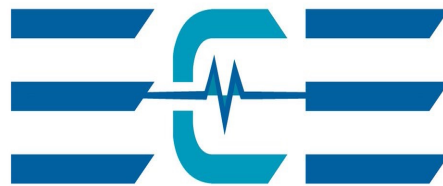


# A Scalable Container-based Virtualized Data Center Emulation Framework



Gaurav Gautam, Sandhya Rathee, Preetam Patil, and Parimal Parag



# Outline

- Motivation
- Features
- Architecture
- Experimental Demo

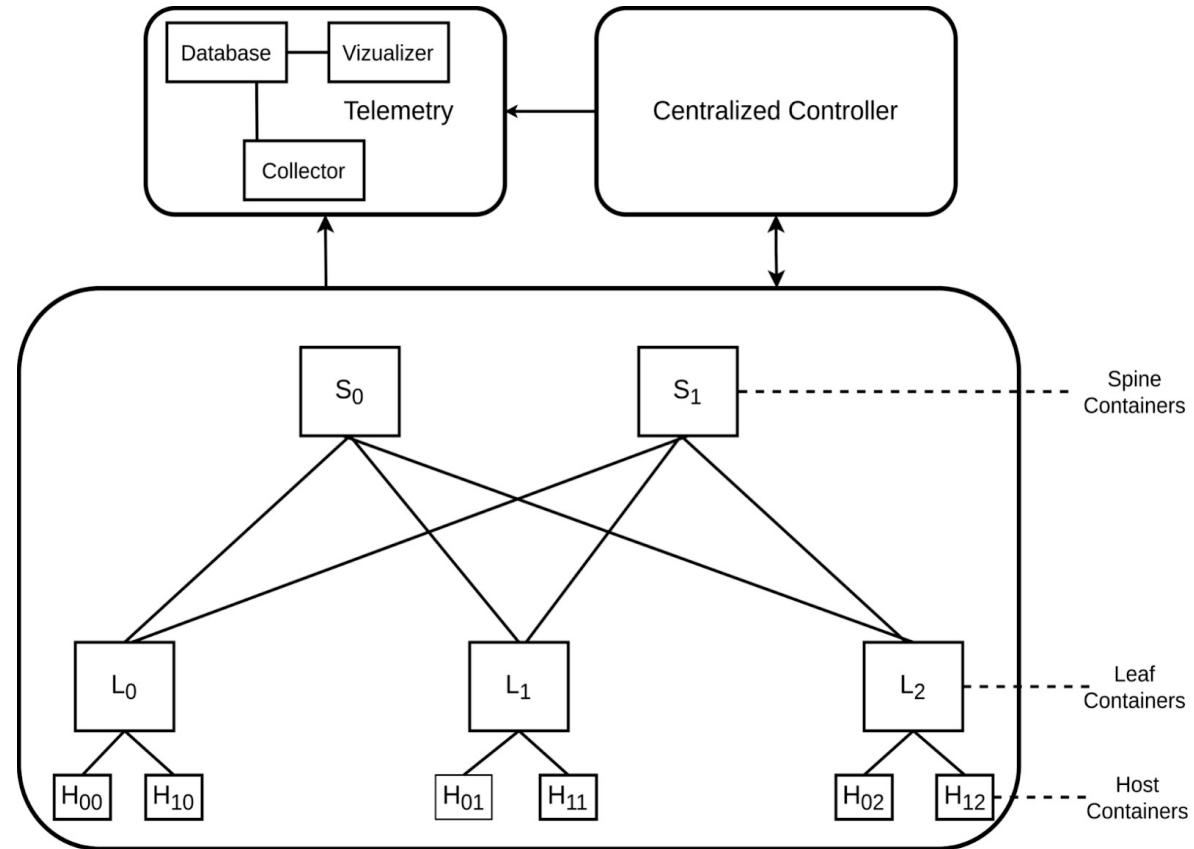
# MOTIVATION

- To satisfy on-demand services and to drive operational efficiency, resources are provisioned dynamically within the Data Center Network (DCN)
- To implement and experiment the validation of proposals solutions for existing problems in DCNs requires a scalable model of data centers with programmable data plane.
- Mininet is commonly used to emulate the networks.
- The major limitations with mininet are
  - Scalability - mininet instance runs within a single host
    - Shares kernel space
    - Ability to Resource allocation (cpu and memory)
- Realistic hyperscale data center consuming proportionally miniscule computation and storage resources is a challenge

# FEATURES

- We present a scalable container-based virtualized platform for programmable DCN
- We provide a scalable topology where you just need to update the script for topology parameters
- It uses containers to instantiate the topology, switches are inside container
- The proposed framework provide freedom to choose different software switches
- Docker container acting as switch with FRRouting and Bmv2 switch

# ARCHITECTURE



# Experimental Demo

- Topology: Leaf-spine topology
- Experiment 1:

Switch: Simple docker container acting as a switch.

Routing: BGP routing using FRRouting.

Traffic: Iperf Traffic

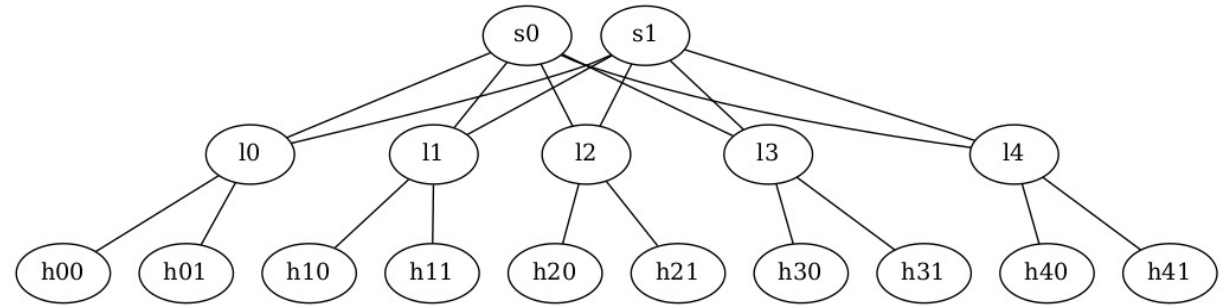
Telemetry: Bandwidth utilization measured and displayed using Grafana.

- Experiment 2:

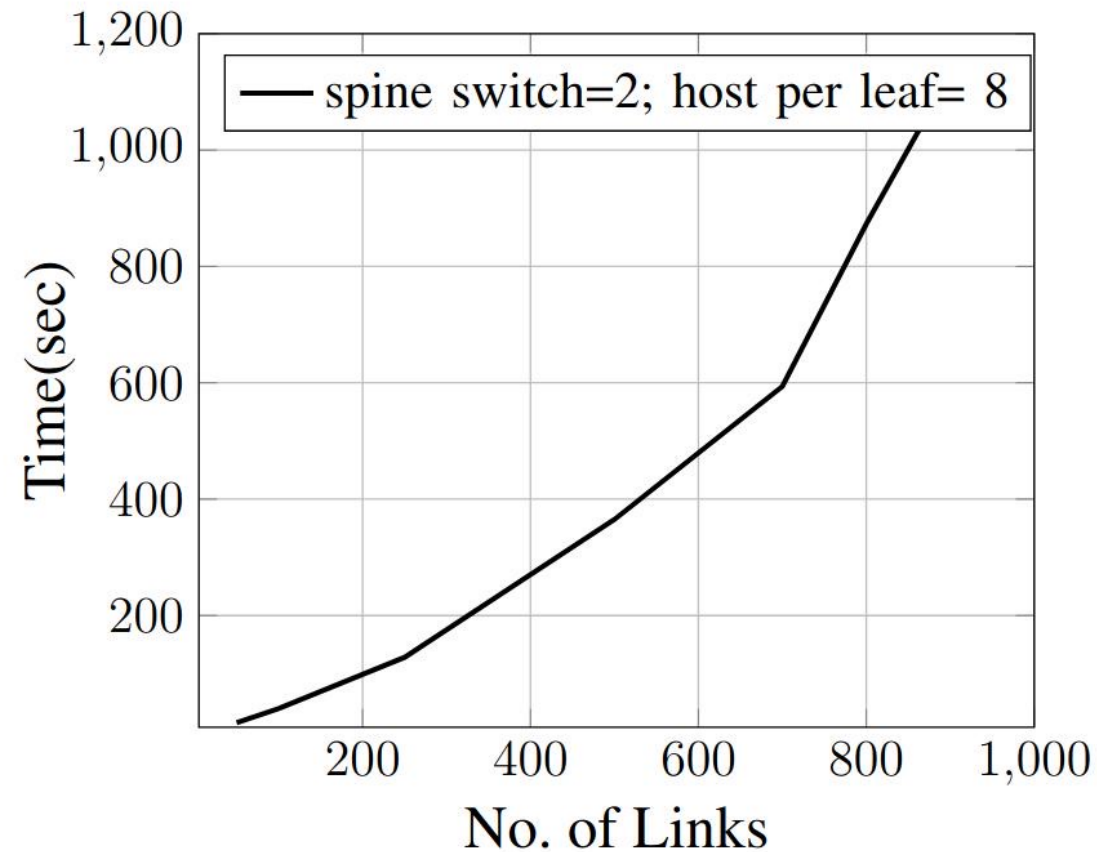
Switch: Bmv2 switch inside docker container

Routing: Forwarding using P4

In-Band Telemetry measurement: Path tracing, Queue Latency



# Number of links Vs time to create and start topology for leaf-spine topology



# Demo 1

- Topology: Leaf-spine topology
  - Switch: Simple docker container acting as a switch.
  - Routing: BGP routing using FRRouting.
  - Traffic: Iperf Traffic
  - Telemetry: Bandwidth measured and displayed using Grafana.
- 
- Spine switch =2, Leaf switchs = 50, Host per leaf= 2
  - No. Of containers =  $2 \times 50 + 50 + 2 = 152$
  - No. Of Links=  $50 \times 2 + 50 \times 2 = 200$



# Demo 2

- Topology: Leaf-spine topology
- Switch: Bmv2 switch inside docker container
- Routing: Forwarding using P4
- In-Band Telemetry using P4 code.
- Measurement: Path tracing, Queue Latency

# Acknowledgment

This work is supported by the Centre for Networked Intelligence (a Cisco CSR initiative) at the Indian Institute of Science, Bengaluru.

