

ECE792/CSC791-038 Homework Assignment #2

Virtualization

Due Tuesday, October 8, 2018

No late homework will be accepted; turn in whatever you have completed.

Provide CLI output/Screenshot to support your answer

Problem 1. (10 Points) Reading datasheets. Read the Cisco cloud service router (CSR 1000V) datasheet.
<https://www.cisco.com/c/en/us/products/collateral/routers/cloud-services-router-1000v-series/datasheet-c78-733443.html> .

What is the CPU/Memory requirement for 500 Mbps, 1000 Mbps, and 5000 Mbps throughput for CSR routers. List three features of CSR for each of the following: a) Networking, b) Security, c) Management interface. What would be per year cost to use two 1000 Mbps CSR in Amazon cloud (Hint: Search CSR in AWS market place)?

Problem 2. (10 Points) Creating VMs. Create a VM (name the VM as *< your – unity – id >VM1*) and use apt get to install iperf Traffic generator and wireshark in newly created VM. We will use this VM frequently, in other problems in this homework. Answer the following questions related to the VM you just created.

1. What is the IP address and MAC address of the: a) VM's NIC, b) Bridge l2 and SVI (switch virtual Interface) and c) hypervisor NIC used to go to internet?
2. Ping google.com from your VM. Use Wireshark to capture packets at a) the output interface of the VM, b) the output interface of the hypervisor. List 4 tuples (source IP, destination IP, source MAC, and destination MAC) of a packet going out of the VM and your hypervisor. Are these tuples the same or different? If different, explain why.

Problem 3. (25 Points) virsh - managing VMs. In all tasks, provide the appropriate verification commands to verify that the task was successfully executed.

1. Use libvirt-CLI methods to add a network (name it as *< your – unity – id >NETWORK2*) in bridge mode.
2. Use libvirt-CLI methods to add an interface to your VM to connect to *< your – unity – id >NETWORK2*.
3. Use libvirt-CLI methods to clone your VM (name it as *< your – unity – id >lab2VM2*).
4. List MAC and IP addresses of all interfaces of each VM.
5. Ping one VM from the other using IP from the *< your – unity – id >NETWORK2* subnet. Use wireshark on the VMs and List down 4 tuples (srcIP, Dest IP, srcMAC and dest MAC) of packet going out at first VM and received at second VM. Are the tuple fields same or different for the packet? if different, explain why.

6. Set up a UDP traffic flow between the two VMs using iperf to answer the following question.
 - What is the maximum UDP throughput achieved?
 - Which is the bottleneck resource, CPU, memory, or I/O? Why? Provide logs of system commands to support your answer.
 - Does change in the packet size improves the throughput? Explain your answer and provide the maximum throughput achieved and corresponding packet size.
7. Set up a TCP traffic flow between the two VMs with default packet sizes. What is the maximum TCP throughput achieved? Explain any difference in throughput as compared to UDP traffic flow. Tune any TCP parameter to increase the throughput. Explain your TCP tuning experiment with supporting logs.

Problem 4. (20 Points) Ansible. Write two playbooks to do the following task.

1. An Ansible playbook to
 - Create an L3 network using OVS switch and give it a name **unityid-netl3**.
 - Create two VMs connected to network **unityid-netl3**.
2. Ansible playbook to collect logs (CPU load averages) from both host and the two guest created in part1 at 1 minute granularity (You can use your scripts from Homework 1). Store the collected in the /var/customlogs/logs folder. Log file should be in CSV format with following entry: hostname, cpulmin, cpu5 min, cpu 15min

Problem 5. (20 Points) Python - Libvirt API. Use the python libvirt API to do the following:

1. *Host Information* Obtain host information of your interest (at least 5 items).
2. *Guest Domain Information* Obtain guest information of your interest (at least 5 items).
3. *Performance monitoring* Write a python application to monitor VCPU and VMEM of all VMs. Your application should the following:
 - (a) print all VMs in ascending order, based on CPU /memory usage.
 - (b) print and also log in a file an alert message (Vm name, time stamp, CPU usage) if CPU usage crossed a threshold T (take from user input)

The TA will give CPU or MEM as the argument to your program and your program should return the list of VMs sorted by the given argument.

Bonus (10 Additional points): Your program should also make a sorted list of moving averages for all VMs. The polling interval + moving window size should be user defined parameter.

Problem 6. (15 Points)

What if: Set up a lab experiment to support your explanation to answer the following questions. What breaks if:

1. Two VMs connected to same bridge (bridge mode) have: (a) same mac addresses, (b) same IP addresses.
2. Two VM connected to different bridge (both bridges in 'bridge' mode) have: (a) same mac addresses, (b) same IP addresses.
3. Two VMs connected to different bridge (both Bridges in routed mode) have: (a) same mac addresses, (b) same IP addresses.