CS 240
Computer Systems

#13: Virtualization and Abstractions

Oct. 12, 2021 · Wade Fagen-Ulmschneider

Operating Systems: A Great Illusionist

Throughout this entire course, we have discussed how the operating system abstracts away the complexity of real systems:

- As a process, it appears that we have _____.
- ...and has _____!

Do we need additional abstractions?

Virtualization

Big Idea:

- All states $\mathbf{S_X}$ can be represented on a host system $\mathbf{H(S_X)}$.
- For all sequences of transitions between $\mathbf{S_1} \Rightarrow \mathbf{S_2}$, there is a sequence of transitions between $H(\mathbf{S_1}) \Rightarrow H(\mathbf{S_2})$.

What is a "machine"?

- Language Virtualization:

- Process Virtualization:

- System Virtualization:

Language Virtualization: Example w/ a JVM

Initial State ($\mathbf{S_1}$):		
Transition ($\mathbf{S_1} \Rightarrow \mathbf{S_2}$):		

System #1
`COPY r1 1`
`SHIFTL x 2`
`ADD x r1`
	System #2
	`COPY r1 x`
	`SHIFTL x`
	`SHIFTL x`
	`ADD x r1`
		System #3
		`COPY r1 x`
		`ADD r1 x`
		`ADD r1 x`
		`ADD r1 x`
		`ADD r1 x`

Final State ($\mathbf{S_2}$):		

Process Virtualization: Example w/ Rosetta and the M1 chip

Initial State ($\mathbf{S_1}$):
Transition ($\mathbf{S_1} \Rightarrow \mathbf{S_2}$):
Final State ($\mathbf{S_2}$):

System Virtualization: Your CS 240 Virtual Machine / EC2

- Type 1 Hypervisor:

- Type 2 Hypervisor:

Q: How has this changed the deployment of software?

Abstractions Using Virtualization Technologies
Modern cloud computing is full of abstractions on top of virtualization technologies. There's at least five major categories that abstract different levels of complexity away from the end user.

Legend:

	Abstracted by Provider/Vendor		Customer Managed Unit of Scale		Customer Managed

IaaS Infrastructure as a Service	**CaaS** Containers as a Service	**PaaS** Platform as a Service	**FaaS** Functions as a Service	**SaaS** Software as a Service
Data	Data	Data	Data	Data
Functions	Functions	Functions	Functions	Functions
Applications	Applications	Applications	Applications	Applications
Runtime	Runtime	Runtime	Runtime	Runtime
Containers*	Containers	Containers*	Containers*	Containers*
OS	OS	OS	OS	OS
Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
Hardware	Hardware	Hardware	Hardware	Hardware

**: May or may not be present on a specific technology stack.*

Infrastructure as a Service (IaaS)
When you choose to host your technology on IaaS, you are provided virtualization running on top of managed hardware.

- **Example:** Amazon EC2 / Google Compute Engine

- Why IaaS?

Containers:
Containers provide an _____ of a system that can be deployed in an isolated environment on heterogeneous systems.

Key Modern Container Technology: _____

- **As a container developer,** you build a Dockerfile that specifies the **snapshot** of the system you want to provide and then **build** that snapshot into a _____.

- Example (creating a docker image):

mp3/Dockerfile

```
1 FROM gcc:latest
2 COPY ./docker/entrypoint.sh /
3 RUN chmod +x entrypoint.sh
4 ENTRYPOINT ["/entrypoint.sh"]
```

Line 1 (FROM):

Lines 2 (COPY):

Line 3 (RUN):

Line 4 (ENTRYPOINT):

To build it:

```
$ docker build --tag mp3-docker .
```

- As **a user of a container**, you specify the name of the docker image that you want to use to launch that image:

```
$ docker run -it --rm -v "pwd":/mp3  mp3-docker "make"
$ docker run --rm -it -p 27017:27017 mongo
```