

## Report:

### Comparison of Running a Heavy Application on a Local Host vs a Virtual Machine

#### Introduction:

We used a python script and ran it on the two platforms to compare the running of a heavy application on a Local Host vs a Virtual Machine. The evaluation is done upon the speed of execution, ease of deployment, and resource allocation.

#### Local host:

- MacOS with an Apple M2 Chip and 16GB Ram

#### Virtual Machine:

- Amazon Linux t3.medium, Number of vCPUs: 2

#### About the python script:

A python script, "image-recognition.py", was made which trains a Convolutional Neural Network for image recognition by using the CIFAR-100 dataset as its training dataset. Since each image is repeated 10 times to increase the dataset, this requires a lot of memory for training.

#### Observation:

- Environment Feedback:
  - o Local host:
    - Warnings were shown on local that the terminal is using a lot of memory
  - o VM:
    - No such warning was observed.
- Training Performance:
  - o Local Host:
    - Average time for each epoch 30s
    - Despite warning for heavy which might be a response of the local host OS, the training was done at a consistent speed.
  - o VM:
    - Average time for each epoch: 60s
    - Despite having more generous resources, when the container with the script was run, it seemed to underperform against the local host.
- Test Accuracy:
  - o It is almost the same for both environments indicating not much difference and that the overall performance of the model remains same despite the difference in runtime environments

Reasons for discrepancy in reasoning:

- The general thought process suggests that VMs are the preferred choice for heavy applications when comparing between running them locally or on VM.
- The reason why the local host performed better here is due to the architecture difference between the environments
  - o The Apple M2 silicon chip is optimized for efficiency and performance and TensorFlow has native support for the Apple Silicon. The 16 GB unified memory helps make the process fast and optimized for both CPU and GPU tasks
  - o The VM in AWS used is t3.medium that have 2 vCPUs and are optimized for burst workloads which means they do not offer consistently high CPU performance.

Why are VMs still preferred:

- While Apple's M2 worked efficiently, there are many reasons why VMs are still the best choice for heavy workload applications.
  - o **Scalability:** Local hosts cannot be scaled to accommodate increase in workload. The python script used just happen to work better because of the reasons outlined above. VMs offer scalability that allow users to use more powerful instances so that they can handle much heavier workloads.
  - o **Hardware:** Cloud providers that offer VMs also have hardware that can accommodate training a very large neural network which is not possible on local host.
  - o **Cost-effectiveness:** In the long run, paying for what you run and scaling helps save money for the users when deploying VMs rather than purchasing hardware for a specific task.
  - o **Remote collaboration:** As the word suggests, teams can work together when a VM is deployed by accessing the instance which cannot be done on a local host. A heavy workload application typically requires inputs from different users and VMs help make collaboration possible.

5.

**How long did the execution (training and testing procedures) take?**

Time cumulative for training and testing: VM = 661seconds; Local Terminal: 97 seconds

**Did you face any issues running this program on your local machine?**

No issues were faced on local machine

**How running the application on VM differs from local host?**

For local host, I had to install packages in my environment and run the script. For VM, I had to containerize the application so no issues for python environment would be there. Other than that, the running of the application did not differ drastically unless we consider the setting up of the VM part which inherently does not occur on local host.