

ECE792-038 Homework Assignment #4

Containers and tracing Linux

Due Thursday, November 29, 2018

Problem 1. (50 Points)

Docker Containers:

1. Use docker to create the container topology shown in Figure 1. The Containers (LC1, CS1, CS2) are connected via Bridge 1, Containers (LC2, CS3 and CS4) are connected by Bridge 2, and Container (LC2, CS5) are connected by the Bridge 3. Rest all links between containers should be done with veth pair interfaces. Also all containers should have one interface connected to default NAT bridge (not shown the figure).

Following are the two tunneling requirements :

- (a) Bridge 1 and Bridge 3 are in same network. Traffic between the containers connected to bridge 1 and bridge 3 should use L2 Overlay. (HINT: Implement VXLAN)
 - (b) LC1 and LC2 should have a GRE tunnels to pass traffic between LC1 and LC2 bridge networks.
2. Automate this solution, so that you can connect customer's containers with a required network type (support gre, vxlan, l3 and l2 network type) on **demand** . Keep leaf spine switch (LC and SC configuration) fabric architecture as initial setup. Next, take input file as the following CSV to create containers and required network:

Name src container, Name destination container, Network type

CS1, CS2, Bridge

CS1X, CS2X, VXLAN

CSX, CSY, L3

CSA, CSB, GRE

Make other required assumptions. TA will test your answer with a similar CSV file.

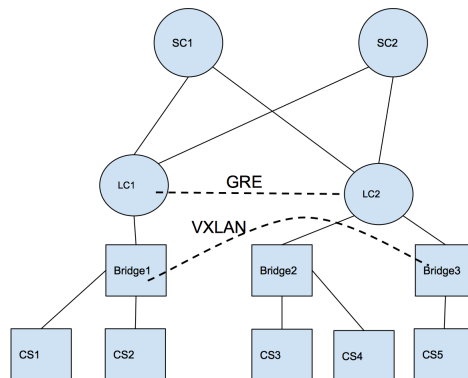


Figure 1: Containers topology.

Problem 2. (50 Points)

VM vs Container using Tracing Performance of the VMs or containers depends upon availability of resources and the resource isolation mechanism. In this question, you will compare the performance of the container and VM implementations. Before you simulate a VM or container topology as given in Figures 2 and 3, make sure all other VMs/or containers are in shutdown state.

Answer the following questions and make reasonable assumptions if needed. (HINT: Use Strace/fttrace tools to consider IPstack performance)

1. Configure VMs as shown in Figure 2. Allocate the **maximum** amount of resources (CPU/Memory) you can to this new VM. Run Iperf traffic between the VMs. What is the observed throughput in the system? Explain and provide calculations for the limit on throughput (make reasonable assumptions).
2. Configure an Iperf container (similar to the one given in lab exercises) as shown in Figure 3. Run Iperf traffic between the containers. What is the observed throughput? Explain in sufficient detail the limit on throughput. Show your calculations (make reasonable assumptions).
3. Are container and VM performance test results similar? If not, explain the differences.
4. Compare the performance (throughput) of the linux IPstack for different packet sizes. Which functions in IPstack are getting impacted more due to different size of packets?

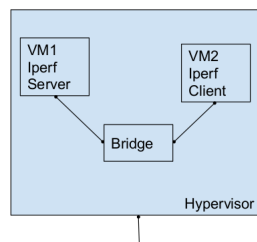


Figure 2: Containers VM problem: Two iperf VMs in the hypervisor.

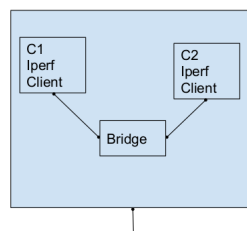


Figure 3: Containers vs VM problem: Two iperf containers in the host.

Problem 3. *EXTRA - CREDIT (20 Points)- Kubernetes Networking*

Setup experiments to compare throughput performance of three different Kubernetes Plugin. To measure networking throughput, run iperf UDP traffic between two kubernetes container connected with different networking plugins. Take atleast 10 times, the five minute throughput reading for each plugin.