Group 10

-Ishu Chaudhary

rest… did not do the work inspite of me continuously poking them to do.
Harshit and Harshavardhan did something different

Title: Exploring Cloud Computing Simulations with CloudSim

Introduction

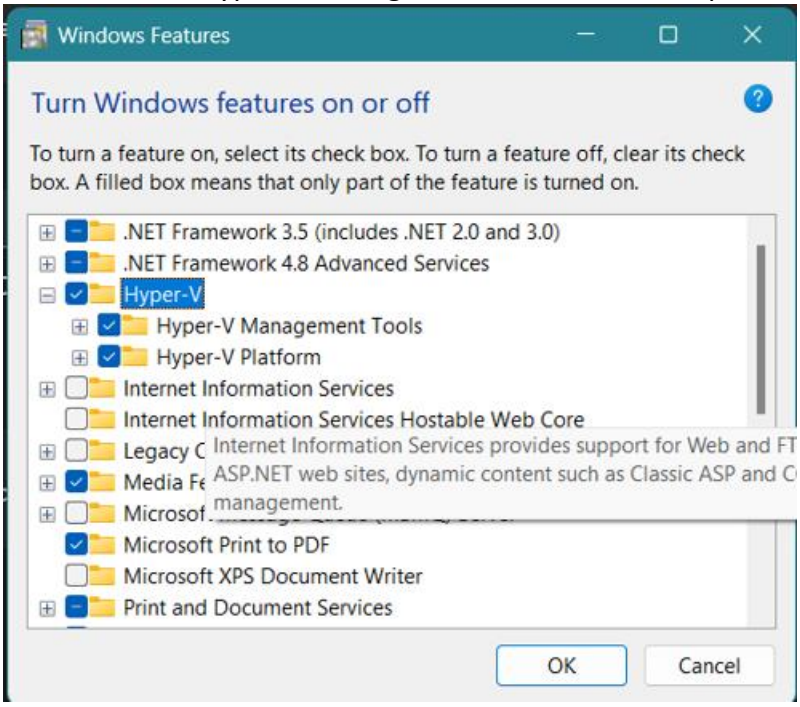
In this article, we'll delve into the world of cloud computing simulations using CloudSim. We'll walk through the process of creating a virtual environment on Windows with Hyper-V Manager, where we'll install Ubuntu 22.04. In this environment, we'll set up the Java Runtime Environment, configure environment variables, and work with CloudSim 3.0.3 to simulate cloud scenarios.

Creating the Virtual Environment

Unlocking Hyper-V Manager:

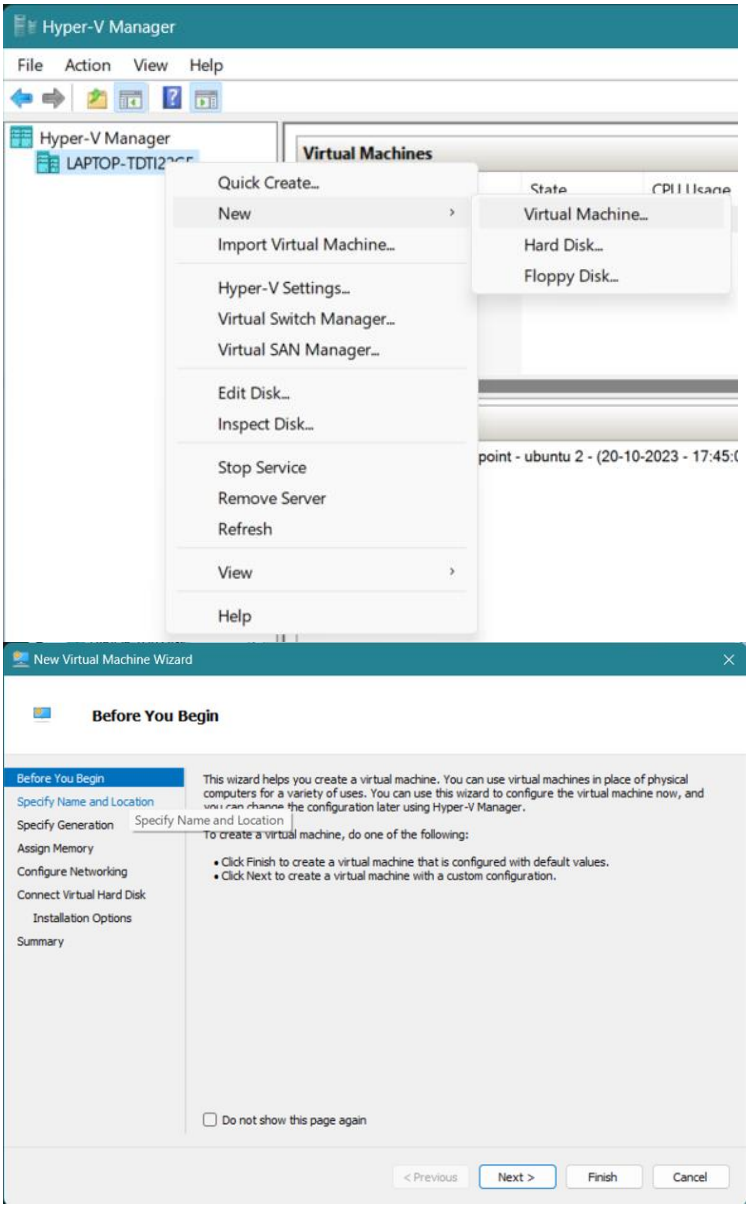
Enable Virtualization in BIOS: Using Hyper-V requires virtualization enabled in the motherboard's BIOS.
1. Open the BIOS by rebooting your computer and pressing the BIOS button (F1, F2, F10, F12, or Delete) when prompted.
2. The virtualization settings are located under the CPU or Advanced BIOS Settings category.

3.let's activate Hyper-V Manager from the Windows Optional Features menu.

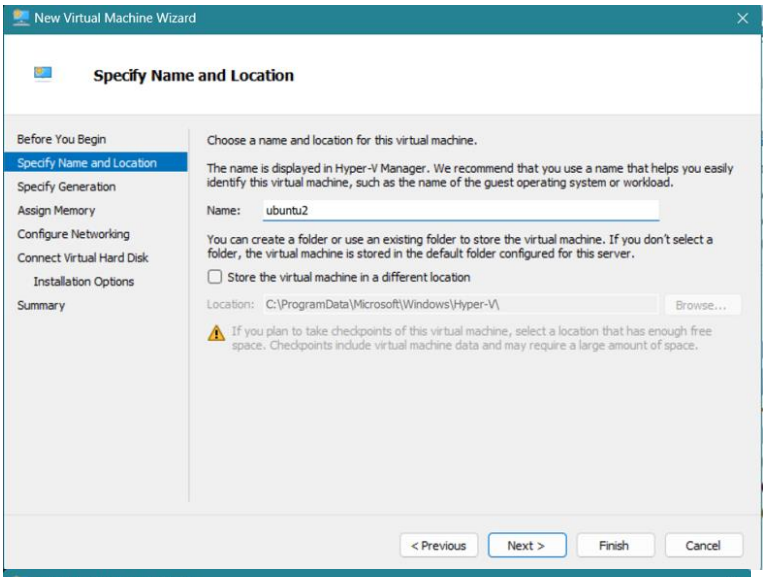


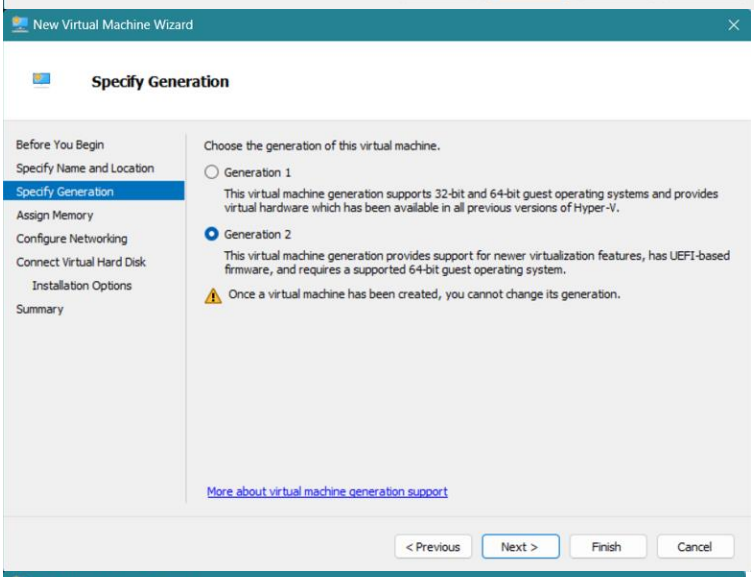
Crafting the Ubuntu VM:

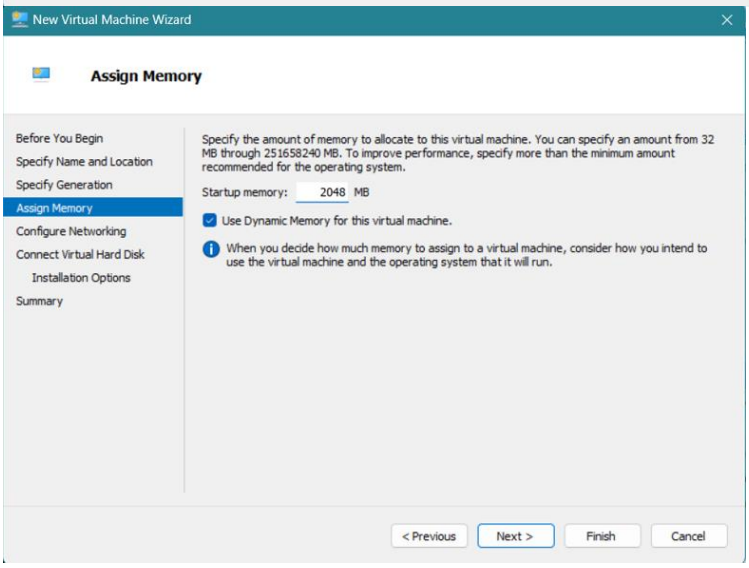
We'll craft a brand-new virtual machine (VM) within Hyper-V Manager, and we'll give it the name "ubuntu2". With that ready, we'll install Ubuntu 22.04 by loading the ISO file into the VM's DVD drive.

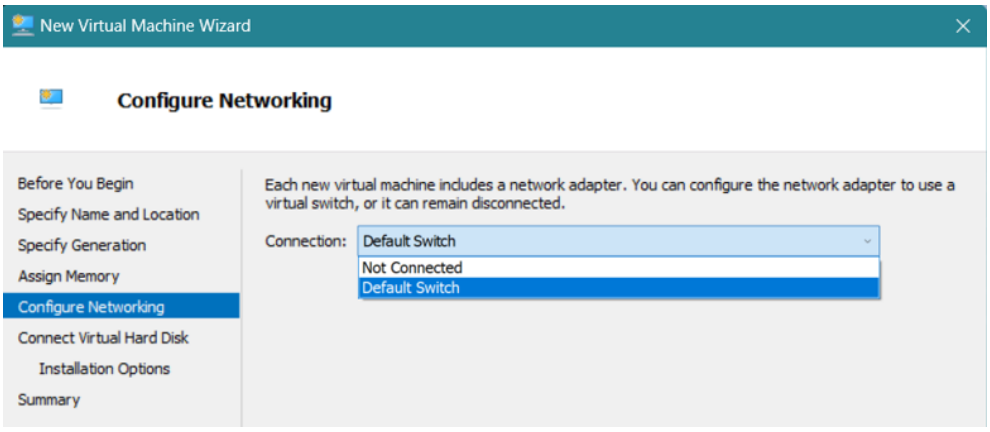

Hyper-V Manager
File Action View Help
Hyper-V Manager
LAPTOP-TDTI2
Virtual Machines
State
CPU Usage
Quick Create...
New
Import Virtual Machine...
Hyper-V Settings...
Virtual Switch Manager...
Virtual SAN Manager...
Edit Disk...
Inspect Disk...
Stop Service
Remove Server
Refresh
View
Help
Virtual Machine...
Hard Disk...
Floppy Disk...
point - ubuntu 2 - (20-10-2023 - 17:45:0

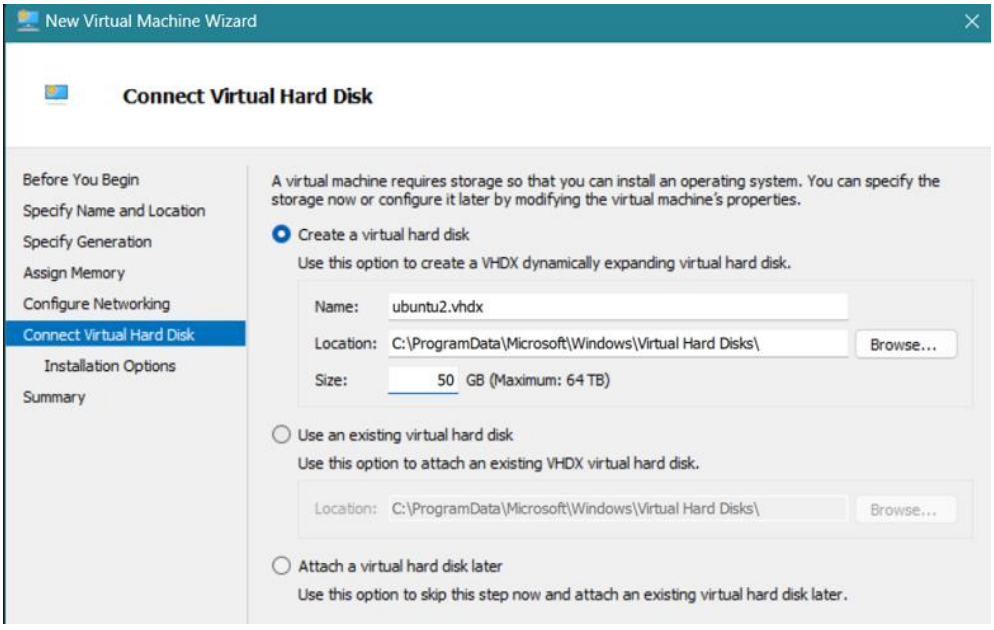
New Virtual Machine Wizard
Before You Begin
Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary
Specify Name and Location
This wizard helps you create a virtual machine. You can use virtual machines in place of physical computers for a variety of uses. You can use this wizard to configure the virtual machine now, and you can change the configuration later using Hyper-V Manager.
To create a virtual machine, do one of the following:
• Click Finish to create a virtual machine that is configured with default values.
• Click Next to create a virtual machine with a custom configuration.
Do not show this page again
< Previous
Next >
Finish
Cancel

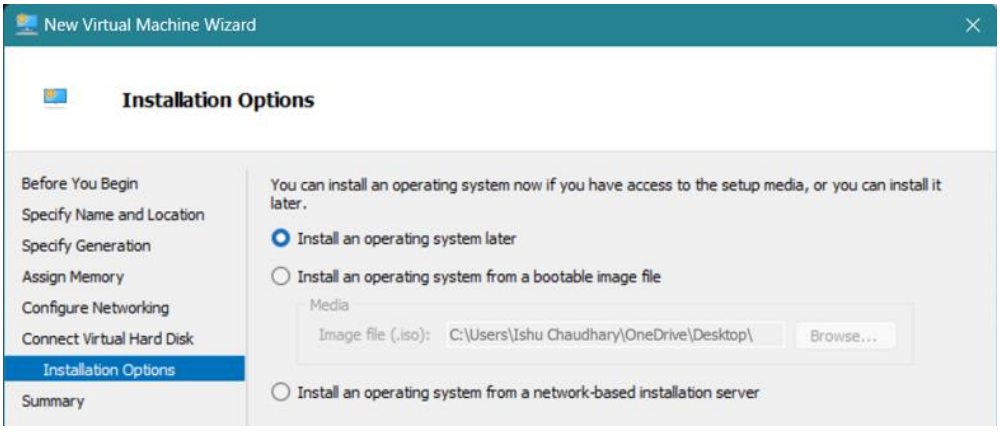

New Virtual Machine Wizard
Specify Name and Location
Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary
Choose a name and location for this virtual machine.
The name is displayed in Hyper-V Manager. We recommend that you use a name that helps you easily identify this virtual machine, such as the name of the guest operating system or workload.
Name: ubuntu2
You can create a folder or use an existing folder to store the virtual machine. If you don't select a folder, the virtual machine is stored in the default folder configured for this server.
Store the virtual machine in a different location
Location: C:\ProgramData\Microsoft\Windows\Hyper-V\
Browse...
If you plan to take checkpoints of this virtual machine, select a location that has enough free space. Checkpoints include virtual machine data and may require a large amount of space.
< Previous
Next >
Finish
Cancel

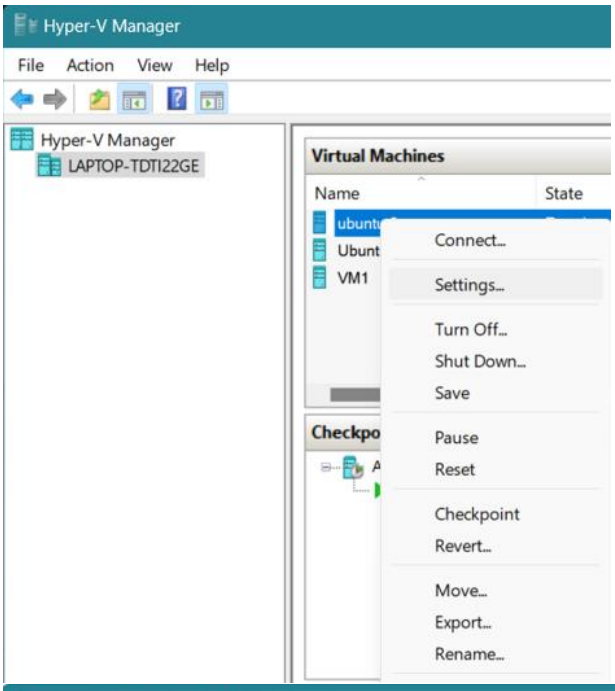

New Virtual Machine Wizard
Specify Generation
Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary
Choose the generation of this virtual machine.
Generation 1
This virtual machine generation supports 32-bit and 64-bit guest operating systems and provides virtual hardware which has been available in all previous versions of Hyper-V.
Generation 2
This virtual machine generation provides support for newer virtualization features, has UEFI-based firmware, and requires a supported 64-bit guest operating system.
Once a virtual machine has been created, you cannot change its generation.
More about virtual machine generation support
< Previous
Next >
Finish
Cancel

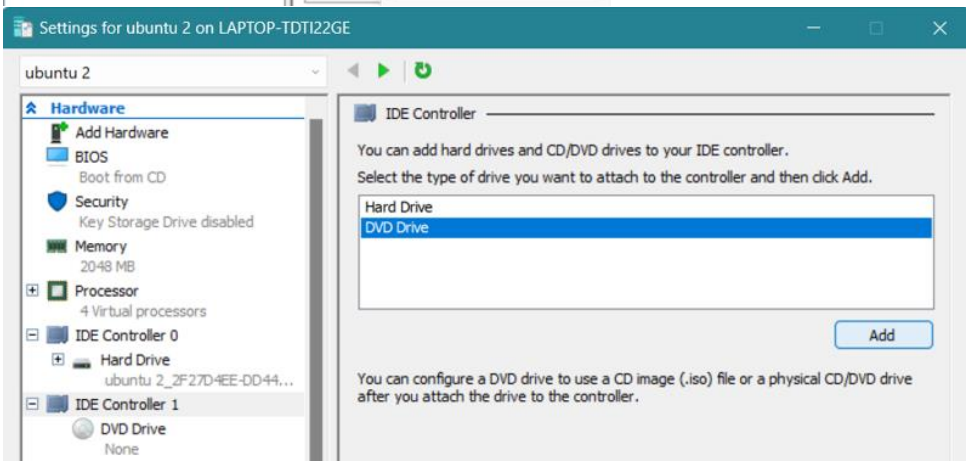

New Virtual Machine Wizard
Assign Memory
Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary
Specify the amount of memory to allocate to this virtual machine. You can specify an amount from 32 MB through 251658240 MB. To improve performance, specify more than the minimum amount recommended for the operating system.
Startup memory: 2048 MB
Use Dynamic Memory for this virtual machine.
When you decide how much memory to assign to a virtual machine, consider how you intend to use the virtual machine and the operating system that it will run.
< Previous
Next >
Finish
Cancel


New Virtual Machine Wizard
Configure Networking
Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary
Each new virtual machine includes a network adapter. You can configure the network adapter to use a virtual switch, or it can remain disconnected.
Connection:
Default Switch
Not Connected
Default Switch


New Virtual Machine Wizard
Connect Virtual Hard Disk
Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary
A virtual machine requires storage so that you can install an operating system. You can specify the storage now or configure it later by modifying the virtual machine's properties.
Create a virtual hard disk
Use this option to create a VHDX dynamically expanding virtual hard disk.
Name:
ubuntu2.vhdx
Location:
C:\ProgramData\Microsoft\Windows\Virtual Hard Disks\
Browse...
Size:
50
GB (Maximum: 64 TB)
Use an existing virtual hard disk
Use this option to attach an existing VHDX virtual hard disk.
Location:
C:\ProgramData\Microsoft\Windows\Virtual Hard Disks\
Browse...
Attach a virtual hard disk later
Use this option to skip this step now and attach an existing virtual hard disk later.


New Virtual Machine Wizard
Installation Options
Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary
You can install an operating system now if you have access to the setup media, or you can install it later.
Install an operating system later
Install an operating system from a bootable image file
Media
Image file (.iso):
C:\Users\Ishu Chaudhary\OneDrive\Desktop\
Browse...
Install an operating system from a network-based installation server

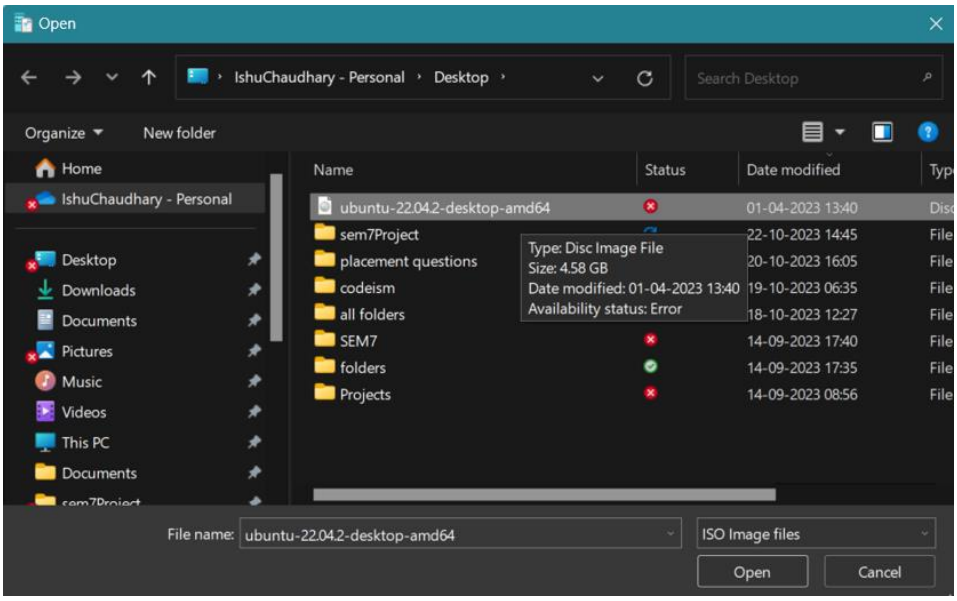




A full list of available files, including BitTorrent files, can be found below.

If you need help burning these images to disk, see the Image Burning Guide.

	Name	Last modified	Size	Description
	Parent Directory		-	
	SHA256SUMS	2023-08-10 18:33	202	
	SHA256SUMS.gpg	2023-08-10 18:33	833	
	ubuntu-22.04.3-desktop-amd64.iso	2023-08-08 01:19	4.7G	Desktop image for 64-bit PC (AMD64) computers (standard download)
	ubuntu-22.04.3-desktop-amd64.iso.torrent	2023-08-10 18:30	376K	Desktop image for 64-bit PC (AMD64) computers (BitTorrent download)
	ubuntu-22.04.3-desktop-amd64.iso.zsync	2023-08-10 18:30	11M	Desktop image for 64-bit PC (AMD64) computers (zsync metafile)



Installing Java:

Inside our Ubuntu VM, we'll sprinkle in the Java Runtime Environment with the magic command

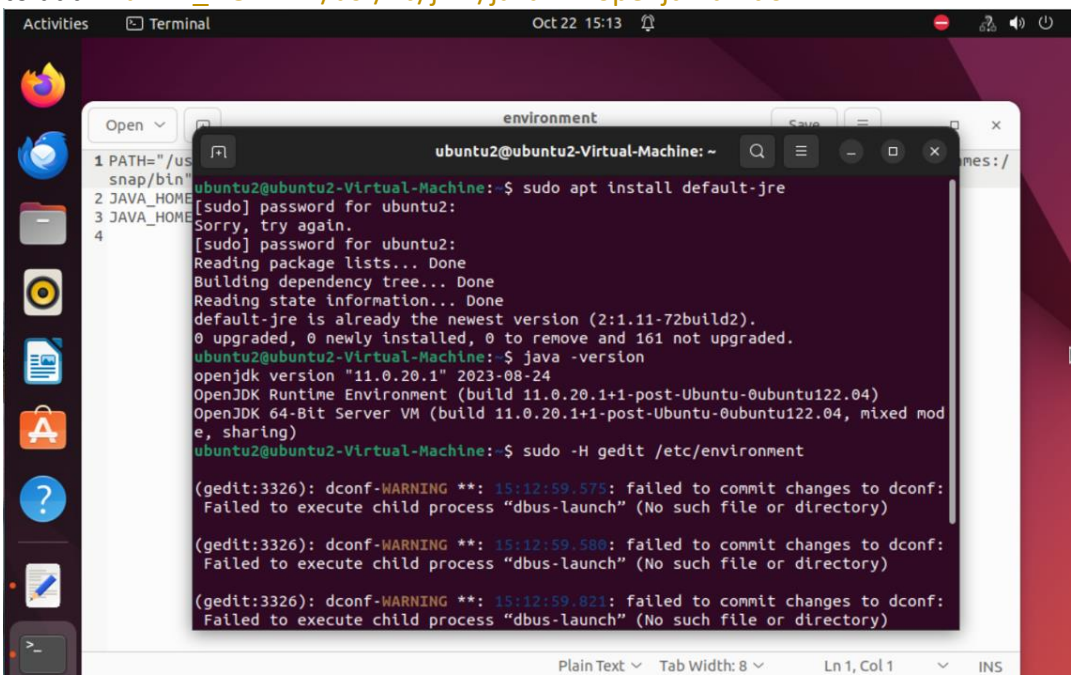
sudo apt install default-jre.

Fine-Tuning Environment Variables:

Now, let's fine-tune our environment by setting up Java environment variables. This involves using the command

sudo -H gedit /etc/environment

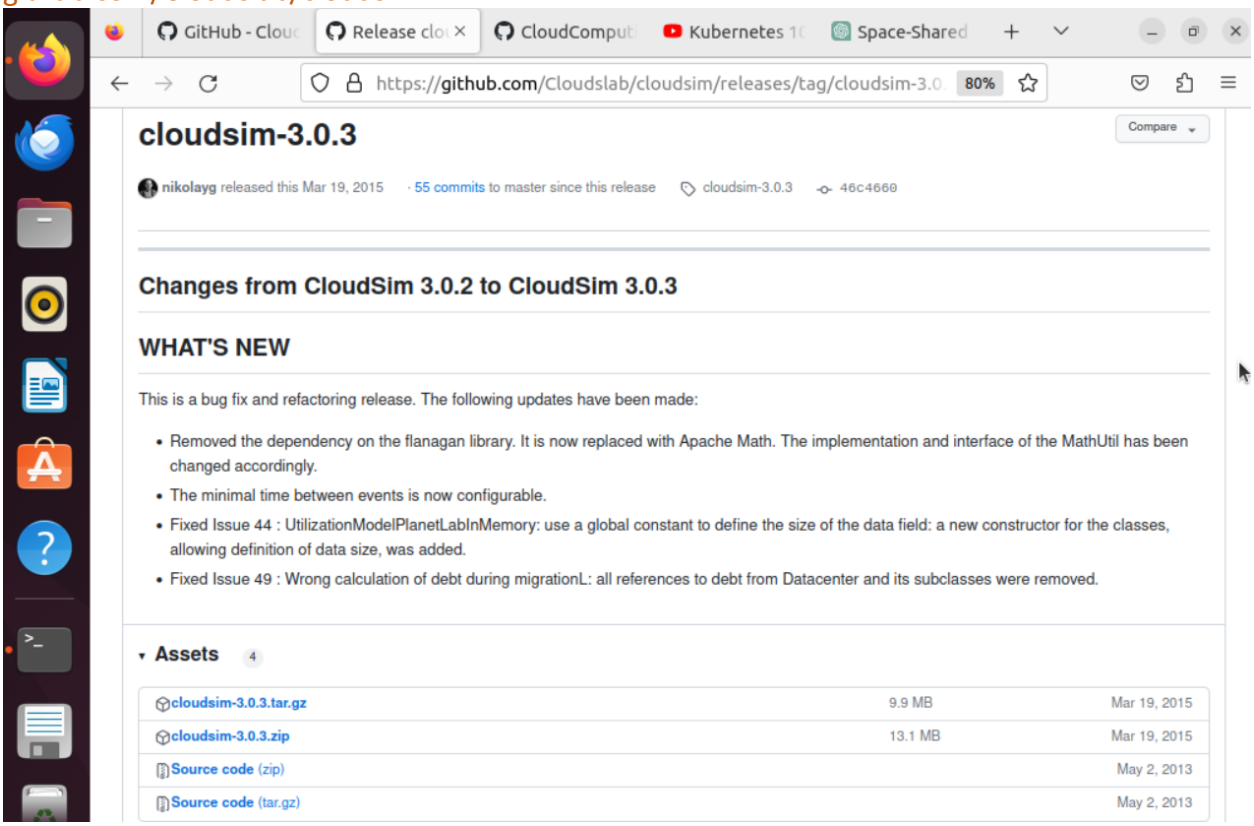
to add -> JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64".



Working with CloudSim

Snagging CloudSim:

We'll snag CloudSim 3.0.3 from a handy GitHub link, typically found on the CloudSim website.
github.com/Cloudslab/cloudsim



Customizing the Datacenter Broker:

The Datacenter Broker is our secret sauce in CloudSim. It acts as a mediator between cloud servers and the cloud infrastructure, orchestrating the provisioning of VMs and the scheduling of cloudlets.
You can give it a personalized twist by modifying the DataCenterBroker Java file to implement diverse cloud scheduling algorithms.

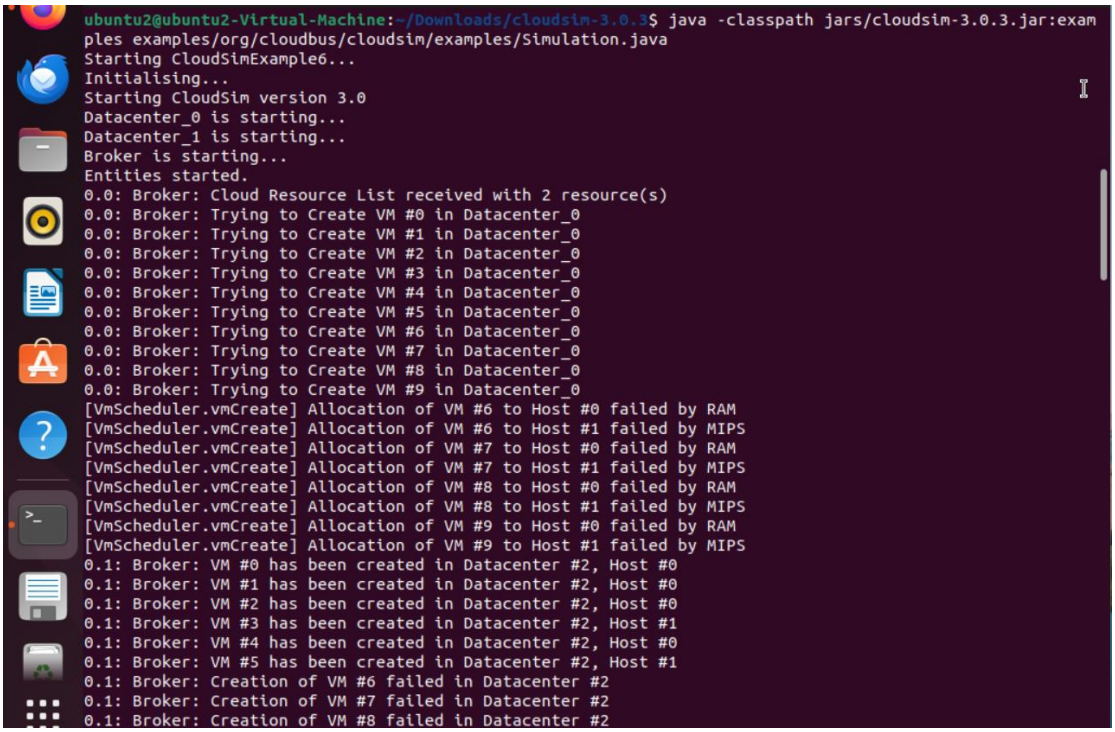
Crafting the Main Simulation File:

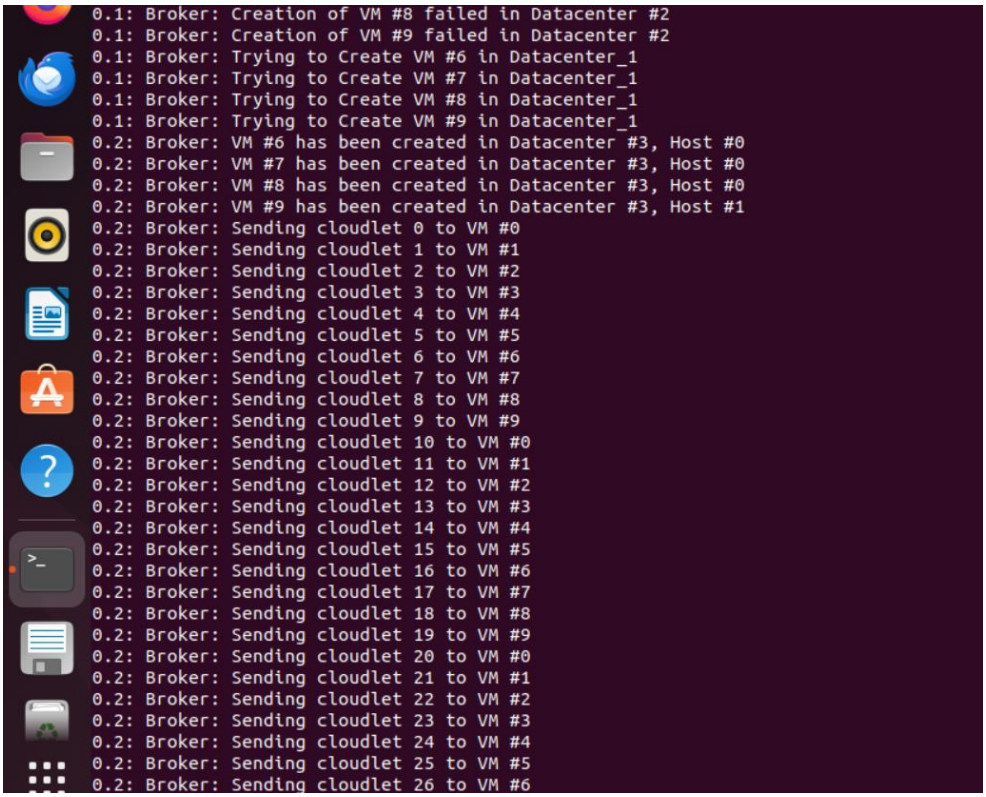
We'll whip up a Java file for our cloud computing simulation, naming it "Simulation.java." The main course includes

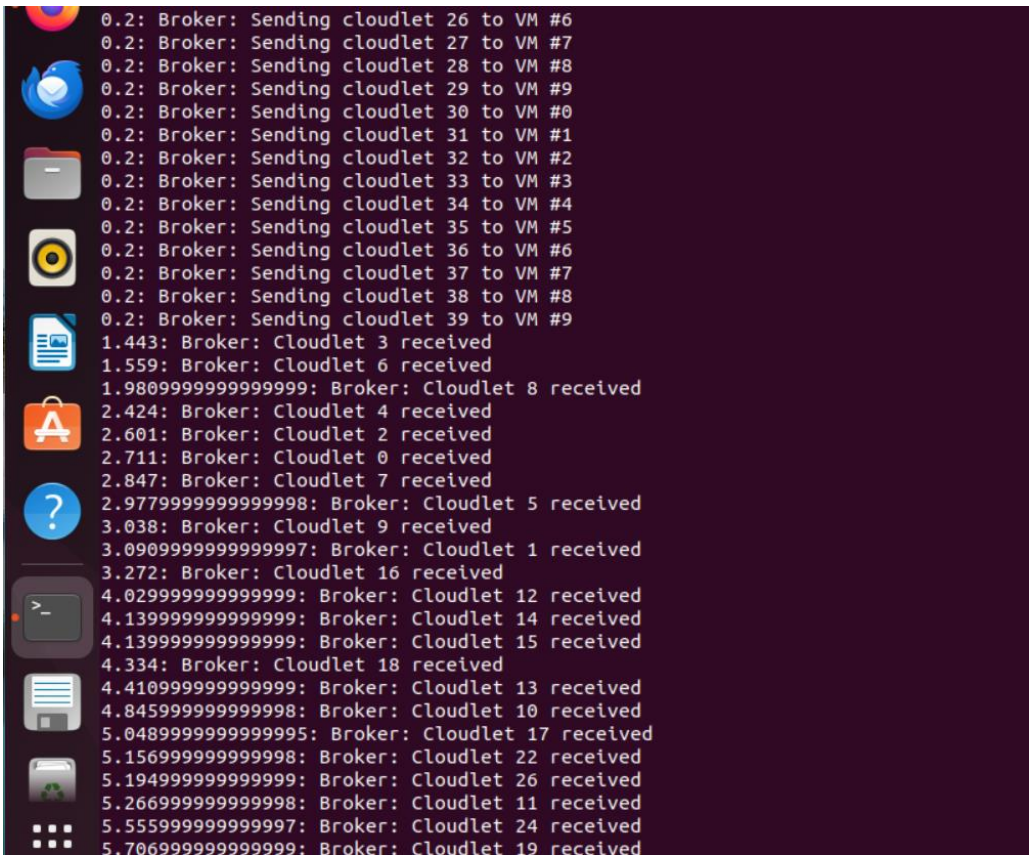
1.importing CloudSim packages and classes
2.static variables that stores list od VMs and Cloudlets
3.crafting VMs and cloudlets
4.Datacenter creation (with properties (RAM, CPU, Cores, OS, Architecture etc)
5.Broker Creation
4. submission of cloudlets and VMs to broker
5.simulation execution
6.simulation results
7.simulation termination
8.PrintCloudletList (prints cloudlet details like IDs, status, DataCenter IDs, VM IDs, execution time, start times, finish times and UserIDs)

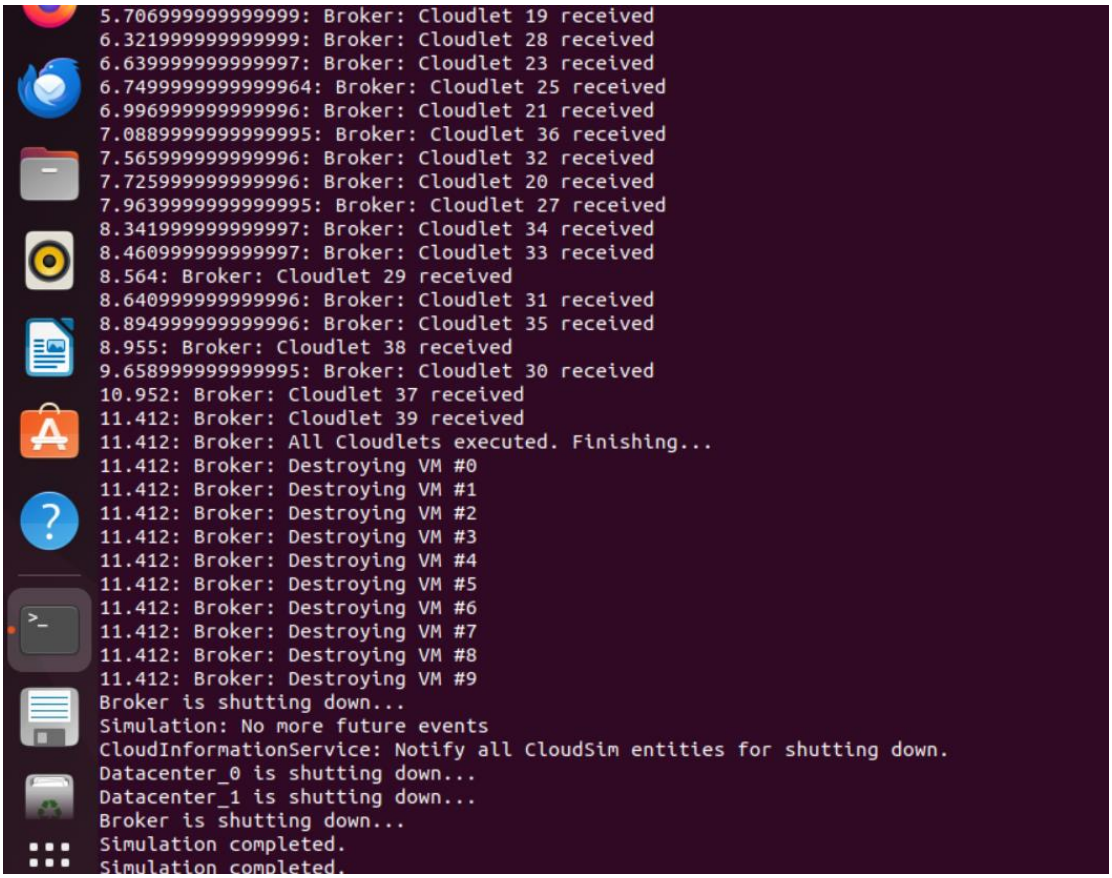
Running the Simulation

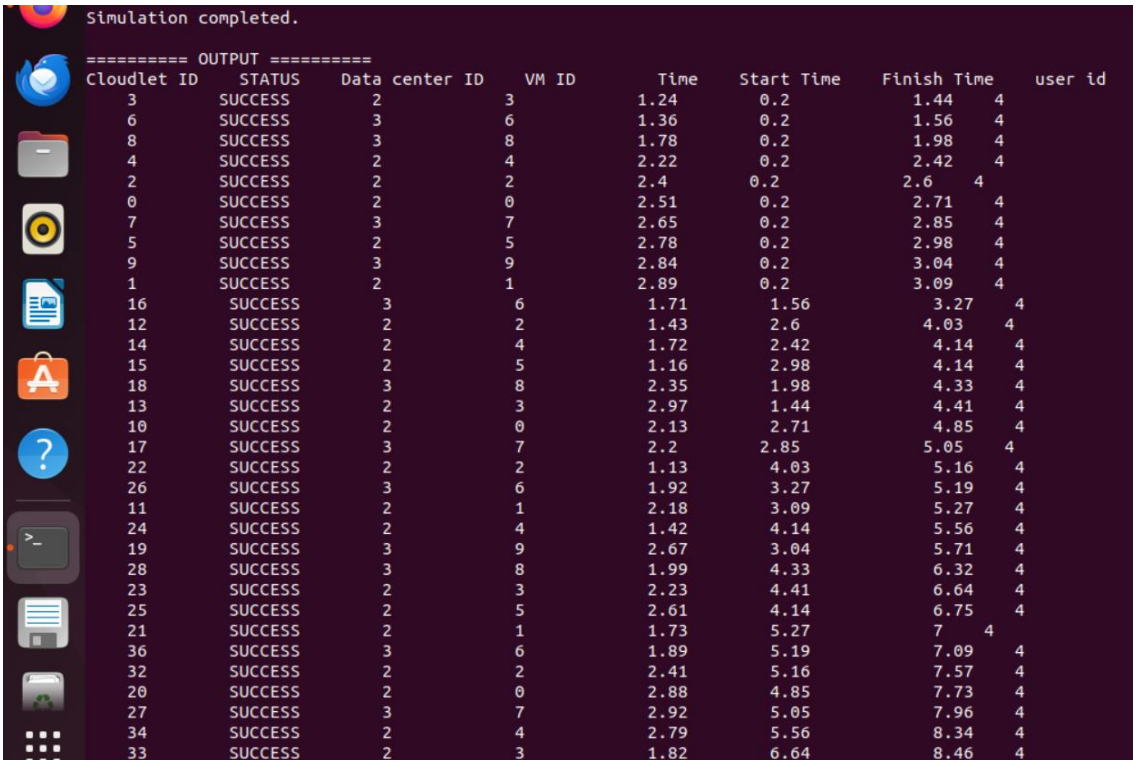
To run our simulation file, we used the command:
java -classpath java/cloudsim-3.0.3.jar:examples example/org /cloudbus/cloudsim/ examples/ simulation.java
I discovered this command in the process outlined in the readme.txt and examples.txt files located in the CloudSim 3.0.3 folder we extracted after downloading. We also delved into various tutorials on simulating cloud environments.




0.1: Broker: Creation of VM #8 failed in Datacenter #2
0.1: Broker: Creation of VM #9 failed in Datacenter #2
0.1: Broker: Trying to Create VM #6 in Datacenter_1
0.1: Broker: Trying to Create VM #7 in Datacenter_1
0.1: Broker: Trying to Create VM #8 in Datacenter_1
0.1: Broker: Trying to Create VM #9 in Datacenter_1
0.2: Broker: VM #6 has been created in Datacenter #3, Host #0
0.2: Broker: VM #7 has been created in Datacenter #3, Host #0
0.2: Broker: VM #8 has been created in Datacenter #3, Host #0
0.2: Broker: VM #9 has been created in Datacenter #3, Host #1
0.2: Broker: Sending cloudlet 0 to VM #0
0.2: Broker: Sending cloudlet 1 to VM #1
0.2: Broker: Sending cloudlet 2 to VM #2
0.2: Broker: Sending cloudlet 3 to VM #3
0.2: Broker: Sending cloudlet 4 to VM #4
0.2: Broker: Sending cloudlet 5 to VM #5
0.2: Broker: Sending cloudlet 6 to VM #6
0.2: Broker: Sending cloudlet 7 to VM #7
0.2: Broker: Sending cloudlet 8 to VM #8
0.2: Broker: Sending cloudlet 9 to VM #9
0.2: Broker: Sending cloudlet 10 to VM #0
0.2: Broker: Sending cloudlet 11 to VM #1
0.2: Broker: Sending cloudlet 12 to VM #2
0.2: Broker: Sending cloudlet 13 to VM #3
0.2: Broker: Sending cloudlet 14 to VM #4
0.2: Broker: Sending cloudlet 15 to VM #5
0.2: Broker: Sending cloudlet 16 to VM #6
0.2: Broker: Sending cloudlet 17 to VM #7
0.2: Broker: Sending cloudlet 18 to VM #8
0.2: Broker: Sending cloudlet 19 to VM #9
0.2: Broker: Sending cloudlet 20 to VM #0
0.2: Broker: Sending cloudlet 21 to VM #1
0.2: Broker: Sending cloudlet 22 to VM #2
0.2: Broker: Sending cloudlet 23 to VM #3
0.2: Broker: Sending cloudlet 24 to VM #4
0.2: Broker: Sending cloudlet 25 to VM #5
0.2: Broker: Sending cloudlet 26 to VM #6


0.2: Broker: Sending cloudlet 26 to VM #6
0.2: Broker: Sending cloudlet 27 to VM #7
0.2: Broker: Sending cloudlet 28 to VM #8
0.2: Broker: Sending cloudlet 29 to VM #9
0.2: Broker: Sending cloudlet 30 to VM #0
0.2: Broker: Sending cloudlet 31 to VM #1
0.2: Broker: Sending cloudlet 32 to VM #2
0.2: Broker: Sending cloudlet 33 to VM #3
0.2: Broker: Sending cloudlet 34 to VM #4
0.2: Broker: Sending cloudlet 35 to VM #5
0.2: Broker: Sending cloudlet 36 to VM #6
0.2: Broker: Sending cloudlet 37 to VM #7
0.2: Broker: Sending cloudlet 38 to VM #8
0.2: Broker: Sending cloudlet 39 to VM #9
1.443: Broker: Cloudlet 3 received
1.559: Broker: Cloudlet 6 received
1.9809999999999999: Broker: Cloudlet 8 received
2.424: Broker: Cloudlet 4 received
2.601: Broker: Cloudlet 2 received
2.711: Broker: Cloudlet 0 received
2.847: Broker: Cloudlet 7 received
2.9779999999999998: Broker: Cloudlet 5 received
3.038: Broker: Cloudlet 9 received
3.0909999999999997: Broker: Cloudlet 1 received
3.272: Broker: Cloudlet 16 received
4.029999999999999: Broker: Cloudlet 12 received
4.139999999999999: Broker: Cloudlet 14 received
4.139999999999999: Broker: Cloudlet 15 received
4.334: Broker: Cloudlet 18 received
4.410999999999999: Broker: Cloudlet 13 received
4.845999999999998: Broker: Cloudlet 10 received
5.0489999999999995: Broker: Cloudlet 17 received
5.156999999999998: Broker: Cloudlet 22 received
5.194999999999999: Broker: Cloudlet 26 received
5.266999999999998: Broker: Cloudlet 11 received
5.555999999999997: Broker: Cloudlet 24 received
5.706999999999999: Broker: Cloudlet 19 received

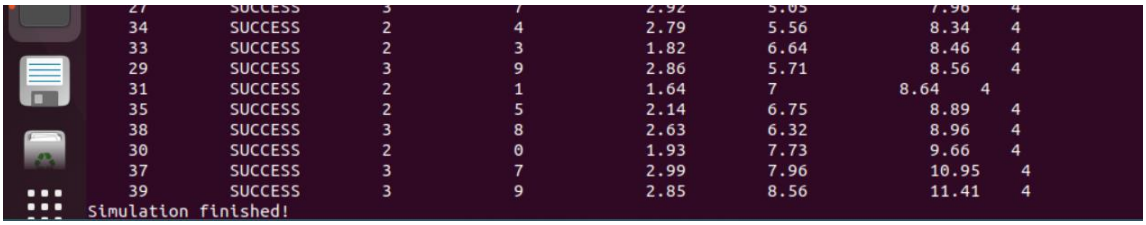




Simulation completed.

========== OUTPUT ==========

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time	user id
3	SUCCESS	2	3	1.24	0.2	1.44	4
6	SUCCESS	3	6	1.36	0.2	1.56	4
8	SUCCESS	3	8	1.78	0.2	1.98	4
4	SUCCESS	2	4	2.22	0.2	2.42	4
2	SUCCESS	2	2	2.4	0.2	2.6	4
0	SUCCESS	2	0	2.51	0.2	2.71	4
7	SUCCESS	3	7	2.65	0.2	2.85	4
5	SUCCESS	2	5	2.78	0.2	2.98	4
9	SUCCESS	3	9	2.84	0.2	3.04	4
1	SUCCESS	2	1	2.89	0.2	3.09	4
16	SUCCESS	3	6	1.71	1.56	3.27	4
12	SUCCESS	2	2	1.43	2.6	4.03	4
14	SUCCESS	2	4	1.72	2.42	4.14	4
15	SUCCESS	2	5	1.16	2.98	4.14	4
18	SUCCESS	3	8	2.35	1.98	4.33	4
13	SUCCESS	2	3	2.97	1.44	4.41	4
10	SUCCESS	2	0	2.13	2.71	4.85	4
17	SUCCESS	3	7	2.2	2.85	5.05	4
22	SUCCESS	2	2	1.13	4.03	5.16	4
26	SUCCESS	3	6	1.92	3.27	5.19	4
11	SUCCESS	2	1	2.18	3.09	5.27	4
24	SUCCESS	2	4	1.42	4.14	5.56	4
19	SUCCESS	3	9	2.67	3.04	5.71	4
28	SUCCESS	3	8	1.99	4.33	6.32	4
23	SUCCESS	2	3	2.23	4.41	6.64	4
25	SUCCESS	2	5	2.61	4.14	6.75	4
21	SUCCESS	2	1	1.73	5.27	7	4
36	SUCCESS	3	6	1.89	5.19	7.09	4
32	SUCCESS	2	2	2.41	5.16	7.57	4
20	SUCCESS	2	0	2.88	4.85	7.73	4
27	SUCCESS	3	7	2.92	5.05	7.96	4
34	SUCCESS	2	4	2.79	5.56	8.34	4
33	SUCCESS	2	3	1.82	6.64	8.46	4



Demystifying Simulation Details

In our mystical cloud realm, we've brewed up a simulation featuring not one but two Datacenters, a broker, 10 VMs, and 40 cloudlets.
Our scheduling wizardry is powered by "CloudletSchedulerSpaceShared," which is perfect for space-shared scheduling.

Code

https://github.com/lshu-dev/cloudComputingProjectSem7

Glossary

Processing Elements (PEs):

These are the brainiac cores inside a VM, responsible for churning through instructions and performing computations.

Apache ANT:

Think of this as your magical wand for building and deploying Java projects.
Datacenter Broker: This is the maestro within the CloudSim realm. It plays matchmaker between the cloud servers and the cloud infrastructure, orchestrating VM provisioning and cloudlet scheduling on behalf of the user.

Cloudlet:

Picture a cloudlet as a little spark of computation that leaps from a device to a nearby cloud server. It's a part of a larger application, and it dances its routine on the cloud server rather than the device, making our lives more efficient.

Broker:

In the world of cloud simulation, a broker is your trusty assistant. It represents you, the user, in the cloud computing world. It handles the nitty-gritty of resource allocation, including VMs and cloudlets, ensuring your needs and app demands are met.

Conclusion

With CloudSim, you unlock the power to explore and experiment with cloud computing scenarios, scheduling algorithms, and resource allocation strategies. By setting up a virtual environment, installing the necessary tools, and crafting your simulations, you embark on a journey of discovery in the world of cloud computing