School of Engineering & Applied Science

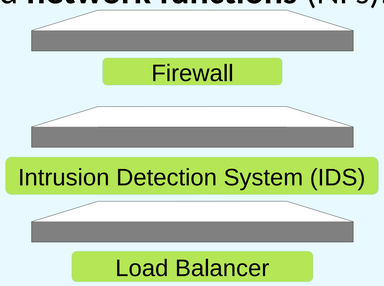
THE GEORGE WASHINGTON UNIVERSITY

Distributed openNetVM: Automatic Scaling for Enterprise Network Function Virtualization Using Commodity Hardware

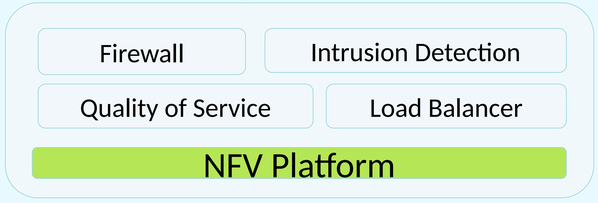
Phil Lopreiato
The George Washington University

Background

- Traditionally, networks are comprised of individual *hardware components* called **network functions** (NFs):



- This model is very *expensive* and *inflexible* which makes changes/upgrade difficult
- Trends in networking produced network function virtualization (NFV)
  - Cost effectiveness of software
  - Flexibility of software
  - All network functions on one or multiple hosts
  - Like NFs are grouped into **services**



Results

- We can easily route packet flows through dynamic topology "on-the-fly" without performance loss (at network line speeds)
- The distributed openNetVM system can automatically scale NF capacities to meet the current demand
- Improved allocation and utilization of computing resources

openNetVM Platform

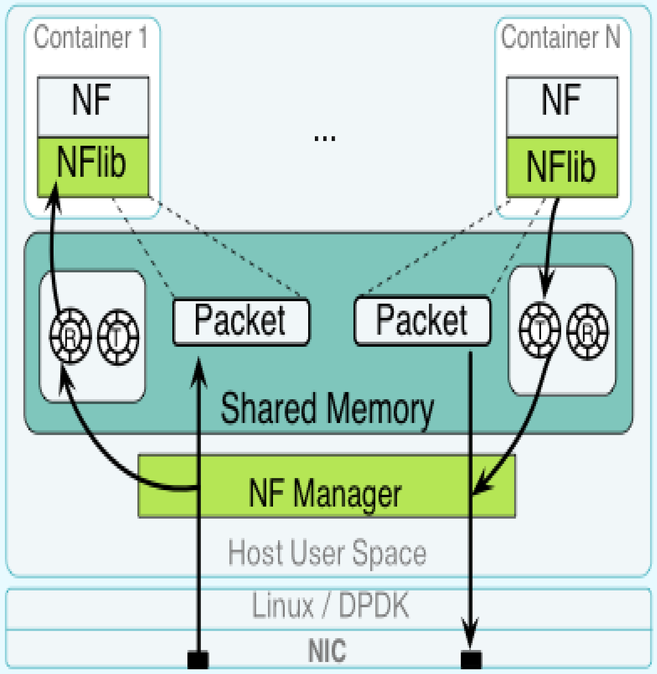
Container based NFs: Ease of user-space process management brought to networking

NF Manager: Orchestrates traffic flow between various NFs to bring elasticity

Zero-Copy IO: Packets DMA'd into shared memory granting NFs direct access to data without copies

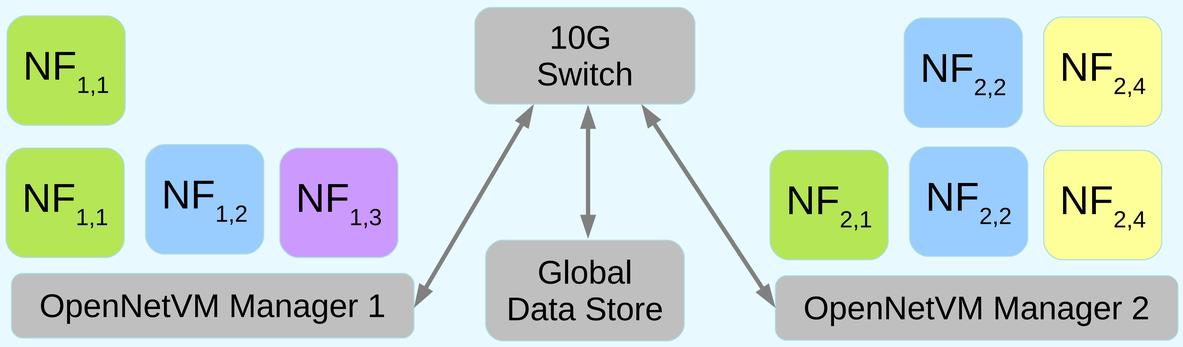
NUMA-Aware: Maximizes performance by ensuring data in memory is local to a thread's CPU Socket

Interrupt-Free: DPDK's poll mode driver allows non-traditional network to process incoming traffic at **10Gbps and beyond**



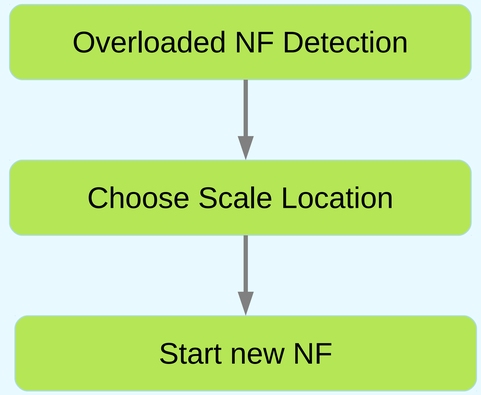
Distributed openNetVM Manager

- As networks grow, more middle boxes need to be deployed to scale efficiently
- openNetVM has a dynamic manager which makes networks elastic
  - Aware of all active and newly created NFs
  - Re organizes global data structures upon NF creation and destruction
- The openNetVM manager can route traffic to the middle box instance with most available processing capacity



Auto-Scaling NFs

- Single NFs can be limited by hardware resources available on the host server
- The openNetVM manager can detect when an NF is operating at its performance limits
- Automatically start additional instance to handle packet load



- Need to track which hosts have available CPU resources to potentially start new NFs
- When needed, find a host that already has the desired NF service running (in order to maintain locality of packet flows)
- Choose the host that has the most available resources (to properly load balance computation across the cluster)
- Enqueue a "scale request" into a distributed queue. Each openNetVM manager instance polls for incoming messages and handles starting new NF instances on that machine
- Overall, prefer to start additional instances "locally" to avoid network overhead on critical packet flow paths