# Section 15: Special Topics

CS 162

August 12, 2020

## Contents

## 1 Virtual Machines, Containers

1. What happens when an application running on the Guest OS issues a system call?

   The system call is forwarded to the VMM (since the host knows that a Guest OS application is running on that core), which forwards it to the Guest OS transparently.

2. What happens when the Guest OS tries to execute a privileged instruction?

   The guest traps to the Host OS, which performs the privileged instruction (modified to operate on the Guest OS' state) on behalf of the guest.

3. Which page table is used for address translation when an application running on the Guest OS is executing?

   A shadow page table, maintained by the VMM, is used for address translation.

## 2 Orchestration (Optional)

1. Explain what a container orchestrator does and why its necessary.

   (a) Provides a clean, easy to use virtualization of your data center resources
   (b) It manages the sharing of these resources between different applications and users
   (c) It provides common services such as load balancing, authentication, service discovery, etc
   (d) It manages the complexity of deploying and running containerized microservice applications

2. What does Mesos (without additional libraries) offer that is similar to modern container orchestrators? How is it different from an orchestrator the likes of Kubernetes?

   Mesos provides data center resource virtualization and can be thought of as an operating system for the data center / cloud. Its different because it concerned primary with resource allocation management rather than the container and service lifecycles, so it doesn't do as much to manage the complexity of deploying and running microservice applications.

3. Explain why reconciliation is a useful method of desired state management.

   Reconciliation is an example of eventual consistency. Its useful because we can make changes the the central desired state, and that change will get propagated through reconciliation to those who care about it. It makes managing the desired state much easier because the responsibility of making sure the actual state is valid then falls on the local actor rather than the user making the change. In Kubernetes, changes to the desired state are done declaratively by the user using high level concepts like Deployments and Services. Through reconcilation, the Kubernetes components responsible will update the actual state to match what is desired by the user.

4. Explain what each of the following Kubernetes objects are: Node, Pod, Deployment, Service, Ingress

   - Node: an object that represents a virtual or physical machine where containers can be deployed. A Kubelet sits on each Node.
   - Pod: the unit of execution, a small group of one or more containers that is always deployed together
   - Deployment: an object that creates a number of replicas for a given Pod template
   - Service: an onject that most closely represents a microservice. Has a persistent abstract name and load balances to all Pods registered under it.
   - Ingress: an endpoint where outside traffic is allowed to enter your cluster, responsible for load balancing and routing