Comparison Between Docker on CoreOS and Virtual Machine Implementation

Team 1

David Alban (dalban, Lead)
Abhimanyu Kumar (akumar24)
Jaspreet Kaur (jkaur3)
Lahari Kommi (lkommi)
Mrunal Mozarkar (mmozark)
Shyam Ramakrishnan (sramakr9)

Problem Statement

- Both Virtual Machine and Container technologies provide certain level of abstraction and isolation

- Adopting either of these technologies require a careful consideration of various factors like:
  - Services offered
  - Performance
  - Cost

Functional Design Requirements

Aside from the infrastructure already provided, we require the following components:

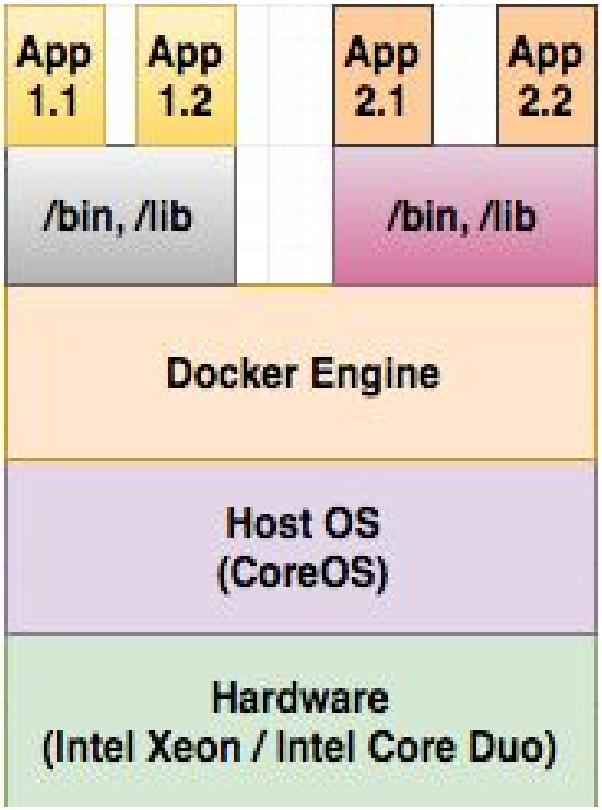
1. Ability to measure memory utilization
2. Ability to measure CPU utilization
3. Ability to measure network performance
4. Ability to measure disk utilization
5. Dashboard to display statistics

Non-functional Design Requirements

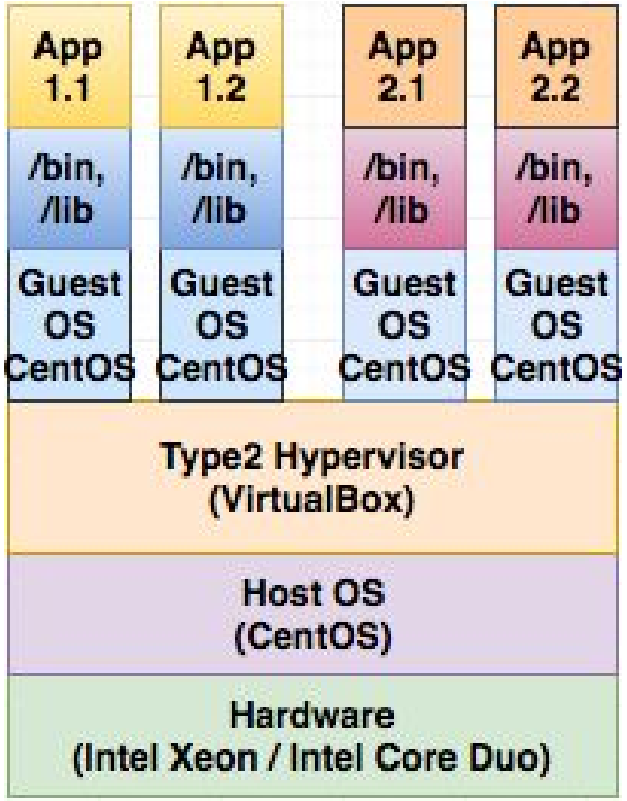
1. Display live statistics in real-time
2. Display data in graphical format
3. Run application automatically from dashboard
4. Documentation

System Functional Diagram

Container Architecture



VM Architecture



System Environment

Bare-Metal

	Dell T310 (x2)	**Dell 390**	**Dell T610**
	Container | VM	Container | VM	Container | VM
OS	CoreOS | CentOS 7	CoreOS | CentOS 7	CoreOS | CentOS 7
RAM	8 GB	2 GB	26 GB
Processor	Intel Xeon X3430 - 2.4GHz	Intel Core 2 Duo 6600 - 2.4 GHz	Intel Xeon E5620 - 2.4GHz
Storage	500 GB	250 GB	1000 GB

System Environment

Virtual Machine

	Dell T310 (x2)	Dell 390	Dell T610
OS	CentOS 7 Minimal	CentOS 7 Minimal	CentOS 7 Minimal
RAM	1 GB	1 GB	1 GB
vCPUs	1	1	1
Storage	8 GB	8 GB	8 GB

Performance Measuring Applications

- Stress and Stress-ng: For stress testing on physical subsystem using configurable amount of CPU compute, memory, cache, I/O, and disk stress on the system

- Bonnie++ and IOzone3: Benchmark suites aimed at performing a number of simple tests (read, write) of hard drive and file system performance

- iPerf: To measure the quality of network, maximum bandwidth performance, loss and other parameters like TCP, UDP and SCTP

Benchmark Test Workflow

- Front end application: Simple web UI to trigger the desired test cases on the Container and VM systems
- Python program: Used to connect to the nodes and execute the test benchmark applications
- Datasource: Interfaced to fetch results from the testing applications and provide data to the visualization tool
- Performance visualization: Tools like grafana that renders graphs to display performance comparison using the data from datasource

Schedules

11/10	• Research and design • Selecting the monitoring tools
11/14	• Install CoreOS on four machines • Install CentOS on four machines
11/16	• Learn about docker and container functionalities • Install VirtualBox and configure Docker
11/18	• Install docker containers with monitoring tools • Commit and push new images to Docker
11/20	• Install guest OS on VirtualBox • Install monitoring tools on guest OS
12/1	• Dashboard integration • Collect statistics
12/3	• Documentation and finishing touches

Tasks and Responsibility

Tasks	Members Responsible
Researching and Deciding Topology Design	All
CoreOS(host) installation for Container System	Shyam, Mrunal, Jaspreet
CentOS(host) installation for VM System	Abhimanyu, David, Lahari
Docker Setup	Shyam, Mrunal, Jaspreet
VirtualBox Setup	Abhimanyu, David, Lahari
Researching Performance Measuring Applications	All
Container Setup for individual Applications	Shyam, Mrunal, Jaspreet
CentOS VM(guest) installation for VM System	Abhimanyu, David, Lahari
Installing Applications on CentOS VM	Abhimanyu, David, Lahari
Setup Datasource and Dashboard to Display Results	Abhimanyu, David
Perform Benchmark Tests	Mrunal, Shyam
Verify and Validate Test Results	Jaspreet, Lahari
Document and Compile Report	All

Verification and Validation

Test Case #	Test Case	Application	Expected Result
T1	Comparing CPU performance between containers and VM	*stress, stress-ng*	Containers would give better performance than VMs.
T2	Comparing memory utilization	*stress, stress-ng*	Containers would give better performance than VMs.
T3	Comparing Disk I/O performance	*Bonnie++, IOZONE3*	Containers would give better performance than VMs.
T4	Comparing network performance between containers and VM	*iperf*	VMs would perform better than containers with respect to latency

Verification and Validation

Test Case #	Test Case	Application	Expected Result
T5	Scalability (with respect to number of containers and VMs on host with same specifications)	*Light-weight application like 'Midori' and heavy-weight application like 'Chrome', etc.*	Comparatively more number of containers would be supported as compared to number of VMs given a host with same specifications
T6	Recovering from system crash	*n/a*	VMs are stateful , whereas containers are stateless in nature unless committed as a new image
T7	Ease of deployment	*n/a*	Containers would be easy and faster to deploy
T8	Security	*n/a*	Containers would be more vulnerable to various attacks like stack buffer overflow
T9	Flexibility	*n/a*	Guest OS in containers would depend on host OS but no such restriction for VMs

Results/Progress

Tasks	Progress
Researching and Deciding Topology Design	Completed
CoreOS(host) installation for Container System	Completed
CentOS(host) installation for VM System	Completed
Docker Setup	Completed
VirtualBox Setup	Completed
Researching Performance Measuring Applications	Completed
Container Setup for individual Applications	Completed
CentOS VM(guest) installation for VM System	Completed
Installing Applications on CentOS VM	Completed
Setup Datasource and Dashboard to Display Results	Ongoing
Perform Benchmark Tests	Pending
Verify and Validate Test Results	Pending
Document and Compile Report	Pending

"ANY QUESTIONS?"