VMs vs Containers

By: Chinmay W. & Ronan M.

1. Background

Information about Containers & Virtual Machines (VMs)

What is Virtualization?

 Creating a layer of abstraction between hardware and applications.

Multiple simulated environments from a single, physical hardware system.

 Resource sharing by multiple processes without conflicts.

What are Virtual Machines?

A virtual computer with its own CPU, memory, storage and network interface.
A virtual environment with an allocation of resources.
Run their own OS and can be used as a "sandbox".



What are Containers?





Azure Container (Microsoft)

Google Container

What are Containers?

- Virtualize the operating system instead of virtualizing the hardware.
- Abstraction at the app layer.
- Shared parts of the OS are read only while each container has its own mount for writing.



VMs

- Multiple OS instances.
- Allocates required memory.
- More isolation, but heavier.
- Boot up time of several minutes.
- More hardware costs.
- Maintenance time.
- Isolated resources and OS instances highlighting better security.

Containers

- Single OS instance.
- Requires less memory space.
- Less isolation, but lightweight.
- Containerized applications can start up in less than a second.
- Decreased hardware costs.
- Continuous Integration.
- Shared kernel leading to possible interference between containers.

How do these run "on the cloud"?



2. Simulations/Tests

Simulation/Testing approach and process used

Cloud Providers Used

General purpose—Dv3

Microsoft Azure

Balanced CPU and memory

Our Dv3 family is the latest generation of our general purpose VMs powered by Intel[®] Xeon[®] processors. It is appropriate for a variety of workloads.

Specs:

D2 v3 2 vCPU(s) 8 GiB RAM

Starting from

\$0.0125/hour





\$**5**/mo \$0.007/hour

1 GB / 1 CPU 25 GB SSD disk 1000 GB transfer

Simulation/Testing Steps

Run tests on both VMs & Containers Microsoft Azure and Digital Ocean Ο Create scripts for gathering data CPU, Memory, Networking Usage/Load 0 Run simulations/tests at both idle and at load Test Configuration: Node.js Web Application

Simulation/Testing Measured Details

CPU

User, System, Idle, SW Interrupts, Steal Time

Memory

Total, Used, Free, Shared, Buffer/Cache

Networking

- Throughput, Packet Loss, and RTT
- Setup and Configuration Time/Experience

3. Results

Analysis of data gathered from simulations/tests

CPU Usage Comparison

System CPU Usage (Load) Comparison



Time (s)

Memory Usage Comparison

Memory Usage (Load) Comparison



Networking Usage Comparison

RTT Comparison - Digital Ocean VM

10

8

6

2

0

0

Time (ms)

💻 time_idle 🛛 💻 time_load

RTT Comparison - Digital Ocean Container



Idle RTT:

20

Sequence Number

- RTT Min: 1.2 ms
- RTT Avg: 1.6 ms
- RTT Max: 2.4 ms

Load RTT:

40

- RTT Min: 0.85 ms
- RTT Avg: 1.92 ms
 - RTT Max: 9.12 ms

Idle RTT:

- RTT Min: 1.47 ms
- RTT Avg: 1.96 ms
- RTT Max: 3.4 ms

Load RTT:

- RTT Min: 1.11 ms
- RTT Avg: 15.12 ms
- RTT Max: 105.17 ms

Networking Usage Comparison



💻 tx_Load 🛛 💻 rx_Load

Azure VM - TX/RX Network Packet Rate (Load)

🗕 tx Load 🛛 💻 rx Load



4. Conclusion

Summary of project results

Results Analysis

Developer Experience & Setup Time better on Digital Ocean

- VMs use more memory & have more CPU utilization
 - Compared to respective cloud provider's Container offering
- Memory
 - Highest memory usage: Azure VM @ ~49%
 - Lowest memory usage: Azure Container @ ~7.5%
 CPU
 - Highest sys. CPU utilization: Azure VM @ ~0.46%
 - Lowest sys. CPU utilization: DO Container @ ~0.20%

Thank You

Questions?