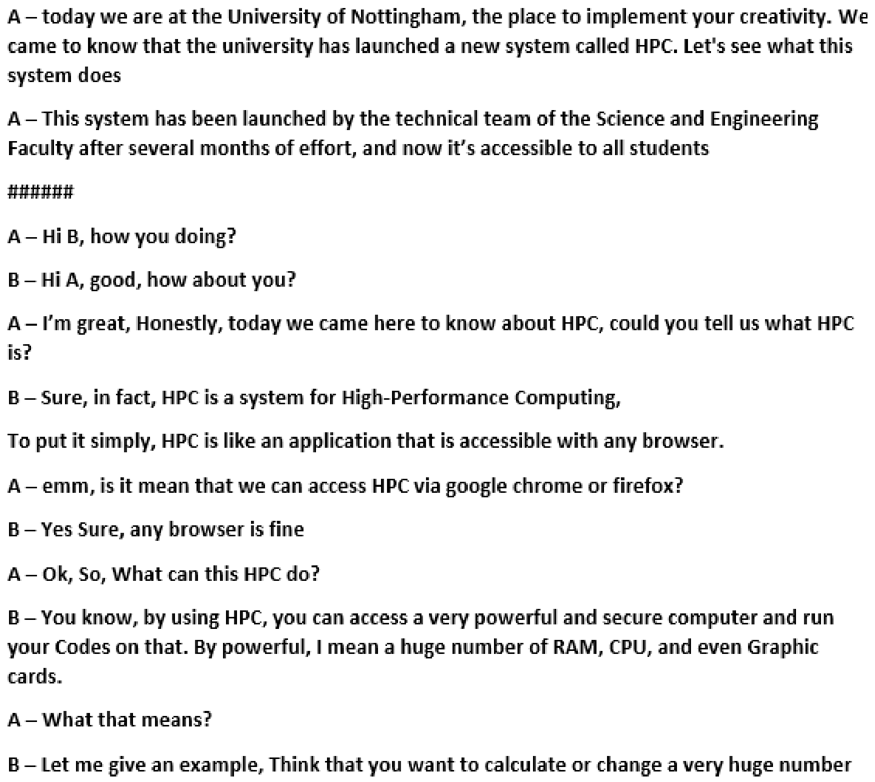
Also, we can run Paraview Web on the Compute Node. This again is exactly the same as the Test Server.

2.15 HPC Script:

- Devices or systems used for performing the tasks: Ubuntu 20.04.
- Given timeframe to complete the tasks: one week, from week 13 until week 15.
- Problems faced during the execution of the tasks: Writing a paper that shows all the features of a system was very hard, I had to learn how to write an efficient script
- How problems were solved: Review some of the university courses and giving advice from my supervisor
- Relation of this task to passed courses at the university: Because of writing information about HPC, it was related to the Information Literacy course (GIG1004), Information System course (WIA1001), and also Knowledge Representation and Reasoning (WID2001).

A team got in charge to create an advertising video for the HPC, and as I was familiar with the system, my new task was writing the script for the HPC system.

This was more of a scenario, which I presented to the team in order to help them come up with a concept.



A – today we are at the University of Nottingham, the place to implement your creativity. We came to know that the university has launched a new system called HPC. Let's see what this system does

A – This system has been launched by the technical team of the Science and Engineering Faculty after several months of effort, and now it's accessible to all students

######

A – Hi B, how you doing?

B – Hi A, good, how about you?

A – I'm great, Honestly, today we came here to know about HPC, could you tell us what HPC is?

B – Sure, in fact, HPC is a system for High-Performance Computing,

To put it simply, HPC is like an application that is accessible with any browser.

A – emm, is it mean that we can access HPC via google chrome or firefox?

B – Yes Sure, any browser is fine

A – Ok, So, What can this HPC do?

B – You know, by using HPC, you can access a very powerful and secure computer and run your Codes on that. By powerful, I mean a huge number of RAM, CPU, and even Graphic cards.

A – What that means?

B – Let me give an example, Think that you want to calculate or change a very huge number

Figure 42 - A part of the HPC Script

They requested certain files on HPC after our meeting, so I prepared the materials for them

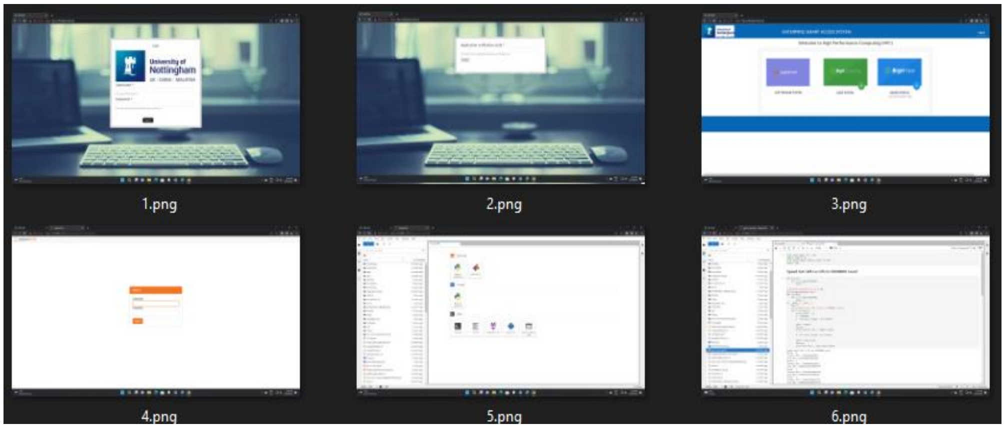


Figure 43 - HPC Snapshots for creating advertising video

After all, was said and done, the advertising team produced a brief commercial using the script I provided.

2.16 Creating a new HPC (installing SLURM):

- Devices or systems used for performing the tasks: Ubuntu 22.04.
- Given timeframe to complete the tasks: one week, from week 15 until week 20.
- Problems faced during the execution of the tasks: Lack of sufficient information for installing and using SLURM and connecting the work area with the hardware.
- How problems were solved: By using UNM HPC as a model and reading some papers for understanding the SLURM and how it is working.
- Relation of this task to passed courses at the university: Because of reasoning the issues and analyzing them to find the best answer, knowledge representing and reasoning course (WID2001), because of Linux environment connections to the OS course (WIA2004), because of dividing computer resources for using on SLURM, it is related to the Computer System Architecture course (WIA3001), and because of a lot of interactions with IP and Networks so it was related to the Network Technology course (WIA1005).

In this task, I attempted to set up and run brand-new HPC systems on the test server while enabling user access.

for this purpose, I install Jupyter Hub on the test server, and it has exactly the same appearance as Jupyter Hub for HPC.

After installing the Jupyter Hub, we can run this as a server, and define a port for using it. This is multiuser, and the user of this Jupyter Hub is the same as the test server users.

In the meantime, I begin studying SLURM and how to install it because we need to install and run SLURM in order to run HPC on the test server.

This is a really helpful chart, and we can regulate SLURM using these instructions.

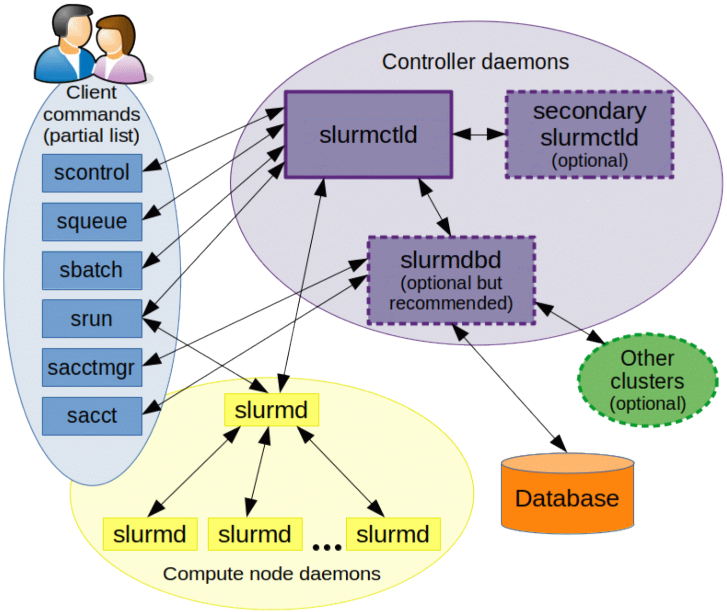
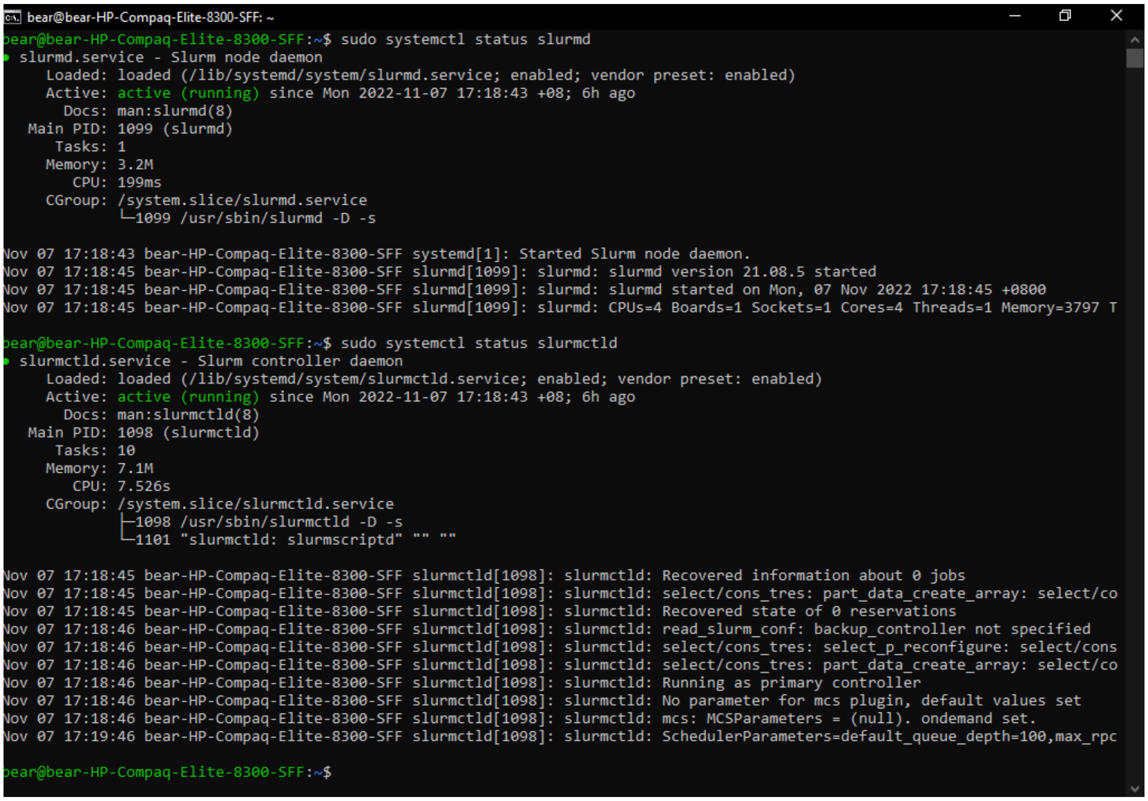


Figure 44 - SLURM components (slurm schedmd n.d.) [8].

I had to install MySQL on the test server first, which was quite difficult before I could install SLURM on it. SLURM is installed on the test server once MySQL has been set up.

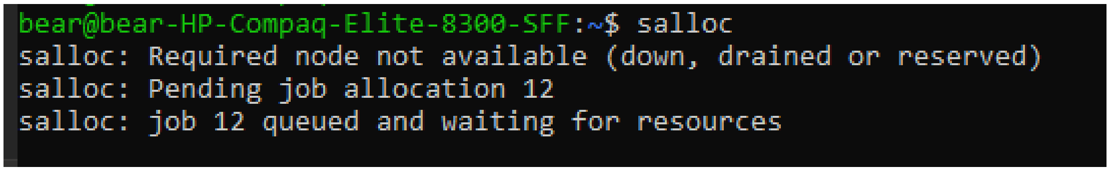
As you can see, we are running both SLURM servers (slurmd and slurmctld), therefore SLURM ought to now function. I created a configuration file as well for the servers.



```
bear@bear-HP-Compaq-Elite-8300-SFF:~$ sudo systemctl status slurmd
• slurmd.service - Slurm node daemon
     Loaded: loaded (/lib/systemd/system/slurmd.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2022-11-07 17:18:43 +08; 6h ago
       Docs: man:slurmd(8)
   Main PID: 1099 (slurmd)
      Tasks: 1
     Memory: 3.2M
        CPU: 199ms
     CGroup: /system.slice/slurmd.service
             └─1099 /usr/sbin/slurmd -D -s

Nov 07 17:18:43 bear-HP-Compaq-Elite-8300-SFF systemd[1]: Started Slurm node daemon.
Nov 07 17:18:45 bear-HP-Compaq-Elite-8300-SFF slurmd[1099]: slurmd: slurmd version 21.08.5 started
Nov 07 17:18:45 bear-HP-Compaq-Elite-8300-SFF slurmd[1099]: slurmd: slurmd started on Mon, 07 Nov 2022 17:18:45 +0800
Nov 07 17:18:45 bear-HP-Compaq-Elite-8300-SFF slurmd[1099]: slurmd: CPUs=4 Boards=1 Sockets=1 Cores=4 Threads=1 Memory=3797 T

bear@bear-HP-Compaq-Elite-8300-SFF:~$ sudo systemctl status slurmctld
• slurmctld.service - Slurm controller daemon
     Loaded: loaded (/lib/systemd/system/slurmctld.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2022-11-07 17:18:43 +08; 6h ago
       Docs: man:slurmctld(8)
   Main PID: 1098 (slurmctld)
      Tasks: 10
     Memory: 7.1M
        CPU: 7.526s
     CGroup: /system.slice/slurmctld.service
             ├─1098 /usr/sbin/slurmctld -D -s
             └─1101 "slurmctld: slurmscriptd" "" ""

Nov 07 17:18:45 bear-HP-Compaq-Elite-8300-SFF slurmctld[1098]: slurmctld: Recovered information about 0 jobs
Nov 07 17:18:45 bear-HP-Compaq-Elite-8300-SFF slurmctld[1098]: slurmctld: select/cons_tres: part_data_create_array: select/co
Nov 07 17:18:45 bear-HP-Compaq-Elite-8300-SFF slurmctld[1098]: slurmctld: Recovered state of 0 reservations
Nov 07 17:18:46 bear-HP-Compaq-Elite-8300-SFF slurmctld[1098]: slurmctld: read_slurm_conf: backup_controller not specified
Nov 07 17:18:46 bear-HP-Compaq-Elite-8300-SFF slurmctld[1098]: slurmctld: select/cons_tres: select_p_reconfigure: select/cons
Nov 07 17:18:46 bear-HP-Compaq-Elite-8300-SFF slurmctld[1098]: slurmctld: select/cons_tres: part_data_create_array: select/co
Nov 07 17:18:46 bear-HP-Compaq-Elite-8300-SFF slurmctld[1098]: slurmctld: Running as primary controller
Nov 07 17:18:46 bear-HP-Compaq-Elite-8300-SFF slurmctld[1098]: slurmctld: No parameter for mcs plugin, default values set
Nov 07 17:18:46 bear-HP-Compaq-Elite-8300-SFF slurmctld[1098]: slurmctld: mcs: MCSParameters = (null). ondemand set.
Nov 07 17:19:46 bear-HP-Compaq-Elite-8300-SFF slurmctld[1098]: slurmctld: SchedulerParameters=default_queue_depth=100,max_rpc

bear@bear-HP-Compaq-Elite-8300-SFF:~$
```

Figure 45 - SLURM Servers are running

However, there was a glitch, and when we attempted to execute a task, we encountered an odd error.



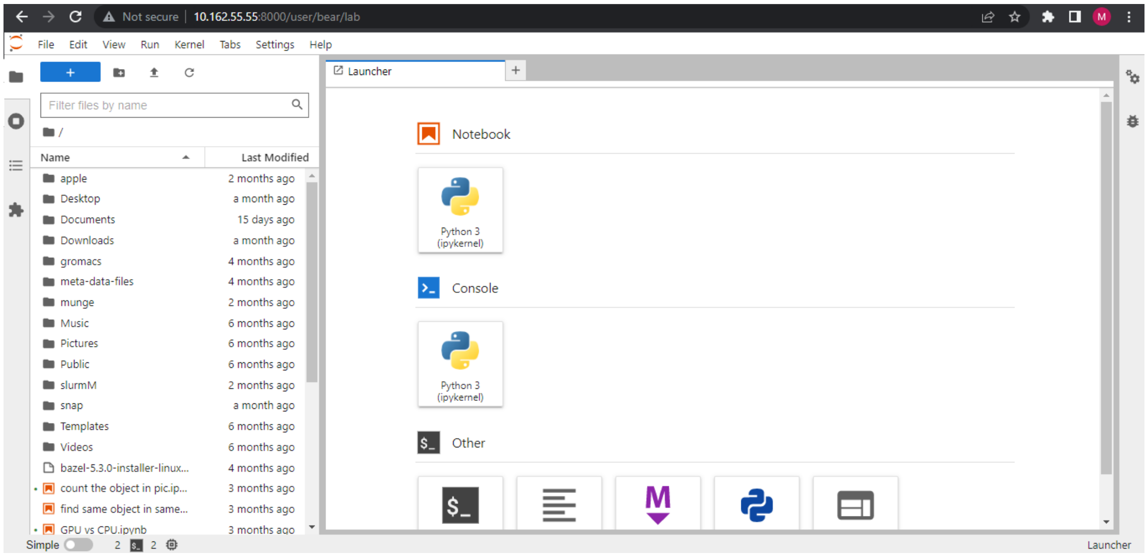
```
bear@bear-HP-Compaq-Elite-8300-SFF:~$ salloc
salloc: Required node not available (down, drained or reserved)
salloc: Pending job allocation 12
salloc: job 12 queued and waiting for resources
```

Figure 46 - The error of submitting job on SLURM

As a result of my extensive research, I came to the conclusion that the primary problem lies with SLURM's configuration file.

First, I looked for the HPC config file and compared it to the new SLURM in an effort to identify the optimal configuration file for SLURM. The new SLURM has a problem processing jobs and that had a problem when trying to reach a compute node.

I discovered that we can get to the precise information that should be included in the config file by using SLURM commands like "slurmd -C." Based on this, I was able to run SLURM on the test server, proving that it was functional and that jobs could submit and processed by the compute node. So finally we have a whole new HPC on the test server.



Name	Last Modified
apple	2 months ago
Desktop	a month ago
Documents	15 days ago
Downloads	a month ago
gromacs	4 months ago
meta-data-files	4 months ago
munge	2 months ago
Music	6 months ago
Pictures	6 months ago
Public	6 months ago
slurmM	2 months ago
snap	a month ago
Templates	6 months ago
Videos	6 months ago
bazel-5.3.0-installer-linux...	4 months ago
count the object in pic.ip...	3 months ago
find same object in same...	3 months ago
GPU vs CPU.ipynb	3 months ago

Figure 47 - The main page of Jupyter Lab and Local HPC

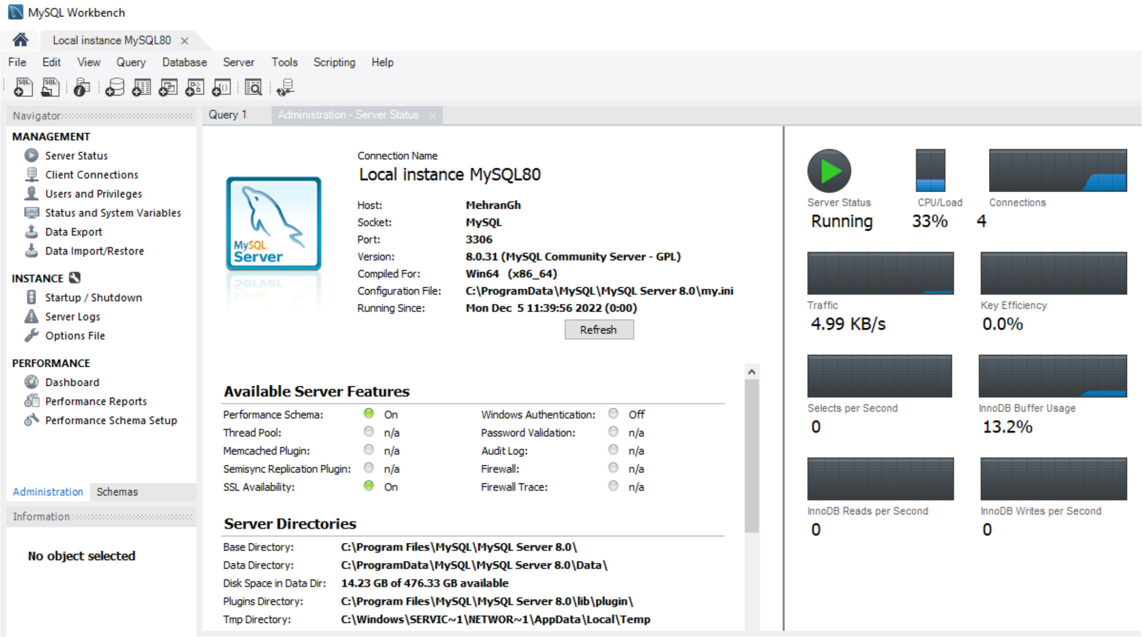
2.17 Chemical Team’s Database:

- Devices or systems used for performing the tasks: Windows 10.
- Given timeframe to complete the tasks: one week, from week 20 until week 23.
- Problems faced during the execution of the tasks: There is no MySQL Database for their tables, and there are a lot of tables and data, and the PK and FK did not exit.
- How problems were solved: Using some applications for transferring tables from MS Access to MySQL and just modifying them. Using DataGrip for creating and connecting PK and FK.

- Relation of this task to passed courses at the university: Because of working with database tables and their connection, it is related to the Data Structure course (WIA1002) and also the Database course (WIA2001)

I was given a new task to complete, which involved developing a new database system for the University of Nottingham's chemistry team. In order to use SQL codes based on the MySQL environment in real-world work, I had to learn about them and review the University of Nottingham database.

So I installed MySQL on my PC and tried to run some codes on that.



Connection Name
Local instance MySQL80

Host: MehranGh
Socket: MySQL
Port: 3306
Version: 8.0.31 (MySQL Community Server - GPL)
Compiled For: Win64 (x86_64)
Configuration File: C:\ProgramData\MySQL\MySQL Server 8.0\my.ini
Running Since: Mon Dec 5 11:39:56 2022 (0:00)

Available Server Features

Performance Schema: ● On
Thread Pool: ○ n/a
Memcached Plugin: ○ n/a
Semisync Replication Plugin: ○ n/a
SSL Availability: ● On
Windows Authentication: ○ Off
Password Validation: ○ n/a
Audit Log: ○ n/a
Firewall: ○ n/a
Firewall Trace: ○ n/a

Server Directories

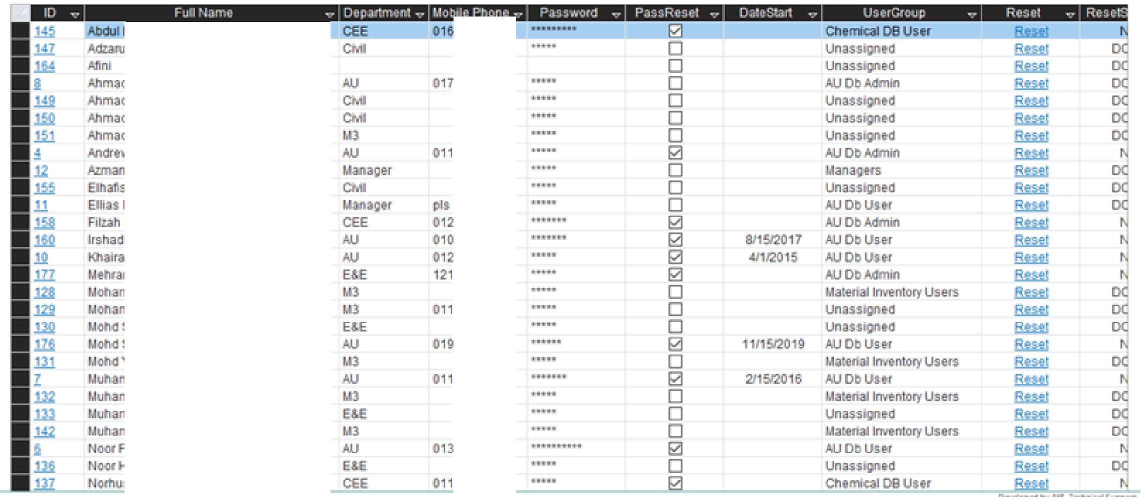
Base Directory: C:\Program Files\MySQL\MySQL Server 8.0\
Data Directory: C:\ProgramData\MySQL\MySQL Server 8.0\Data\
Disk Space in Data Dir: 14.23 GB of 476.33 GB available
Plugins Directory: C:\Program Files\MySQL\MySQL Server 8.0\lib\plugin\
Tmp Directory: C:\Windows\SERVIC~1\NETWOR~1\AppData\Local\Temp

Server Status: Running
CPU/Load: 33%
Connections: 4
Traffic: 4.99 KB/s
Key Efficiency: 0.0%
Selects per Second: 0
InnoDB Buffer Usage: 13.2%
InnoDB Reads per Second: 0
InnoDB Writes per Second: 0

Figure 48 - Running the MySQL Server on the local PC

The data for the Chemistry team is kept in Microsoft Access, but they wish to switch to MySQL. Their Microsoft Access has enormous amounts of data which store around 8 years of data.

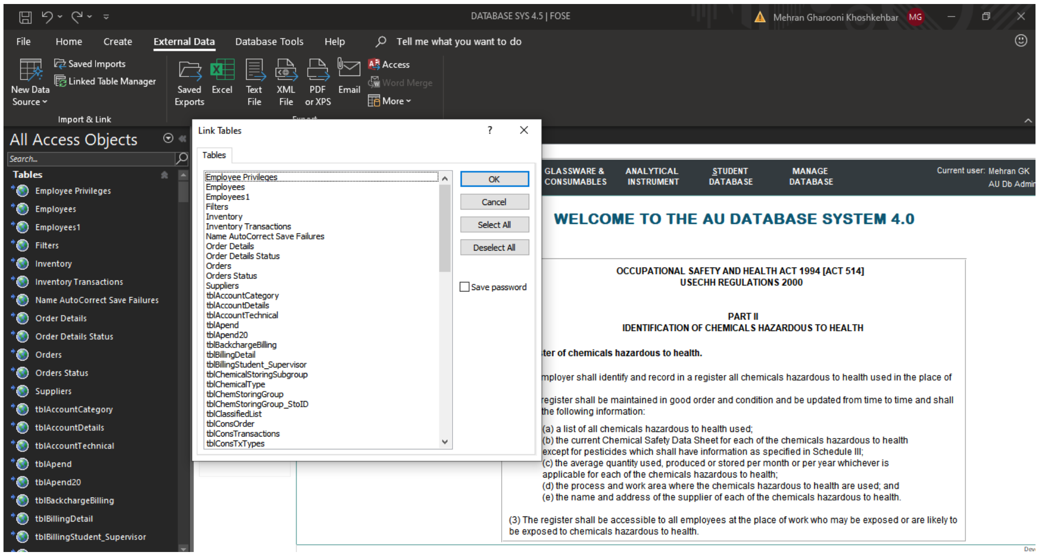
The admin of the Chemistry team gave me the Microsoft Access files so I could learn how to use MS Access databases. I must create a new MySQL database system with precisely the same data as in MS Access.



ID	Full Name	Department	Mobile Phone	Password	PassReset	DateStart	UserGroup	Reset	ResetS
145	Abdul	CEE	016	*********	☑		Chemical DB User	Reset	N
147	Adzaru	Civil		*****	☐		Unassigned	Reset	DC
164	Afini				☐		Unassigned	Reset	DC
8	Ahmad	AU	017	*****	☐		AU Db Admin	Reset	DC
149	Ahmad	Civil		*****	☐		Unassigned	Reset	DC
150	Ahmad	Civil		*****	☐		Unassigned	Reset	DC
151	Ahmad	M3		*****	☐		Unassigned	Reset	DC
4	Andrew	AU	011	*****	☑		AU Db Admin	Reset	N
12	Azman	Manager		*****	☐		Managers	Reset	DC
155	Elhafis	Civil		*****	☐		Unassigned	Reset	DC
11	Ellias	Manager	pls	*****	☐		AU Db User	Reset	DC
158	Filzah	CEE	012	*******	☑		AU Db Admin	Reset	N
160	Irshad	AU	010	*******	☑	8/15/2017	AU Db User	Reset	N
10	Khaira	AU	012	*****	☑	4/1/2015	AU Db User	Reset	N
177	Mehra	E&E	121	*****	☑		AU Db Admin	Reset	N
128	Mohan	M3		*****	☐		Material Inventory Users	Reset	DC
129	Mohan	M3	011	*****	☐		Unassigned	Reset	DC
130	Mohd	E&E		*****	☐		Unassigned	Reset	DC
176	Mohd	AU	019	******	☑	11/15/2019	AU Db User	Reset	N
131	Mohd	M3		*****	☐		Material Inventory Users	Reset	DC
7	Muhan	AU	011	*******	☑	2/15/2016	AU Db User	Reset	N
132	Muhan	M3		*****	☐		Material Inventory Users	Reset	DC
133	Muhan	E&E		*****	☐		Unassigned	Reset	DC
142	Muhan	M3		*****	☐		Material Inventory Users	Reset	DC
6	Noor F	AU	013	***********	☑		AU Db User	Reset	N
136	Noor H	E&E		*****	☐		Unassigned	Reset	DC
137	Norhu	CEE	011	*****	☑		Chemical DB User	Reset	N

Figure 49 - A part of tables from MS Access Database

To enable table sharing for a database from MySQL to MS Access, I need to connect MS Access to MySQL. By doing this, we can utilize MySQL queries directly in the MS Access environment without having to create a new web application to access the MySQL database.



Employee Privileges
Employees
Employees1
Filters
Inventory
Inventory Transactions
Name AutoCorrect Save Failures
Order Details
Order Details Status
Orders
Orders Status
Suppliers
tblAccountCategory
tblAccountDetails
tblAccountTechnical
tblApend
tblApend20
tblBackchargeBilling
tblBillingDetail
tblBillingStudent_Supervisor
tblChemicalStoringSubgroup
tblChemicalType
tblChemStoringGroup
tblChemStoringGroup_StoID
tblClassifiedList
tblConsOrder
tblConsTransactions
tblConsTxTypes

☐ Save password

WELCOME TO THE AU DATABASE SYSTEM 4.0

OCCUPATIONAL SAFETY AND HEALTH ACT 1994 [ACT 514]
USECHH REGULATIONS 2000

PART II
IDENTIFICATION OF CHEMICALS HAZARDOUS TO HEALTH

ter of chemicals hazardous to health.

mployer shall identify and record in a register all chemicals hazardous to health used in the place of

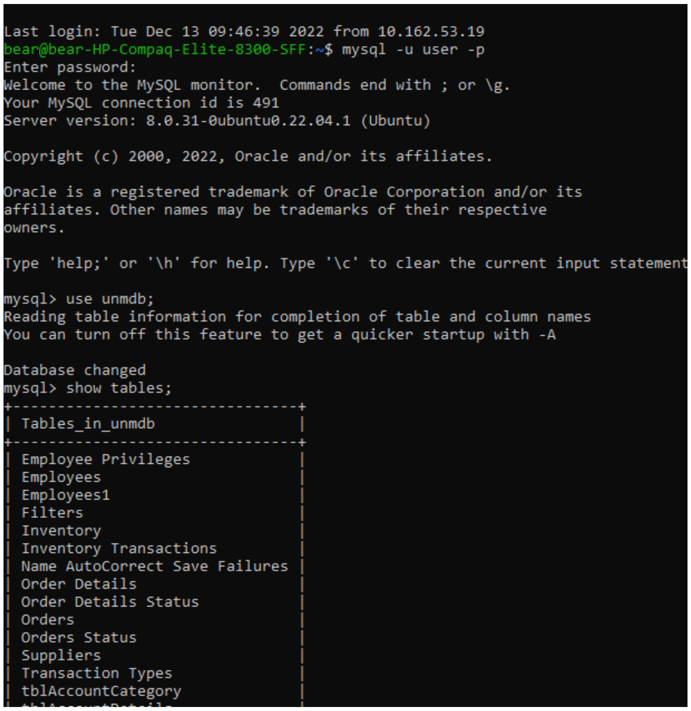
register shall be maintained in good order and condition and be updated from time to time and shall the following information:

(a) a list of all chemicals hazardous to health used;
(b) the current Chemical Safety Data Sheet for each of the chemicals hazardous to health except for pesticides which shall have information as specified in Schedule III;
(c) the average quantity used, produced or stored per month or per year whichever is applicable for each of the chemicals hazardous to health;
(d) the process and work area where the chemicals hazardous to health are used; and
(e) the name and address of the supplier of each of the chemicals hazardous to health.

(3) The register shall be accessible to all employees at the place of work who may be exposed or are likely to be exposed to chemicals hazardous to health.

Figure 50 - Connecting MS Access to the MySQL server

After being successful on localhost, it is now time to launch MySQL on the test server from the remote desktop and establish a connection to it.



```
Last login: Tue Dec 13 09:46:39 2022 from 10.162.53.19
bear@bear-HP-Compaq-Elite-8300-SFF:~$ mysql -u user -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 491
Server version: 8.0.31-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement

mysql> use unmdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------------------------+
| Tables_in_unmdb                 |
+---------------------------------+
| Employee Privileges             |
| Employees                       |
| Employees1                      |
| Filters                         |
| Inventory                       |
| Inventory Transactions          |
| Name AutoCorrect Save Failures  |
| Order Details                   |
| Order Details Status            |
| Orders                          |
| Orders Status                   |
| Suppliers                       |
| Transaction Types               |
| tblAccountCategory              |
```

Figure 51 - Running MySQL on the test server (Ubuntu version)

To make transportation simple, I find a solution for transferrin all the tables from MS Access to MySQL with all attributes features (around 80 table exit is MS Access and it was not rational to transfer them one by one, also with direct export from MS Access to MySQL, PK. FK and AI will not apply for attributes), and also I have to test the functionality of the program with tables on the MySQL server and local MS Access queries.

Finally, all tables are now present on the MySQL server, but it is critical to verify all tables, their primary keys, and foreign keys (because of the transportation, some of the tables transformed without data and I have to fix them).

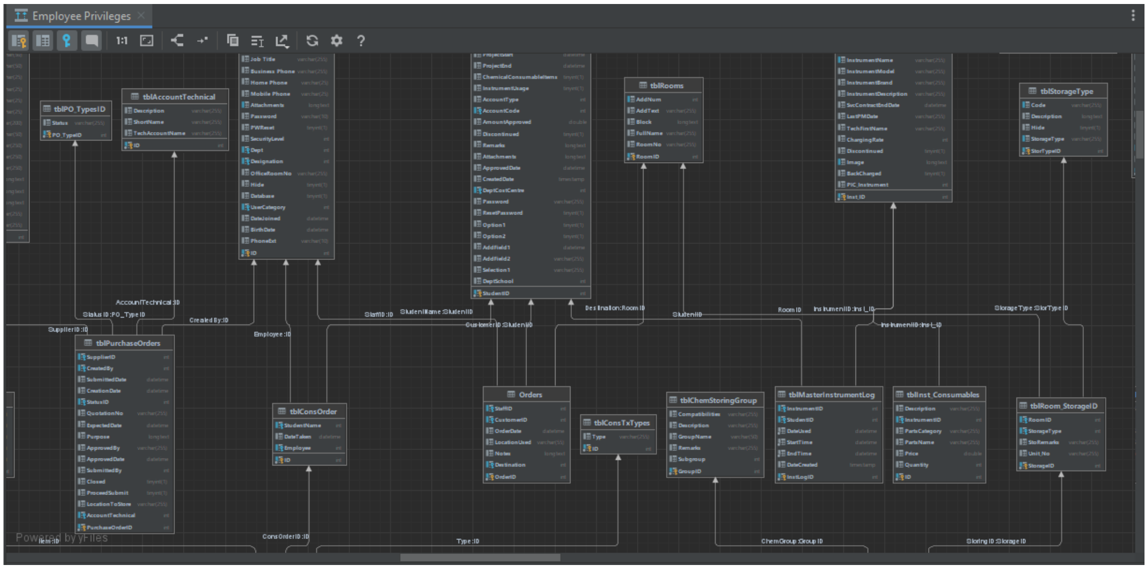
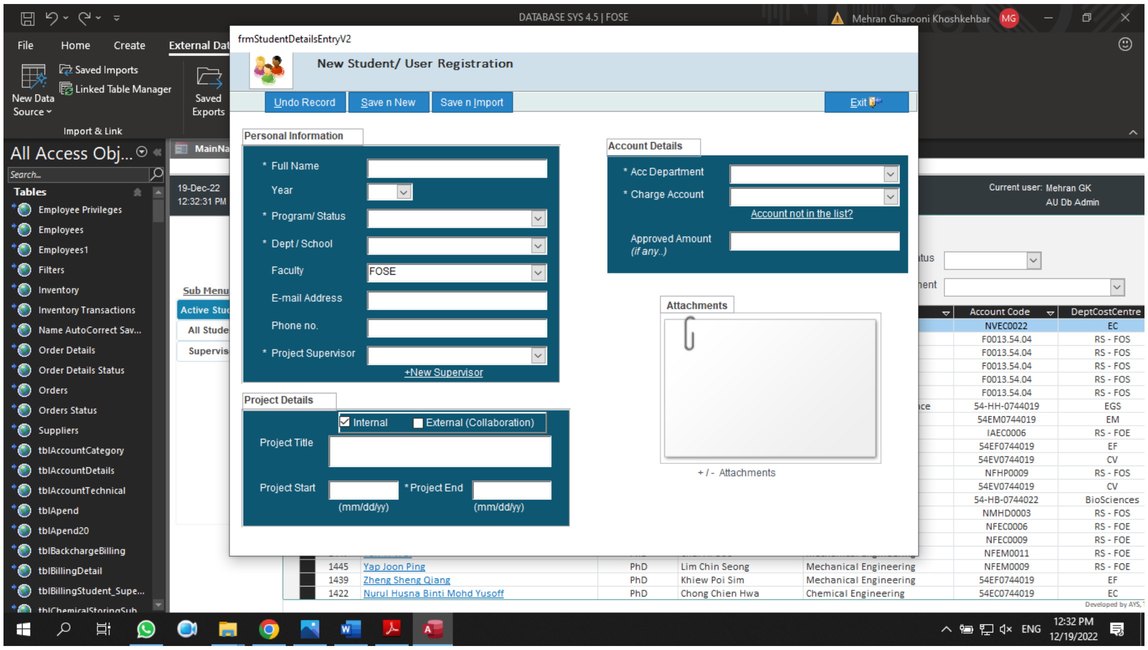


Figure 52 - Creating relation between tables on the MySQL

Connect MS Access to the MySQL server's tables, and all forms and queries in Access begin to function properly (for example for adding a new student, we can use a form in MS Access and the data will add to the whole database)



					Account Code	DeptCostCentre
					NVEC0022	EC
					F0013.54.04	RS - FOS
					F0013.54.04	RS - FOS
					F0013.54.04	RS - FOS
					F0013.54.04	RS - FOS
					F0013.54.04	RS - FOS
					54-HH-0744019	EGS
					54EM0744019	EM
					IAEC0006	RS - FOE
					54EF0744019	EF
					54EV0744019	CV
					NFHP0009	RS - FOS
					54EV0744019	CV
					54-HB-0744022	BioSciences
					NMHD0003	RS - FOS
					NFEC0006	RS - FOE
					NFEC0009	RS - FOE
					NFEM0011	RS - FOE
1445	Yap Joon Ping	PhD	Lim Chin Seong	Mechanical Engineering	NFEM0009	RS - FOE
1439	Zheng Sheng Qiang	PhD	Khiew Poi Sim	Mechanical Engineering	54EF0744019	EF
1422	Nurul Husna Binti Mohd Yusoff	PhD	Chong Chien Hwa	Chemical Engineering	54EC0744019	EC

Figure 53 - Changing the MySQL Database from the MS Access environment

In the end, the final version of the SQL file that contains all the tables (81 tables) is finished and ready to use for local or web applications.

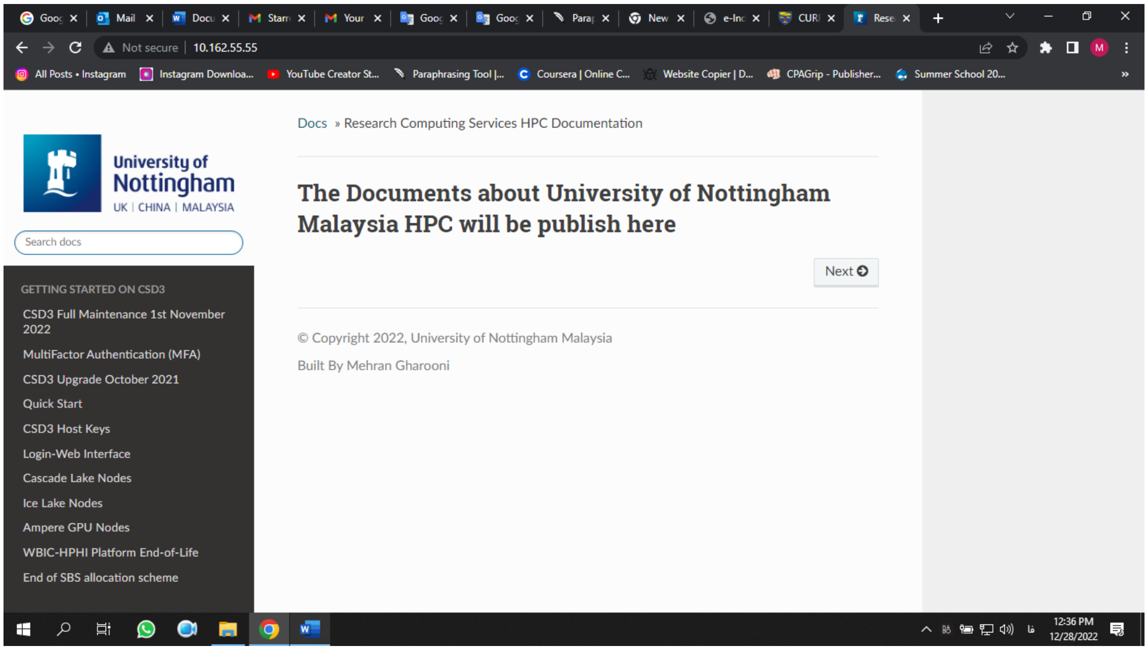
2.18 HPC Portal:

- Devices or systems used for performing the tasks: Web / Windows 10.
- Given timeframe to complete the tasks: One week, from week 23 until week 25.
- Problems faced during the execution of the tasks: Building a website without knowledge of HTML and CSS. It needs very particular HPC information.
- How problems were solved: Using an application for converting python files to the website called Sphinx, and for creating information about UNM HPC, make a sample of other University HPC.
- Relation of this task to passed courses at the university: Because running codes in Python is so much related to the Functional and Logic programing course

(WID3001), and because of writing information about HPC, it was related to the Information System course (WIA1001), and also Knowledge Representation and Reasoning (WID2001).

The final task defined for me which is developing a webpage that describes all the characteristics and information about the university of Nottingham HPC.

First I built a template for the HPC portal website similar to Cambridge University’s HPC portal, (the template got ready we need only fill in the UNM HPC details)



Docs » Research Computing Services HPC Documentation

The Documents about University of Nottingham Malaysia HPC will be publish here

© Copyright 2022, University of Nottingham Malaysia

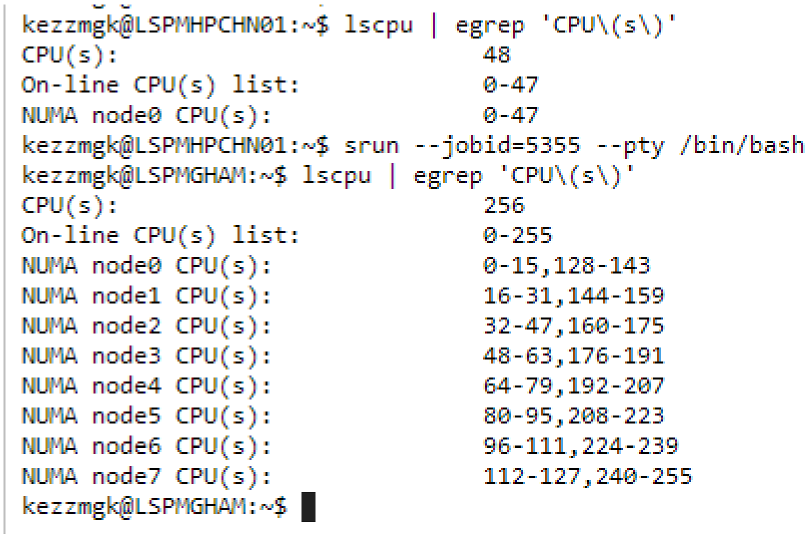
Built By Mehran Gharooni

Figure 54 - The prototype of the University of Nottingham HPC Portal

I added the constructed website (template) as the primary URL on the test server so that anyone who enters the server's IP address as a URL can view the entire website.

In order to create more accurate material, I conducted some research on HPC at other universities, like Cambridge, Oxford, MIT, and others, and compared their HPC with the University of Nottingham's HPC.

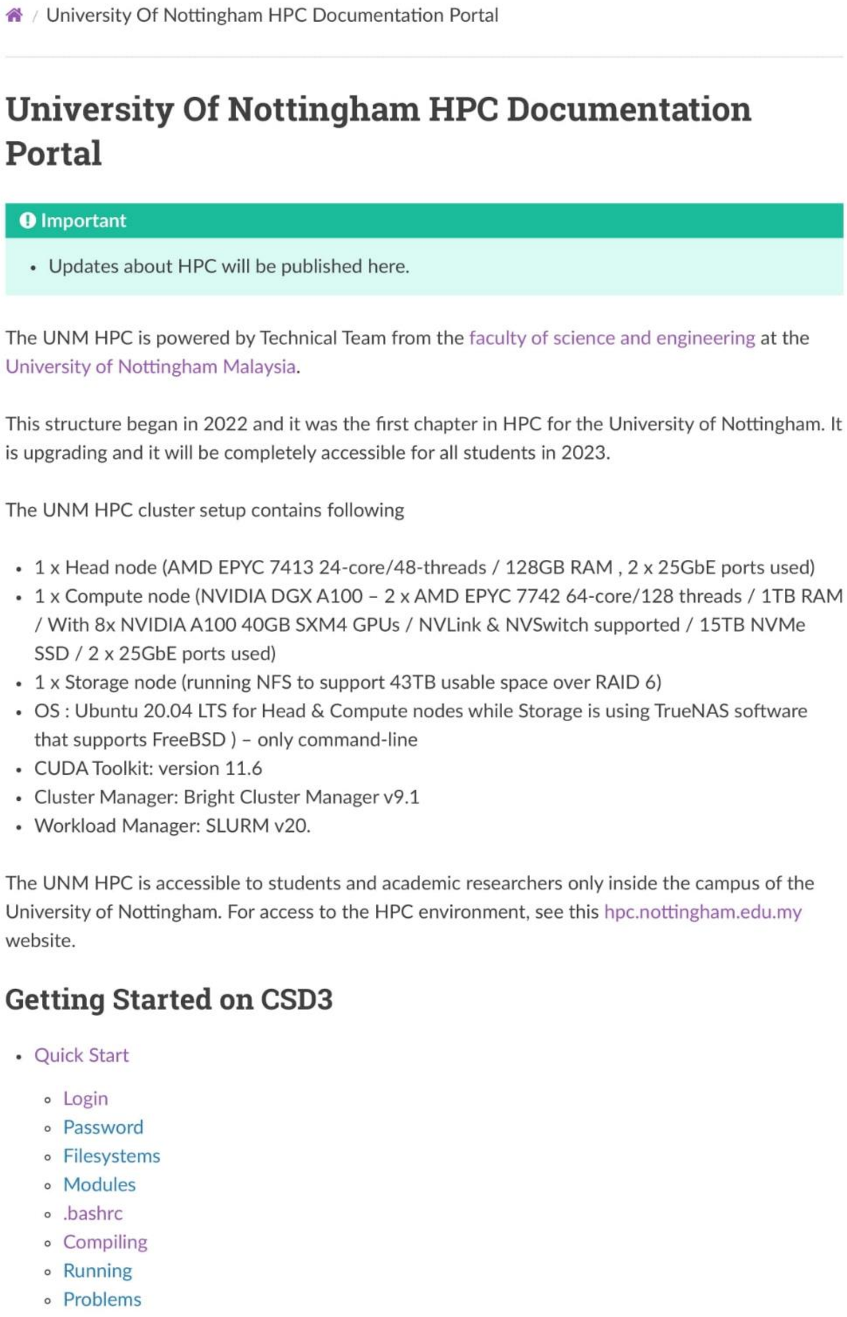
and Extracted the threads, cores, and information about the HPC CPU. (There are 304 CPU cores at the University of Nottingham HPC, of which 256 are used on the compute node and 48 are used in the head node.)



```
kezzmgk@LSPMHPCHN01:~$ lscpu | egrep 'CPU\(s\)'
CPU(s):                          48
On-line CPU(s) list:             0-47
NUMA node0 CPU(s):               0-47
kezzmgk@LSPMHPCHN01:~$ srun --jobid=5355 --pty /bin/bash
kezzmgk@LSPMGHAM:~$ lscpu | egrep 'CPU\(s\)'
CPU(s):                          256
On-line CPU(s) list:             0-255
NUMA node0 CPU(s):               0-15,128-143
NUMA node1 CPU(s):               16-31,144-159
NUMA node2 CPU(s):               32-47,160-175
NUMA node3 CPU(s):               48-63,176-191
NUMA node4 CPU(s):               64-79,192-207
NUMA node5 CPU(s):               80-95,208-223
NUMA node6 CPU(s):               96-111,224-239
NUMA node7 CPU(s):               112-127,240-255
kezzmgk@LSPMGHAM:~$ 
```

Figure 55 - Extracting the information of HPC

I created the University of Nottingham Malaysia HPC's first page and uploaded it to the test server (I explained about HPC, its details, and hardware features).



University Of Nottingham HPC Documentation Portal

Important

- Updates about HPC will be published here.

The UNM HPC is powered by Technical Team from the faculty of science and engineering at the University of Nottingham Malaysia.

This structure began in 2022 and it was the first chapter in HPC for the University of Nottingham. It is upgrading and it will be completely accessible for all students in 2023.

The UNM HPC cluster setup contains following

- 1 x Head node (AMD EPYC 7413 24-core/48-threads / 128GB RAM , 2 x 25GbE ports used)
- 1 x Compute node (NVIDIA DGX A100 – 2 x AMD EPYC 7742 64-core/128 threads / 1TB RAM / With 8x NVIDIA A100 40GB SXM4 GPUs / NVLink & NVSwitch supported / 15TB NVMe SSD / 2 x 25GbE ports used)
- 1 x Storage node (running NFS to support 43TB usable space over RAID 6)
- OS : Ubuntu 20.04 LTS for Head & Compute nodes while Storage is using TrueNAS software that supports FreeBSD) – only command-line
- CUDA Toolkit: version 11.6
- Cluster Manager: Bright Cluster Manager v9.1
- Workload Manager: SLURM v20.

The UNM HPC is accessible to students and academic researchers only inside the campus of the University of Nottingham. For access to the HPC environment, see this hpc.nottingham.edu.my website.

Getting Started on CSD3

- Quick Start
  - Login
  - Password
  - Filesystems
  - Modules
  - .bashrc
  - Compiling
  - Running
  - Problems

Figure 56 - The first page of UNM HPC Portal

I build these pages and websites by a free and open-source package called SPHINX; It is using python and a terminal for converting the specific format to an HTML website.

Since this project is ongoing and my internship is coming to an end, it is very likely that the University of Nottingham and I will continue to work together for finishing this task.

3 Skills Gained, Initiative Taken, Guidance from Organization

3.1 Skills Gained:

I gained a variety of skills during this internship course, the majority of which will be crucial and highly valuable for my future career. I'm delighted that I was able to complete the course successfully and gain useful knowledge and I can state with certainty that this internship course was one of the most significant courses I've ever attended.

Working in the Linux environment was one of the most crucial skills I picked up. I believe this operating system will be very helpful for me in the future because it is both powerful and practical.

Building a server, connecting to it, and exchanging information through it are all important skills that were learned in this course and will be very helpful for projects like websites, mobile and computer applications, or any other type of system.

One of the skills I acquired through this internship program was the ability to install software, troubleshoot it, connect it to the server, change it, and interact with its databases.

However, I'm confident in saying that setting up and using HPC was the most crucial skill I learned in this course. My supervisor was very helpful in teaching me all the crucial components of HPC systems, and I even managed to launch a small-scale HPC using a SLURM system.

3.2 Initiative Taken:

During this 6-month period, I took the initiative to complete some tasks that I can be proud of.

I went into detail about them in the work experience section, but in this section, I'd like to talk more generally about the new approaches I've taken to problem-solving.

According to my knowledge of the HPC system and after analyzing it, I was able to run the Paraview server on the HPC node computer and even users could connect to it.

Also, for running MATLAB jobs on the HPC cluster, after processing my observations from the software and analyzing them, I was able to notice the problem and its solution, and share it with the MATLAB manager.

Finding HPC security issues and bringing them to the attention of my supervisor allowed us to resolve them, which was possibly the most useful initiative I took during this internship.

3.3 Guidance from Organization:

Fortunately, despite the fact that it was an international organization, I was able to complete my internship with a top-notch and professional company.

It is obvious that a large part of what I learned throughout this course was a result of the advice I received from my supervisor. My supervisor showed me the road, and I followed it in order to fulfill the organization's goals.

I was first requested to study Linux by Mr. Ellias Ismail, who also provided me the chance to test these lectures using HPC. My supervisor helped me to understand how to work with global update systems like MySQL.

In conclusion, I would like to say that Nottingham University's direction and support allowed me to develop my profession and gain useful training, as well as advance my career.

4 Conclusion and Suggestion

4.1 Conclusion:

I spend my internship on the technical team for 25 weeks throughout my time at university; it was a fantastic chance for me to advance my abilities. I gained a great deal of organizing skills in addition to technical ones. such as how to communicate with the project manager, work as a team member, and assign duties to team members, and so on.

I advanced because of the special technical skills I obtained in this training. In this internship course, I even learned and used a few of the concepts I struggled with in university (like databases). I got able to turn my potential knowledge into reality and that is very important and valuable.

4.2 Suggestion:

In relation to this internship, I would like to suggest that this course could be held in the final semester so that students can continue their job after the internship and university. For instance, the University of Nottingham would like to collaborate with me, but this is essentially impossible given the requirements of the institution and my visa.

5 References

[1] Reuther, Albert, Jeremy Kepner, William Arcand, David Bestor, Bill Bergeron, Chansup Byun, Matthew Hubbell, et al. "Llsupercloud: Sharing HPC systems for diverse rapid prototyping." In 2013 IEEE High-Performance Extreme Computing Conference (HPEC), pp. 1-6. IEEE.

[2] SETARA. 2009. “The 2009 Rating of Malaysian universities and university colleges.” Published in 2009. https://web.archive.org/web/20150404011133/http:/www.mqa.gov.my/SETARA09/pdf/result_en.pdf

[3] The University of Nottingham. N.d. “About the University, Malaysia Campus History.” Accessed December 25, 2022. https://www.nottingham.edu.my/AboutUs/History/MalaysiaHistory.aspx

[4] University of Nottingham 2022. “About the University, Strategic Plan 2022-26, Vision.” Accessed 2022. https://www.nottingham.edu.my/AboutUs/Strategy/index.aspx

[5] The University of Nottingham. n.d. “About the University, From the Interim Provost and CEO.” Accessed December 2022. https://www.nottingham.edu.my/AboutUs/index.aspx

[6] The University of Nottingham. n.d. “University structure.” Accessed December 2022. https://www.nottingham.ac.uk/about/structure/universitystructure.aspx

[7] Van Der Spoel D, Lindahl E, Hess B, Groenhof G, Mark AE, Berendsen HJ (2005). "GROMACS: fast, flexible, and free". J Comput Chem. 26 (16): 1701–18. doi:10.1002/jcc.20291.

[8] Slurm Schedmd. n.d. “Quick Start User Guide, Overview.” Accessed August 2022. https://slurm.schedmd.com/quickstart.html

6 Appendix A – UNMC’S HPC PERFORMANCE TEST

University of Nottingham

UK | CHINA | MALAYSIA

UNMC’S HPC PERFORMANCE TEST

Mr. Ellias Ismail

Mehran Gharooni

Abstract:

The effectiveness of any freshly introduced system is a crucial component. Various test types should be run on a system to determine if it can fulfill the expected demands. The University of Nottingham Malaysia is recently developing a new system that students, professors, and even university staff can benefit from High-Performance Computing (HPC) to meet their needs to access a supercomputer. Therefore, several tests have been conducted on this system, and one of the tests that have been done on this HPC system is to check the difference in processing performance on CPU and GPU so that we can show the necessity of such a system. Based on the experiment results we were able to get with the aid of Python, the performance of the system is examined and discussed in this report based on the rate at which operations are processed.

Introduction:

The launched HPC system has several CPUs and graphics cards that are designed to run jobs powerful and at high speed, in this report, the real performance of the processors of this system is challenged, and the differences and the proper placement of CPU and GPU have been tested and significant results achieved.

An identical function that measures the CPU and GPU under the same circumstances is necessary to test these two components.

The method used in this experiment is a 100 million circulation loop whose function is to fill an array with the same value. This method runs twice, once with CPU components and again using GPU, then the time for these components for executing this method gets measured. So the performance speed of these two components can be calculated with very high accuracy.

This program has been run continuously 10 times on the HPC system and the result has been recorded, and it has also been run on the local computer of the university so that we can compare the power and speed of the HPC system with a normal computer.

Code explanation:

Packages:

A brief summary of each of the 3 packages utilized in this test is provided below:

Numpy: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.[1]

In this experiment, Numpy is used to generate an array that has 100 million positions, and each position is filled with the number 1. In fact, we created an "all-ones matrix" with 100 million columns and 1 row.

Numba: Numba is an open-source "JIT compiler" that translates a subset of Python and NumPy into fast machine code using LLVM, via the " LLVMlite" Python package. It offers a range of options for parallelizing Python code for CPUs and GPUs, often with only minor code changes.[2]

In this test, Namba is used to run the program on the Nvidia GPU, using the CUDA command, the program is run on the Nvidia graphics card instead of the CPU.

Timeit: Timeit is a package that shows the time with high accuracy.

In this test, Timeit is used to get the exact time of execution of the methods by the components which need to measure

Functions:

In this test, two completely identical functions (Func and Func2) have been used, with the difference that "Func2" is executed on the graphic card with the assistance of "Namba" instead of the CPU, below is how these functions have been working.

```
def func(a):
   for i in range(100000000):
      a[i]+= 1
```

As it is obvious, this function receives an array with 100 million positions "func(a)", and adds 1 to each position at each turn.

This is just so that some action happens while looping through this loop.

```
@jit(target_backend='cuda')
def func2(a):
   for i in range(100000000):
      a[i]+= 1
```

The second function is completely similar to the first function, except that the line was written before that, this is the code that can be performed in GPU by the "Numba" package.

Main Codes:

In the main part of the code, we first create an array using "Numpy". This array has 100 million positions, each of which is filled with the number one.

```
n = 100000000
a = np.ones(n, dtype = np.float64)
```

*A data type object (an instance of numpy.dtype class) describes how the bytes in the fixed-size block of memory corresponding to an array item should be interpreted.

Then we create a variable and keep the time in it, after that we execute the "func (with CPU)" and finally we subtract the time after the "func" execution from the variable and get the duration of the program execution by CPU.

The array created in the previous step is also given to "func" so that the function is executed correctly.

```
start = timer()

func(a)

print("without GPU:", timer()-start)
```

And in the last part, the steps of the previous part are done again, but this time with funk 2, we return the previously made array to its original state so that the conditions are completely the same and equal,

```
a = np.ones(n, dtype = np.float64)

start = timer()

func2(a)

print("with GPU:", timer()-start)
```

You can see the completed code in the following of this report.

This program is run consecutively 10 times to produce accurate and reliable results. And it has been implemented both in the HPC system and in the local system of the university so that the power and ability of the HPC system can be understood on a better scale.

Results review and explanation:

After the code used got complete and efficient, interesting and significant results were obtained. Here you can clearly understand the power of the GPU used in the HPC system.

The following table shows the obtained results:

	CPU	GPU
1	29.77	0.09
2	33.45	0.04
3	31.57	0.04
4	31.68	0.03
5	30.87	0.03
6	28.17	0.04
7	25.70	0.03
8	26.71	0.03
9	28.17	0.04
10	28.63	0.03

100 million lopping in HPC

As you can see, the GPU is very powerful and fast. To the extent that it can operate up to 1000 times faster than the CPU in only 100 million revolutions.

This means that it takes only 1 day of GPU work to do something that would take 3 years on a CPU.

It can be seen in this table that the time taken by the GPU for the first round is almost twice as much as the other rounds. The reason for this is that the graphics card needs some time to start working. Here, the time to start the process is almost twice as long as the normal process

In the table below, the comparison of this test in the HPC system and the normal computer is displayed, and from this comparison, significant results can be obtained.

	HPC		Normal Computer	
	CPU	GPU	CPU	GPU
1	29.77	0.09	37.82	22.71
2	33.45	0.04	33.00	0.22
3	31.57	0.04	33.09	0.28
4	31.68	0.03	32.52	0.17
5	30.87	0.03	33.21	0.16
6	28.17	0.04	33.40	0.19
7	25.70	0.03	32.24	0.16
8	26.71	0.03	32.53	0.17
9	28.17	0.04	32.46	0.16
10	28.63	0.03	31.73	0.16

Comparison between HPC and normal computer in 100 million lopping

In normal mode, the graphics card of the HPC system is about 5 times faster than a normal computer, on the other hand, this has not caused the HPC system to ignore the CPU, as it is known that the CPU is approximately HPC system is about 10-15% faster.

But the most important thing to note in this table is the time difference between the first round and the rest of the rounds. In the previous table, we checked this point for the HPC system that the first round took about 2 times more time than the rest of the rounds. But in a normal computer, this takes up to 100 times longer.

As a result, it's easy to observe how much faster and powerful the HPC system is and how much time it may save students. Furthermore, the files will always be kept in the HPC area, which has a very high level of security.

CPU Benefits:

The rationale for including the CPU in this system at this time could be questioned.

To prove the necessity of CPU in HPC, another experiment was designed. In this new test, only 50 rotations were considered for each function instead of 100 million rotations.

```
def func(a):
   for i in range(50):
      a[i]+= 1
```

```
@jit(target_backend='cuda')
def func2(a):
   for i in range(50):
      a[i]+= 1
```

When the task at hand is not too demanding or complicated, CPU can outperform GPU.

The table below makes this very apparent.

	CPU	GPU
1	0.00	0.03
2	0.00	0.01
3	0.00	0.00
4	0.00	0.00
5	0.00	0.01
6	0.00	0.00
7	0.00	0.01
8	0.00	0.01
9	0.00	0.00
10	0.00	0.02

50 lopping in HPC

This test revealed that the CPU can perform simpler tasks more quickly than the graphics card. And fortunately, both components are advantageous to the HPC system.

Conclusion:

In this paper, the speed and performance of the HPC system's components were examined, and their differences were analyzed for simple and short tasks and for long and complex tasks.

On the HPC system, a number of experiments were run in order to demonstrate the capabilities of the CPU and GPU, and considerable results were attained.

According to the conducted tests, it can be concluded that the GPU of the HPC system has a very high speed for processing long tasks, to the extent that it can work up to 1000 times faster than the CPU.

This feature has the potential to save students a lot of time and encourage them to use the system.

On the other hand, considering that most of the tasks given to the system by students are one-step tasks, so preparing the graphics card to do the work is very important. Because, as discussed in this report, in local computers, the processing time of the first task in the GPU can be up to 100 times slower than normal, while this ratio is only 2 times in the HPC system.

Finally, the necessity of using a CPU to complete simpler tasks was investigated. The results showed that, for the majority of tasks that don't require sophisticated operations, the CPU can function considerably more quickly and that the GPU is not necessary for these circumstances.

Reference:

[1]: Charles R Harris; K. Jarrod Millman; Stéfan J. van der Walt; et al. (16 September 2020). "Array programming with NumPy" (PDF). Nature. 585 (7825): 357–362. doi:10.1038/S41586-020-2649-2. ISSN 1476-4687. PMC 7759461. PMID 32939066. Wikidata Q99413970.

[2]: "Releases · numba/numba". GitHub. Retrieved 2022-09-19.

7 Appendix B - MathLab on HPC Report (Monte Carlo Simulation)

University of Nottingham

UK | CHINA | MALAYSIA

MathLab on HPC Report

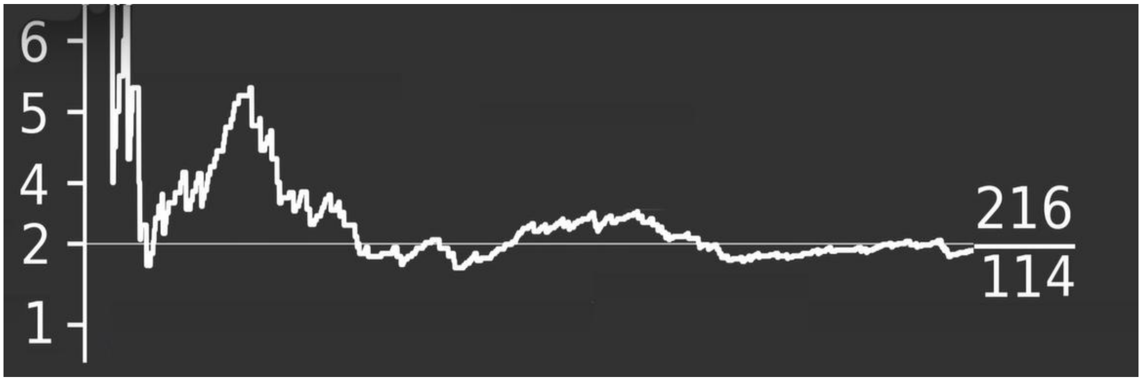
(Monte Carlo Simulation)

Supervisor: Mr. Ellias Ismail

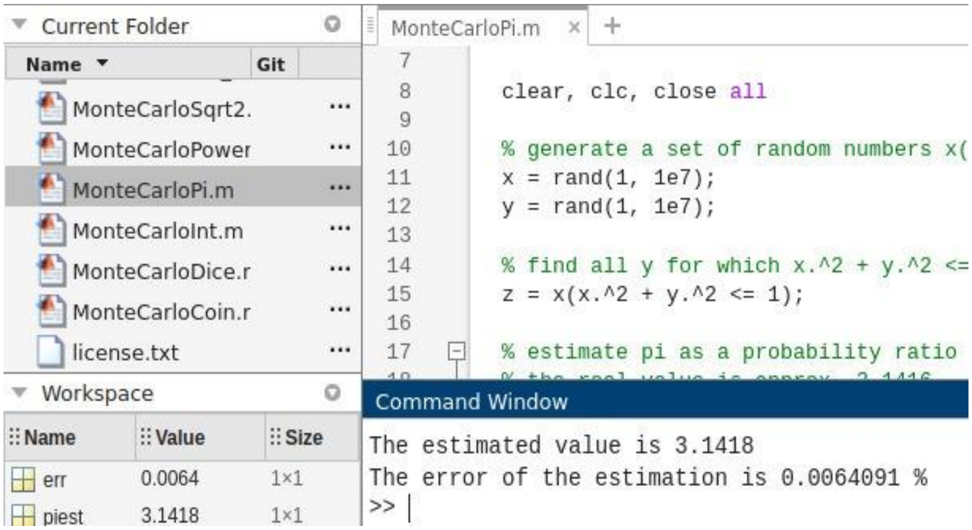
Mehran Gharooni

In this report, I will explain my endeavors in running Monte Carlo Simulation on the HPC system, and local system and find the differences between them.

First I had some research about Monte Carlo Simulation and I understood that Monte Carlo is a method that is based on random happens on the large scale, instead of considering all possible situations. To illustrate, on a surface, we have two squares and one of them has twice the area, and randomly we select a place on our surface and put a sign in that place; after a while, if we count the number of signs, we can see the number of signs in the bigger square is almost twice of the smaller square.



So I run the Monte Carlo code (From MathLab Website) on the local MathLab, and the following picture is one of the results:



```
clear, clc, close all

% generate a set of random numbers x(
x = rand(1, 1e7);
y = rand(1, 1e7);

% find all y for which x.^2 + y.^2 <=
z = x(x.^2 + y.^2 <= 1);

% estimate pi as a probability ratio
% the real value is approx. 3.1416
```

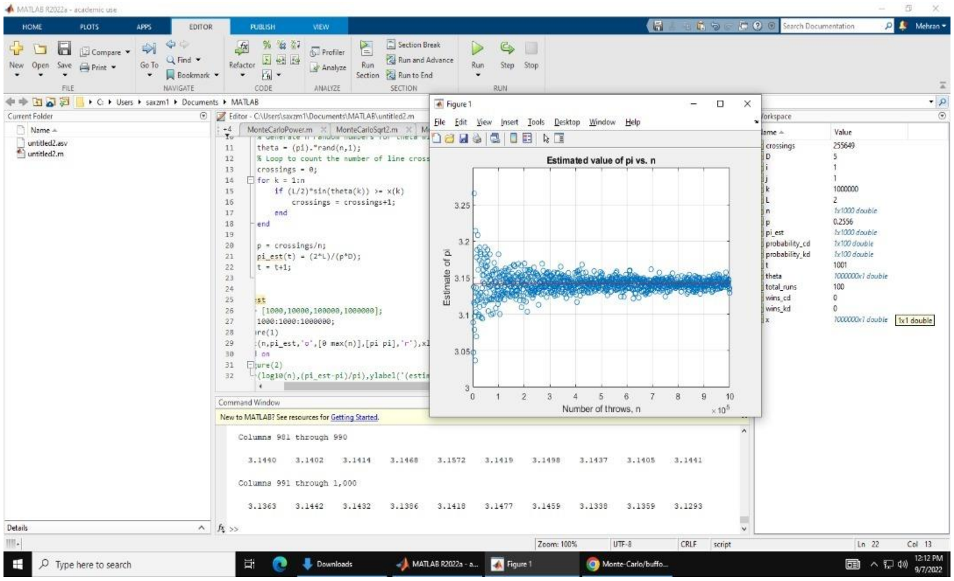
Name	Value	Size
err	0.0064	1×1
piest	3.1418	1×1

```
The estimated value is 3.1418
The error of the estimation is 0.0064091 %
>> |
```

From this picture, we can see x and y were huge numbers (10 Million), and by using Monte Carlo, the estimated Pi got 3.1418 which is very near to the exact Pi.

But I wanted to show some Simulators, so I use another code that shows us the procedure of Monte Carlo by simulating; Also I wrote the code that I used at the end of this report.

This was the output on the local MathLab:



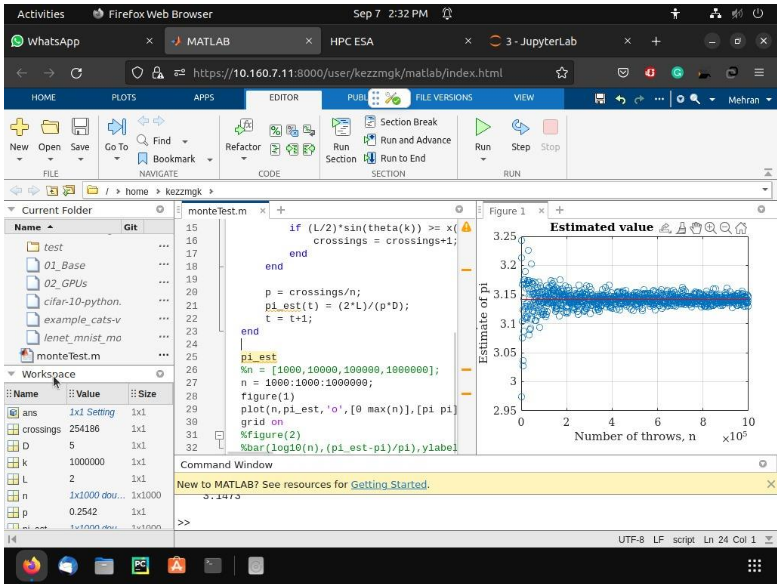
```
% Generate n random numbers for theta wi
theta = (pi).*rand(n,1);
% Loop to count the number of line cross
crossings = 0;
for k = 1:n
    if (L/2)*sin(theta(k)) >= x(k)
        crossings = crossings+1;
    end
end

p = crossings/n;
pi_est(t) = (2*L)/(p*D);
t = t+1;


st
 [1000,10000,100000,1000000];
1000:1000:1000000;
re(1)
(n,pi_est,'o',[0 max(n)],[pi pi],'r'),xl
 on
ure(2)
(log10(n),(pi_est-pi)/pi),ylabel('(esti
```

```
  Columns 981 through 990

    3.1440    3.1402    3.1414    3.1468    3.1572    3.1419    3.1498    3.1437    3.1405    3.1441

  Columns 991 through 1,000

    3.1363    3.1442    3.1432    3.1386    3.1418    3.1477    3.1459    3.1338    3.1359    3.1293

>>
```

Name	Value
crossings	255649
D	5
i	1
j	1
k	1000000
L	2
n	1x1000 double
p	0.2556
pi_est	1x1000 double
probability_cd	1x100 double
probability_kd	1x100 double
t	1001
theta	1000000x1 double
total_runs	100
wins_cd	0
wins_kd	0
x	1000000x1 double 1x1 double

As you can see here, As the number of random choices increases, we get closer to the correct answer.

After this, I tried this code on the HPC MathLab:



```
            if (L/2)*sin(theta(k)) >= x(
                crossings = crossings+1;
            end
        end

        p = crossings/n;
        pi_est(t) = (2*L)/(p*D);
        t = t+1;
    end

    pi_est
    %n = [1000,10000,100000,1000000];
    n = 1000:1000:1000000;
    figure(1)
    plot(n,pi_est,'o',[0 max(n)],[pi pi]
    grid on
    %figure(2)
    %bar(log10(n),(pi_est-pi)/pi),ylabel
```

Name	Value	Size
ans	1x1 Setting	1x1
crossings	254186	1x1
D	5	1x1
k	1000000	1x1
L	2	1x1
n	1x1000 dou…	1x1000
p	0.2542	1x1

New to MATLAB? See resources for Getting Started.

```
    3.1473

>>
```

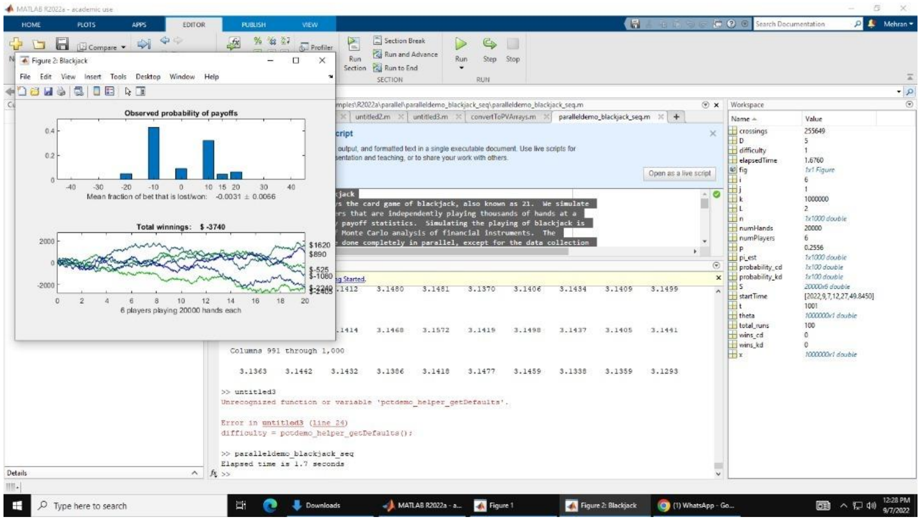
As you can see, everything is the same, except the “Figures pop up”, which means Figures should show on the same webpage.

Another difference between HPC MathLab and local Mathlab for the above example was “Speed”. HPC MathLab is much faster than local MathLab which is very pleasant.

Another test that had done by HPC MathLab was Sequential Blackjack (From MathLab Website).

For this case, we have to upload a zip file that contains some MathLab files, and Here I saw something strange about the HPC’s MathLab.

First I want to show you the output of this test on Local MathLab:



mples\R2022a\parallel\paralleldemo_blackjack_seq\paralleldemo_blackjack_seq.m

cript

output, and formatted text in a single executable document. Use live scripts for
sentation and teaching, or to share your work with others.

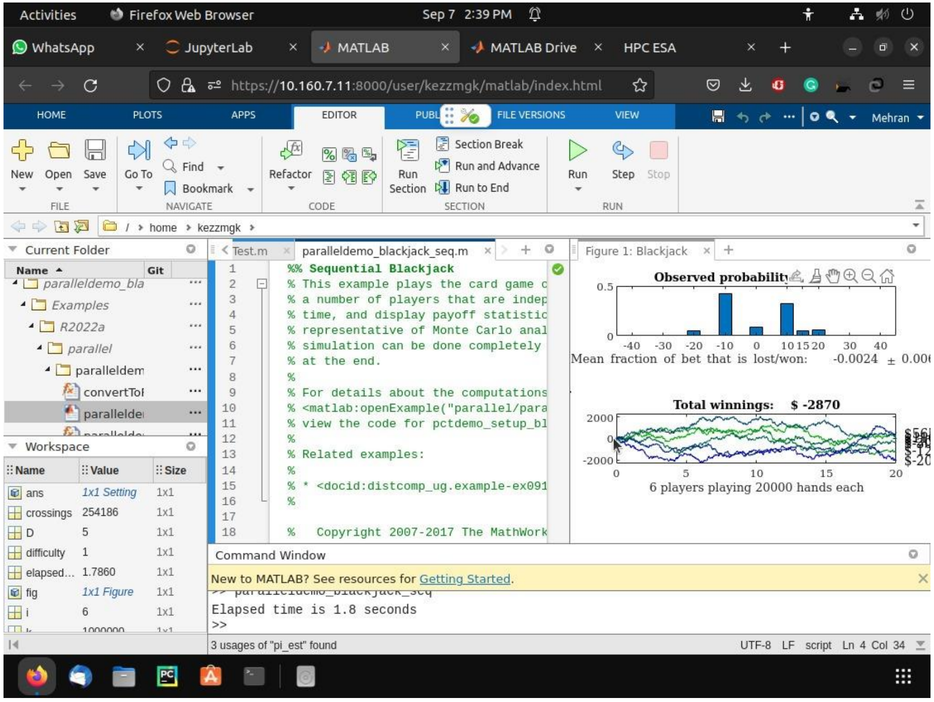
```
jack
s the card game of blackjack, also known as 21.  We simulate
rs that are independently playing thousands of hands at a
 payoff statistics.  Simulating the playing of blackjack is
 Monte Carlo analysis of financial instruments.  The
 done completely in parallel, except for the data collection
```

```
.1412    3.1480    3.1481    3.1370    3.1406    3.1434    3.1409    3.1499

.1414    3.1468    3.1572    3.1419    3.1498    3.1437    3.1405    3.1441

  Columns 991 through 1,000

    3.1363    3.1442    3.1432    3.1386    3.1418    3.1477    3.1459    3.1338    3.1359    3.1293

>> untitled3
Unrecognized function or variable 'pctdemo_helper_getDefaults'.

Error in untitled3 (line 24)
difficulty = pctdemo_helper_getDefaults();

>> paralleldemo_blackjack_seq
Elapsed time is 1.7 seconds
>>
```

Name	Value
crossings	255649
D	5
difficulty	1
elapsedTime	1.6760
fig	1x1 Figure
i	6
j	1
k	1000000
L	2
n	1x1000 double
numHands	20000
numPlayers	6
p	0.2556
pi_est	1x1000 double
probability_cd	1x100 double
probability_kd	1x100 double
S	20000x6 double
startTime	[2022,9,7,12,27,49.8450]
t	1001
theta	1000000x1 double
total_runs	100
wins_cd	0
wins_kd	0
x	1000000x1 double

As you can see, it is similar to the previous example.

And in the HPC output was totally fine:



```
%% Sequential Blackjack
% This example plays the card game c
% a number of players that are indep
% time, and display payoff statistic
% representative of Monte Carlo anal
% simulation can be done completely
% at the end.
%
% For details about the computations
% <matlab:openExample("parallel/para
% view the code for pctdemo_setup_bl
%
% Related examples:
%
% * <docid:distcomp_ug.example-ex091
%

%   Copyright 2007-2017 The MathWork
```

Workspace

Name	Value	Size
ans	1x1 Setting	1x1
crossings	254186	1x1
D	5	1x1
difficulty	1	1x1
elapsed...	1.7860	1x1
fig	1x1 Figure	1x1
i	6	1x1
k	1000000	1x1

New to MATLAB? See resources for Getting Started.

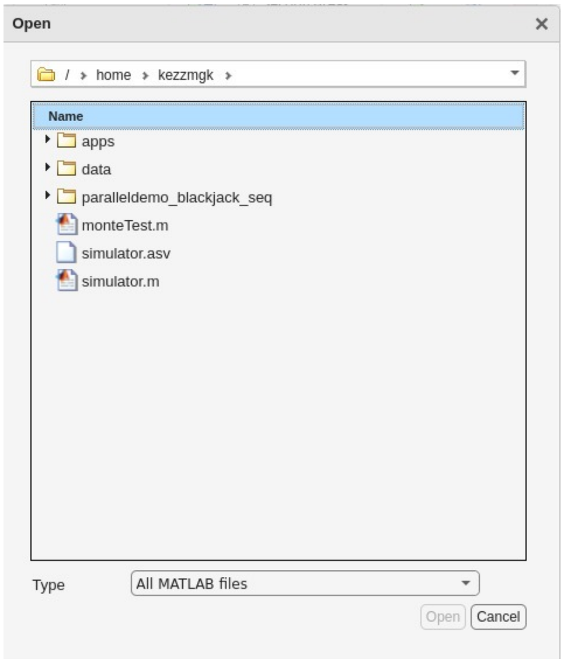
```
>> paralleldemo_blackjack_seq
Elapsed time is 1.8 seconds
>>
```

3 usages of "pi_est" found

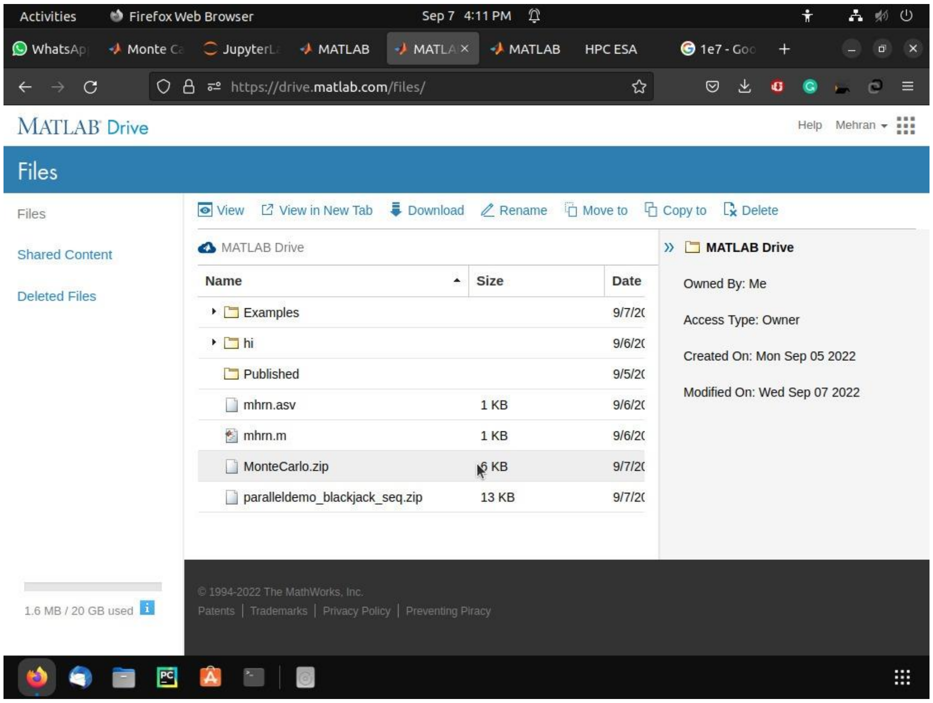
But here is some hardship for HPC’s MathLab.

HPC’s MathLab does not let us upload files from our PC or Laptop, in the HPC, the files that we can select should be on the Jupyter Lab!

It means, first we have to upload the file to Jupyter lab and after that, we can select that file on the MathLab.



And also if we want to use MathLab Drive, then uploaded files went to the MathLab client website, not HPC (“Mathlab has an online application that everyone can use, and it is so similar to HPC’s MathLab, the uploaded files from MathLab Drive are accessible from this application”).



Name	Size	Date
Examples		9/7/20
hi		9/6/20
Published		9/5/20
mhrn.asv	1 KB	9/6/20
mhrn.m	1 KB	9/6/20
MonteCarlo.zip	6 KB	9/7/20
paralleldemo_blackjack_seq.zip	13 KB	9/7/20

MATLAB Drive

Owned By: Me

Access Type: Owner

Created On: Mon Sep 05 2022

Modified On: Wed Sep 07 2022

© 1994-2022 The MathWorks, Inc.

Patents | Trademarks | Privacy Policy | Preventing Piracy

If I want to conclude this report in a few sentences, HPC's MathLab is working properly and fast. Showing plots, huge matrices, and other things are working smoothly on the MathLab; And there is no functional difference between HPC's MathLab and Local MathLab.

I hope this report was useful for you, Thanks

The Monte Carlo Simulator Code:

```
% This program generates one set of estimates of pi corresponding to

% 1000, 10000, 100000, 1000000 throws

L = 2; D = 5;

t = 1;

% No. of needle throws

for n = [1000:1000:1000000]

%for n = [1000,10000,100000,1000000]
```

```
    % Generate n random numbers for x within the range of 0 to D/2
    x = (D/2).*rand(n,1);
    % Generate n random numbers for theta within the range of 0 to pi
    theta = (pi).*rand(n,1);
    % Loop to count the number of line crossings
    crossings = 0;
    for k = 1:n
        if (L/2)*sin(theta(k)) >= x(k)
            crossings = crossings+1;
        end
    end

    p = crossings/n;
    pi_est(t) = (2*L)/(p*D);
    t = t+1;
end

pi_est
%n = [1000,10000,100000,1000000];
n = 1000:1000:1000000;
figure(1)
plot(n,pi_est,'o',[0 max(n)],[pi pi],'r'),xlabel('Number of throws, n'),ylabel('Estimate of pi'),title('Estimated value of pi vs. n')
grid on
```