CS 552/452
Spring 2023
Midterm Exam 1
03/1/2023
Time Limit: 180 Minutes

Name: ____________________

B-Number: ____________________

This exam contains 12 pages (including this cover page) and 6 questions. Total of points is 80. Please notice that less is more — more isn't always better, especially when it comes to answer. Please keep the answers short and to the point. If there are any factual errors and/or the answers are totally irrelevant to the questions, marks will be deducted.

Grade Table (for teacher use only)

Question	Points	Score
1	12	
2	12	
3	13	
4	14	
5	8	
6	21	
Total:	80	

1. (12 points) **Question 1 – Machine-level Virtualization!**

   (1) Please describe the high-level design ideas to virtualize CPU, given that all of the sensitive instructions are the privilege instructions? (6 points)

   (2) Please describe the high-level design ideas to virtualize CPU, given that only part of the sensitive instructions are the privilege instructions? (6 points)

2. (12 points) **Question 2 – Virtualization is Costly!**
(1) Please state the main sources that negatively impact I/O performance in KVM or Xen based virtualization solutions in comparison with the native (i.e., no virtualization). (3 points)

(2) To achieve high I/O performance under Xen/KVM virtualization, what are the possible optimization solutions, and why do you think these solutions will work? (name three) (3 points)

(3) Under a round-robin vCPU scheduler, suppose the time-slice (i.e., maximum time that a vCPU can run) is 30 ms. Four VMs (VM1 to VM4, each with 1 vCPU) share a single CPU. When a client sends an I/O request to VM1, how long does the client expect to receive the response from VM1 in the *worst* case? (2 points)

What are the results if the time-slice is 10 ms and 1 ms, separately? (2 points)

What are the gains and losses using a small time-slice (e.g, 1 ms) in comparison with a large one? (2 points)

3. (13 points) **Question 3 – Containerization!**
   (1) How does the OS kernel support containerization? (2 points)

   (2) Can you guess what is the key functionality in Docker? (2 points)

   (3) Why is I/O performance under containerization much better than KVM or Xen based virtualization? (If you don't think so, please also justify.) (3 points)

   (4) Then would the I/O performance of containers be the same as the native case (i.e., running processes directly on an OS), and why? (3 points)

(5) Containers are much light-weight in comparison with machine-level virtualization solutions like VMs. However, in practice (e.g., in public clouds) we do not completely replace VMs with containers. Why is that? (3 points)

4. (14 points) **Question 4 – Other Sandboxes!**

(1) gVisor (Google's container solution) provides another isolation mechanism for container, which intercepts application system calls and acts as the guest kernel. Instead of passing system calls to the native kernel, gVisor implements a substantial portion of the Linux system surface. Thus each container may have its own user-level kernel. This design supposes to have **better security** properties than traditional ones (e.g., Docker containers). How does gVisor achieve this? (4 points)

(2) Actually, gVisor trades performance for such better security properties. Which types of applications do you think suffer *less* in terms of performance drop when we use gVisor containers, and why? Do you have some ideas to mitigate performance overhead of gVisor? (4 points)

(3) Our ultimate goal is to achieve the speed of containers and the security of VMs. Do you have some ideas to achive this goal? (6 points)

5. (8 points) **Question 5 – Server Consolidation**
One typical usage scenario for containers and/or virtual machines (i.e., server virtualization techniques) is server consolidation. Server consolidation is an approach to place multiple virtualized servers (either in containers or virtual machines) on the same physical machine in order to reduce the total number of servers or server locations that an organisation requires. The practice originally developed in response to the problem of server sprawl, a situation in which multiple, under-utilised servers take up more space and consume more resources than can be justified by their workload. Later, the concept of server consolidation has been well materialized in cloud platforms where a single physical machine is shared by multiple cloud tenants.

Figure 1 shows the total performance (summed from all VMs running on the same physical host) changes as we consolidate more VMs on the same physical host.

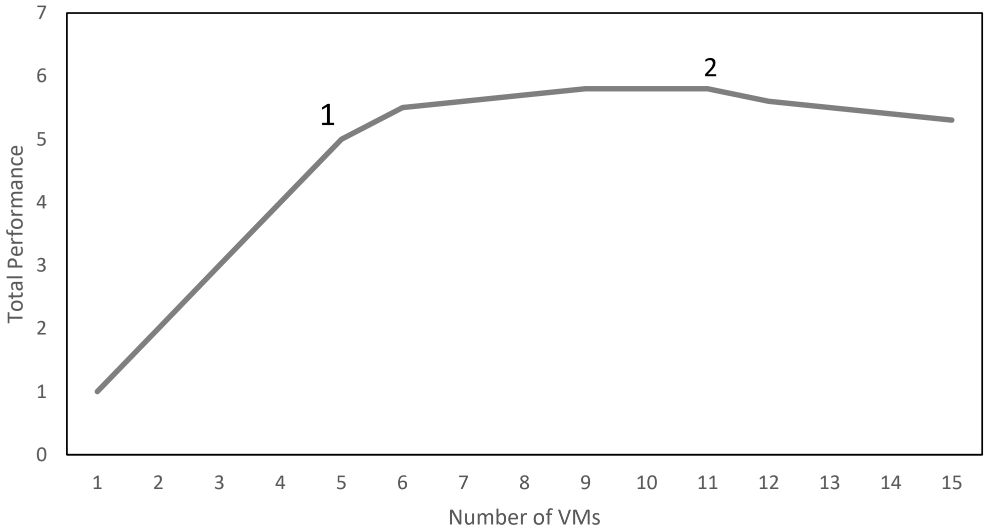

Figure 1: Total performance changes as more VMs consolidate on a single physical host.

(1) As you can observe, after point 1 in Figure 1, the total performance stop increasing. Can you guess why? (4 points)

(2) After point 2 in Figure 1, the total performance start dropping with more VMs, and why? (4 points)

6. (21 points) **Question 6 – Orchestration**

(1) What are the key motivation that you think for many IT companies (e.g., Netflix, Amazon, Uber) to move their services to a microservices based architecture? (3 points)

(2) In which situation(s), you prefer NOT to use micro-services based architecture for designing your applications? (3 points)

(3) What is the key goal that Kubernetes wants to achieve? (3 points)

(4) Can Kubernetes choose VMs as the hosting service (instead of containers)? (3 points)

(5) Kubernetes uses “Pod” as the basic abstraction. Why doesn’t it use “container” as the basic abstraction? (3 points)

(6) What is the “service” object used for in Kubernetes? (3 points)

(7) How does Kubernetes provide high availability? (3 points)