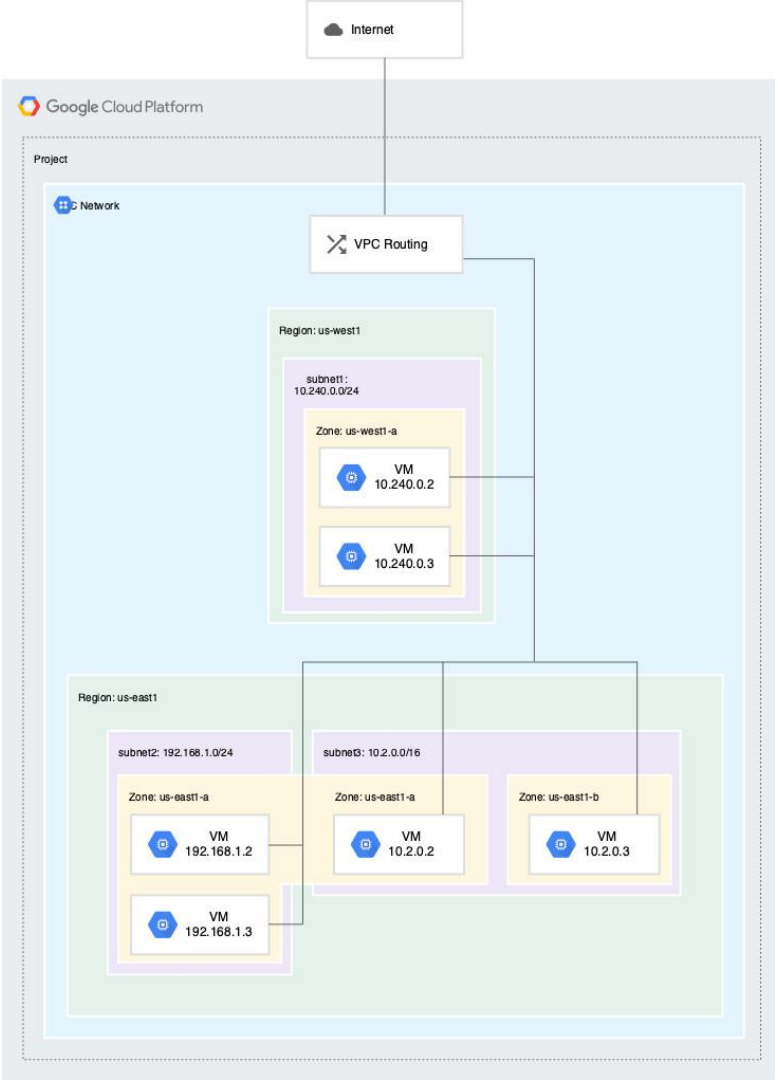
Cloud-scale Virtual Networking

Parveen Patel
(Google)

Cloud Networking from Outside

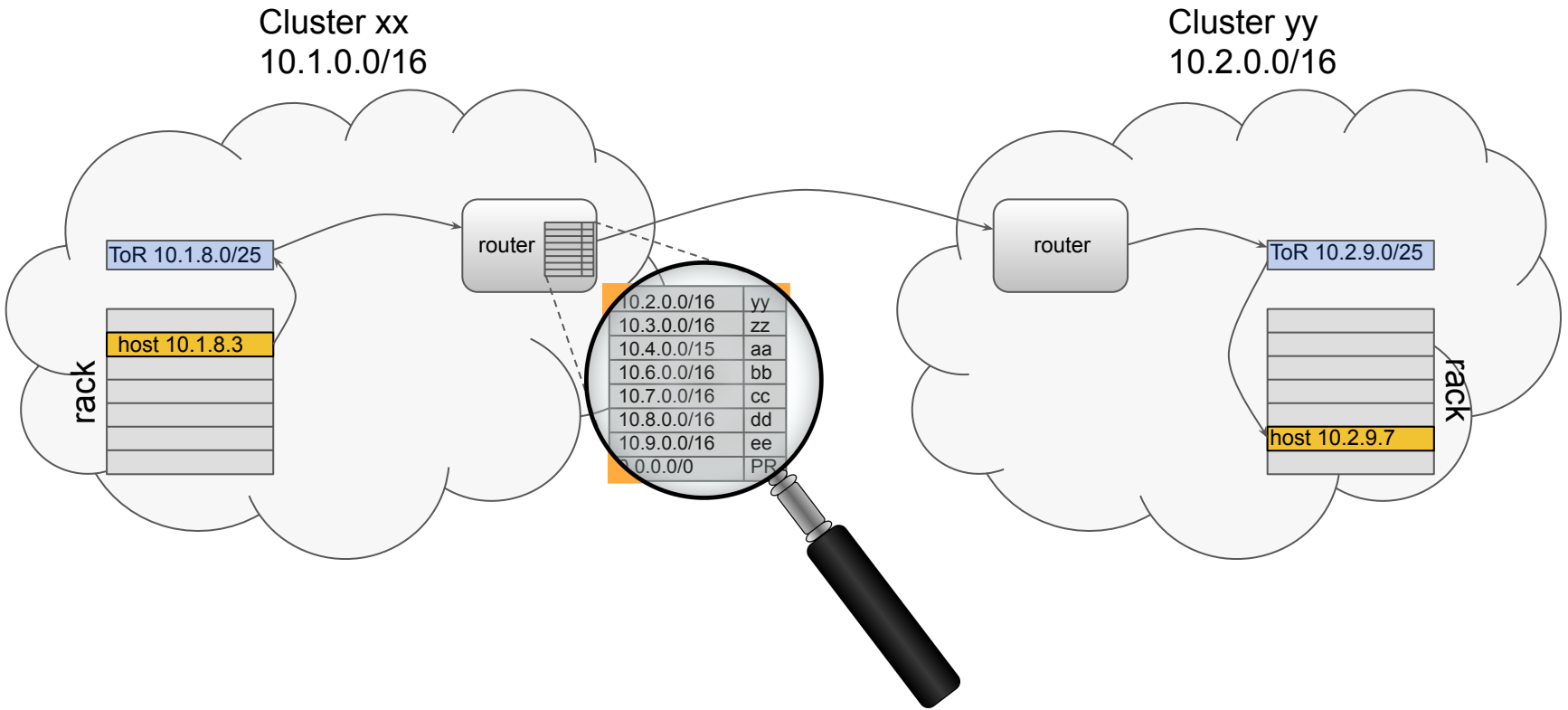
- VPC - Virtual Private Cloud
  - Typically refers to "virtual private network"
- Regional subnetworks within the VPC
  - Global connectivity
- VMs within the subnetworks
- "Private" IPs within GCP
  - "Public" IPs to reach internet



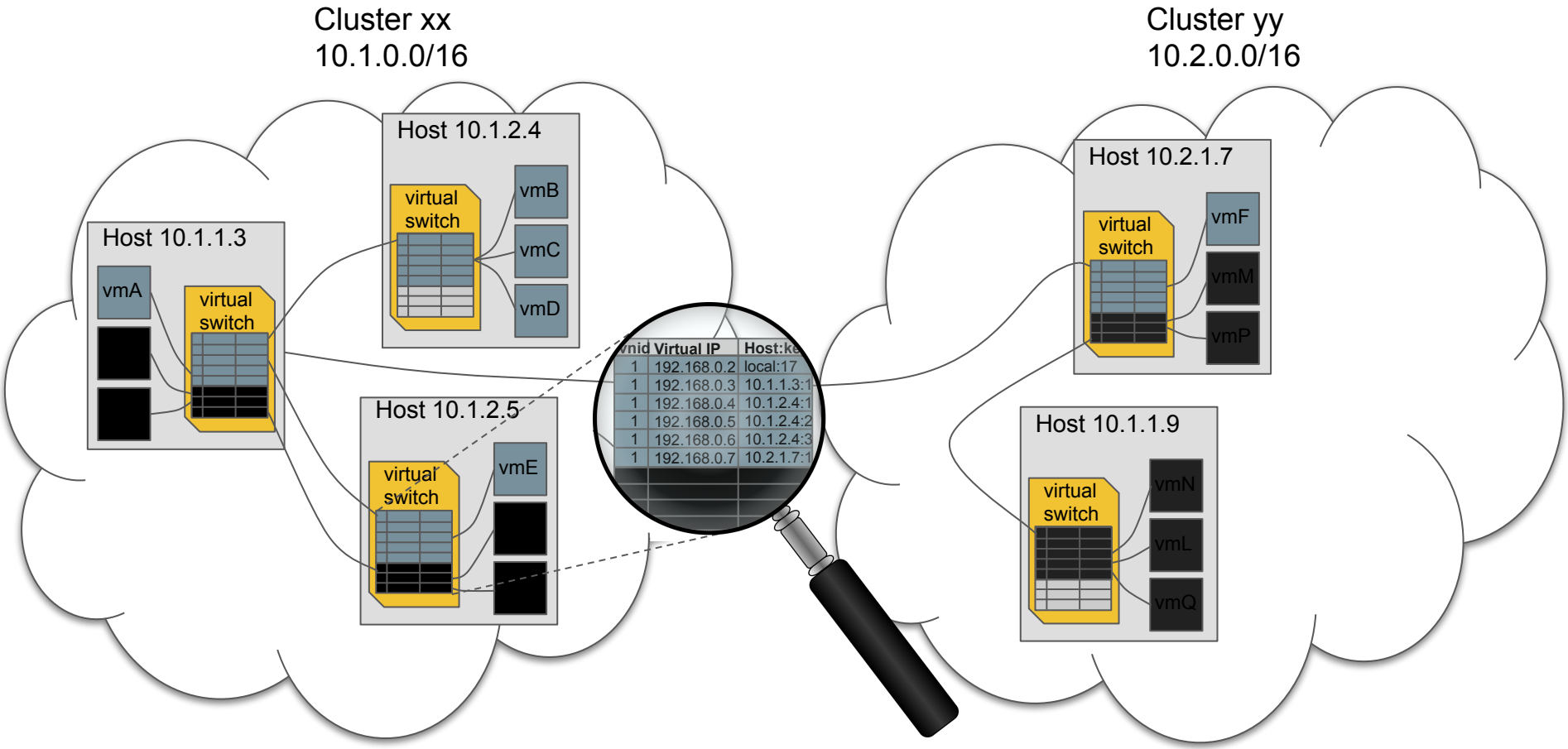
Key Features

- Network Virtualization
- Traffic shaping & steering (BwE, Espresso)
- Inbound & outbound DoS detection and throttling
- Network ACLs
- Packet Encryption
- NAT
- Stateful Firewall
- Load balancing (L3/L7, internal/external)
- Routing; connectivity to VPN, Interconnect
- VPC Peering
- …

Routing in Physical Networks



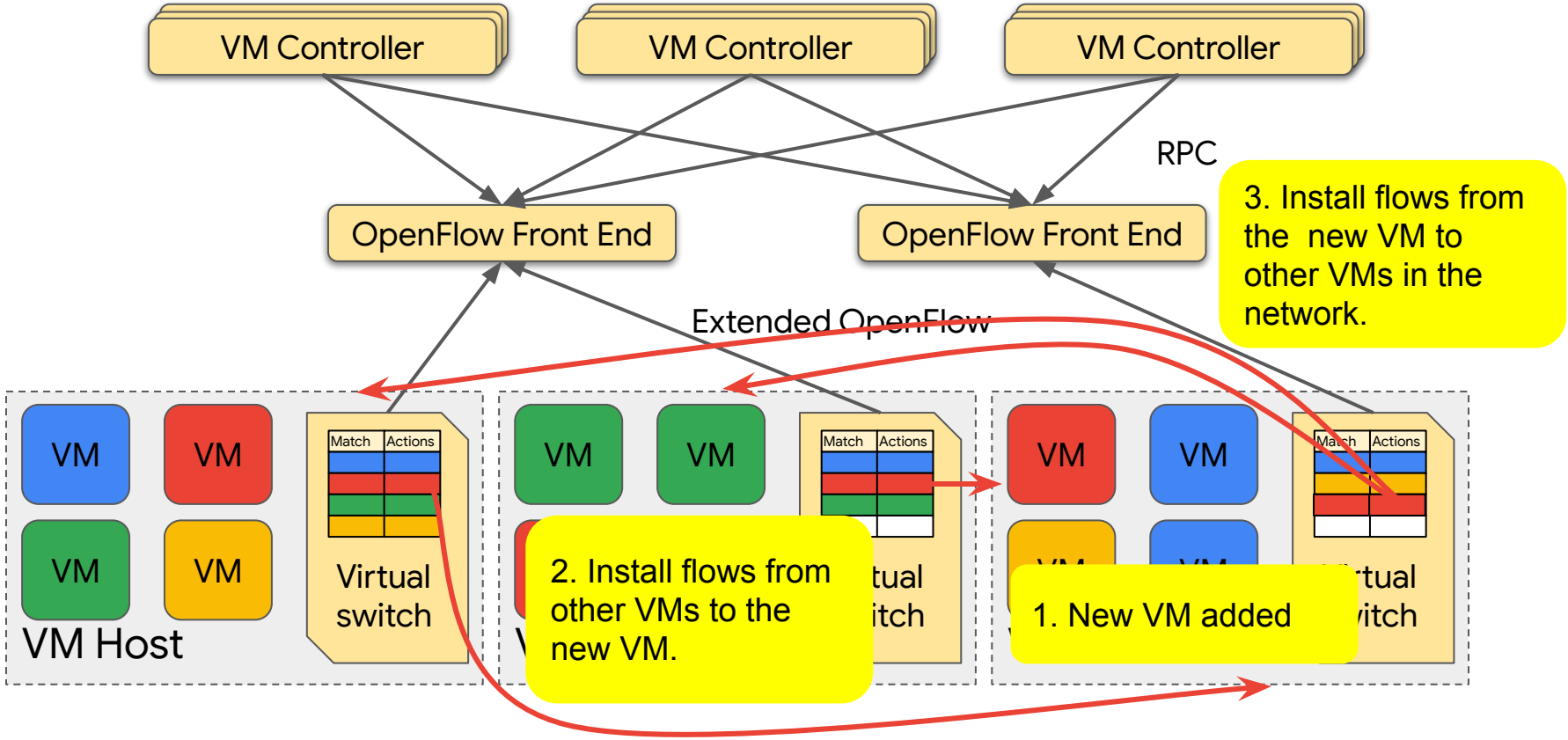
Routing in Virtual Networks at Google



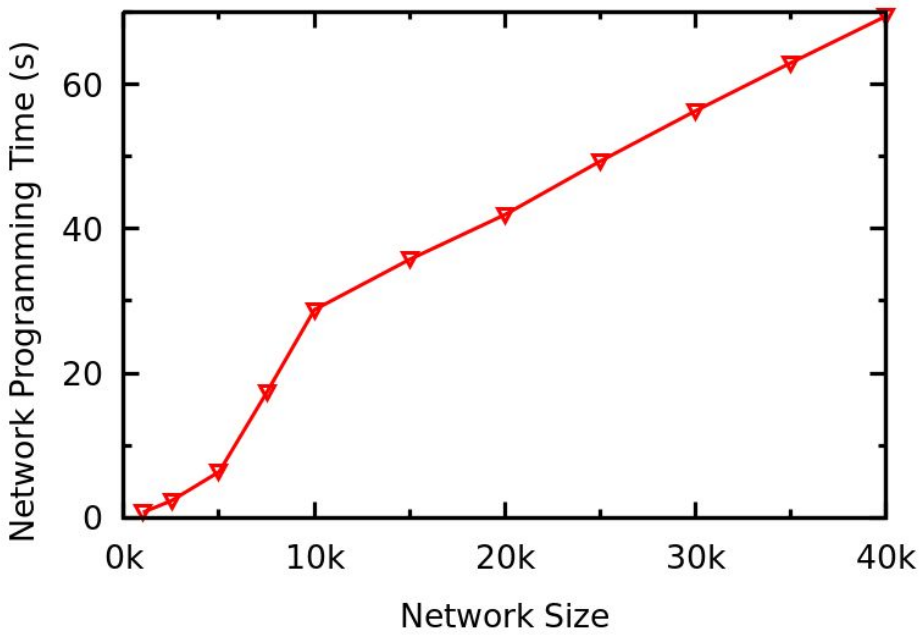
Google Andromeda

Built on the Snap framework: https://research.google/pubs/pub48630/

Andromeda Architecture



Programming Time for Large Networks



Setup:

- VMs are placed on 10,000 hosts
- 30 VM Controller shards

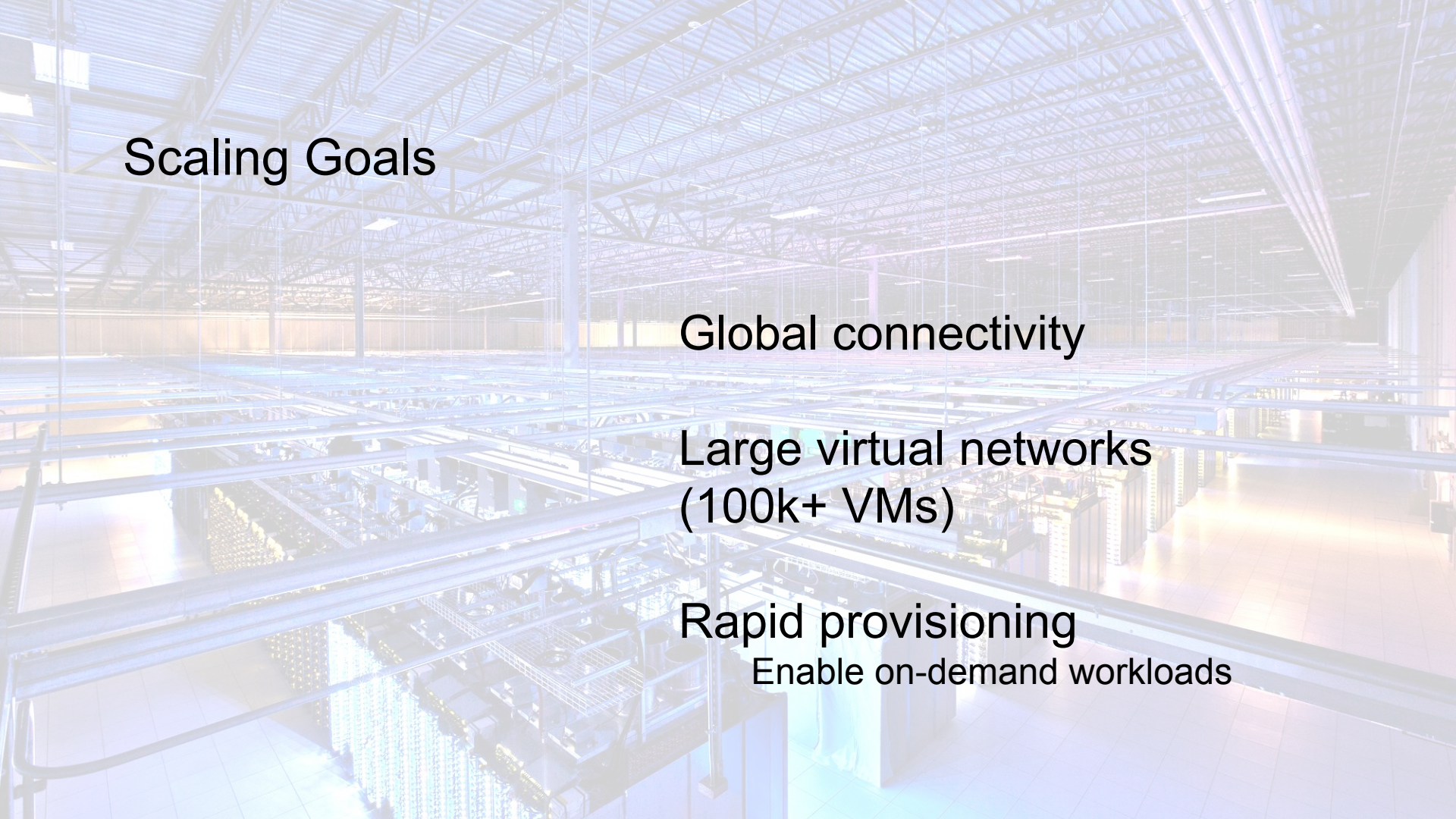
Programming time is $O(n \times H)$

n = number of VMs

H = number of hosts

Quadratic scaling leads to provisioning challenges

- Control plane CPU and memory
- Dataplane memory



Scaling Goals

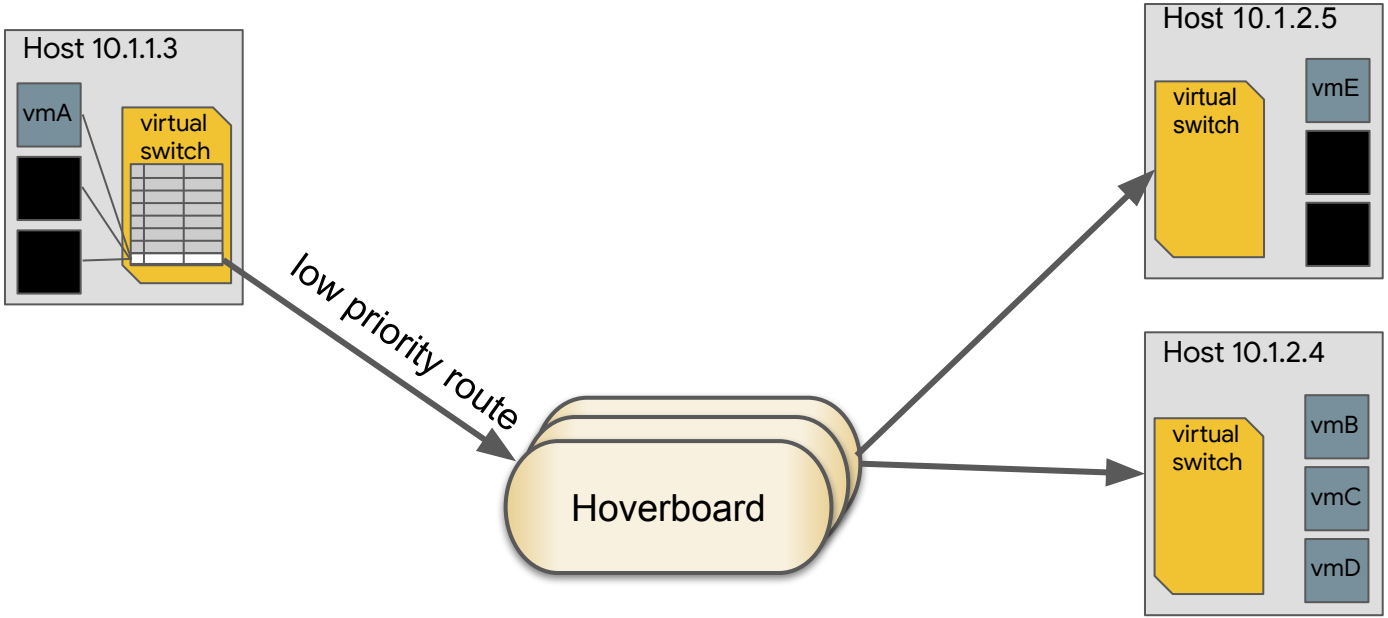
Global connectivity

Large virtual networks (100k+ VMs)

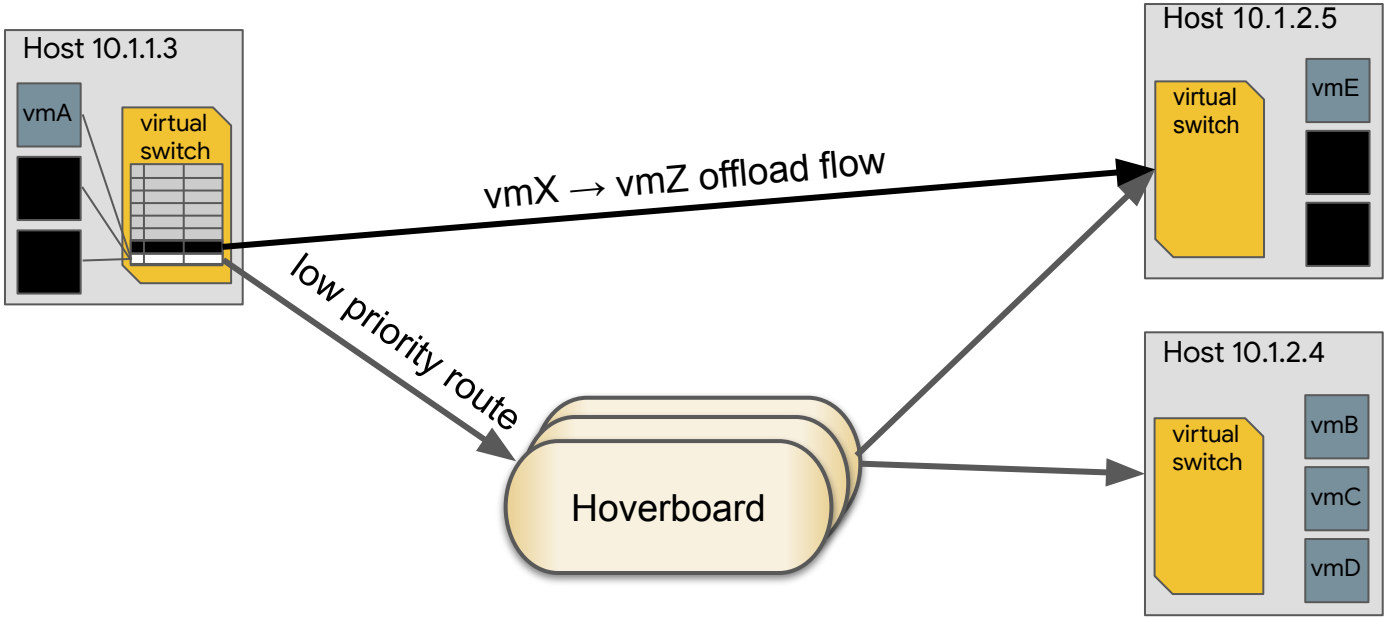
Rapid provisioning

Enable on-demand workloads

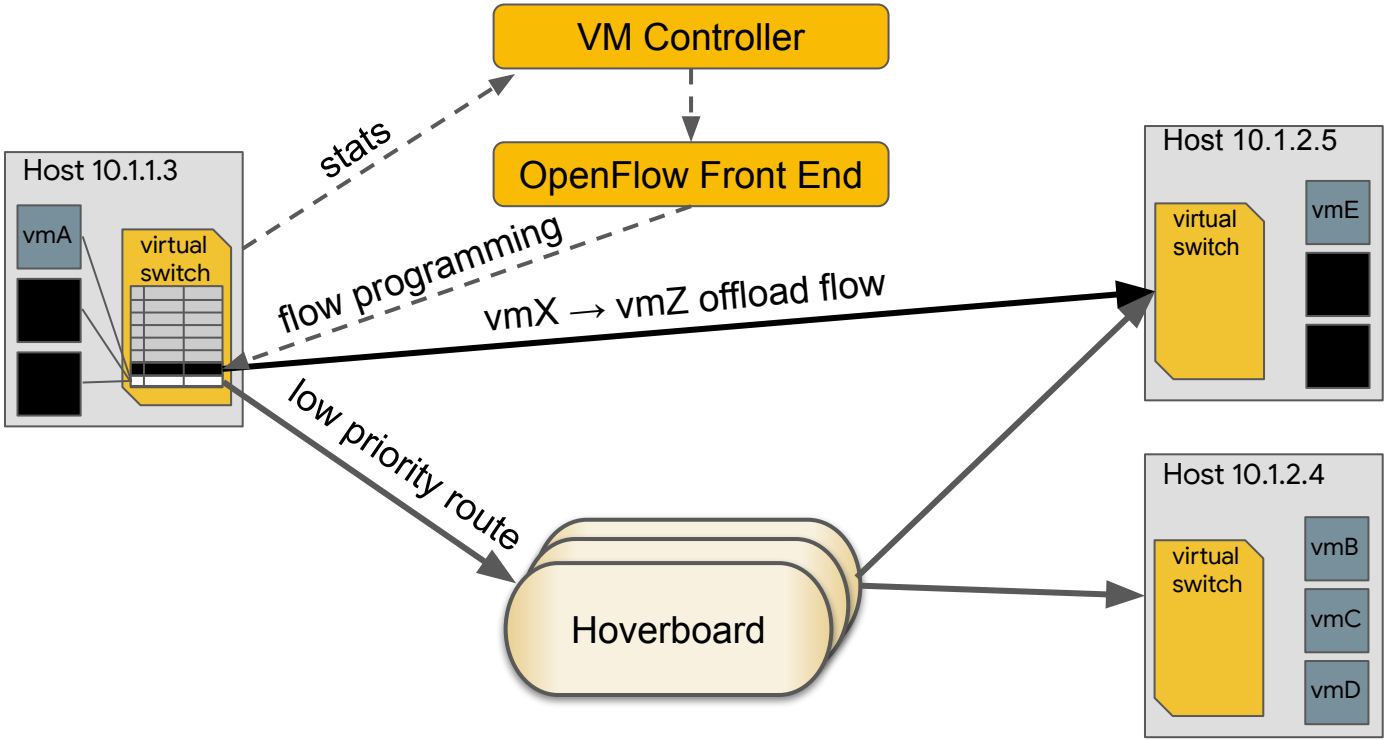
Scaling with Hoverboards



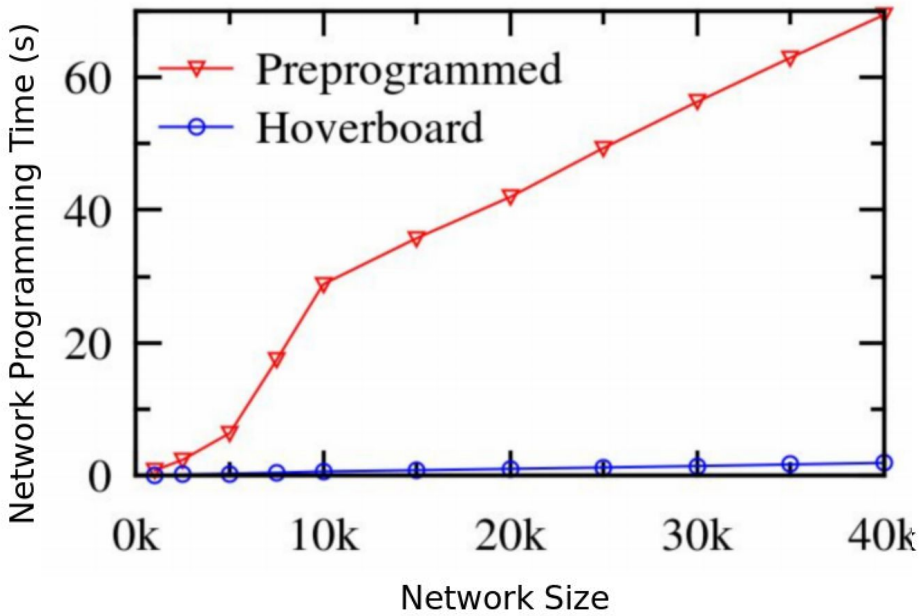
Hoverboard Offloading



Hoverboard Offloading



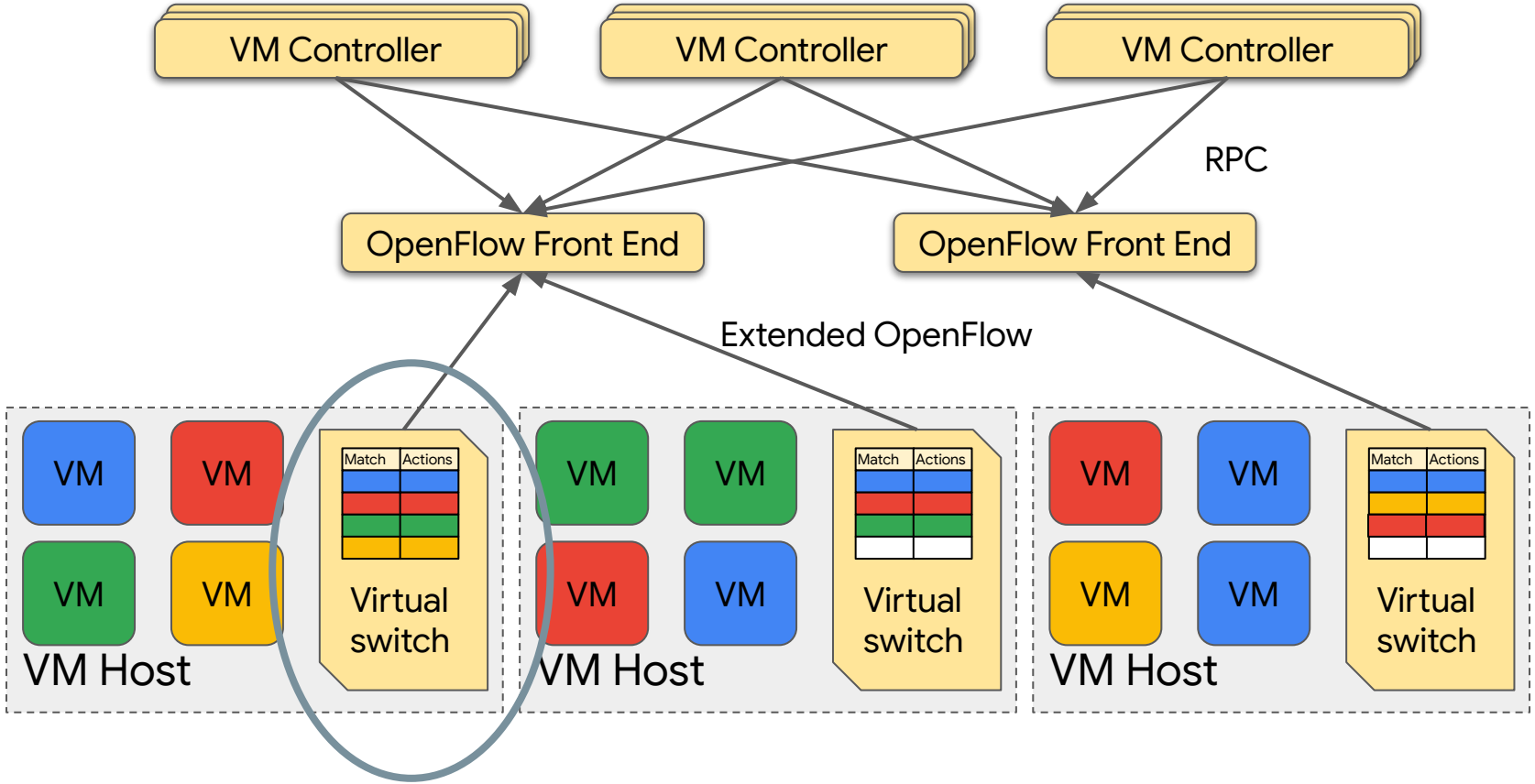
Programming Time with Hoverboard



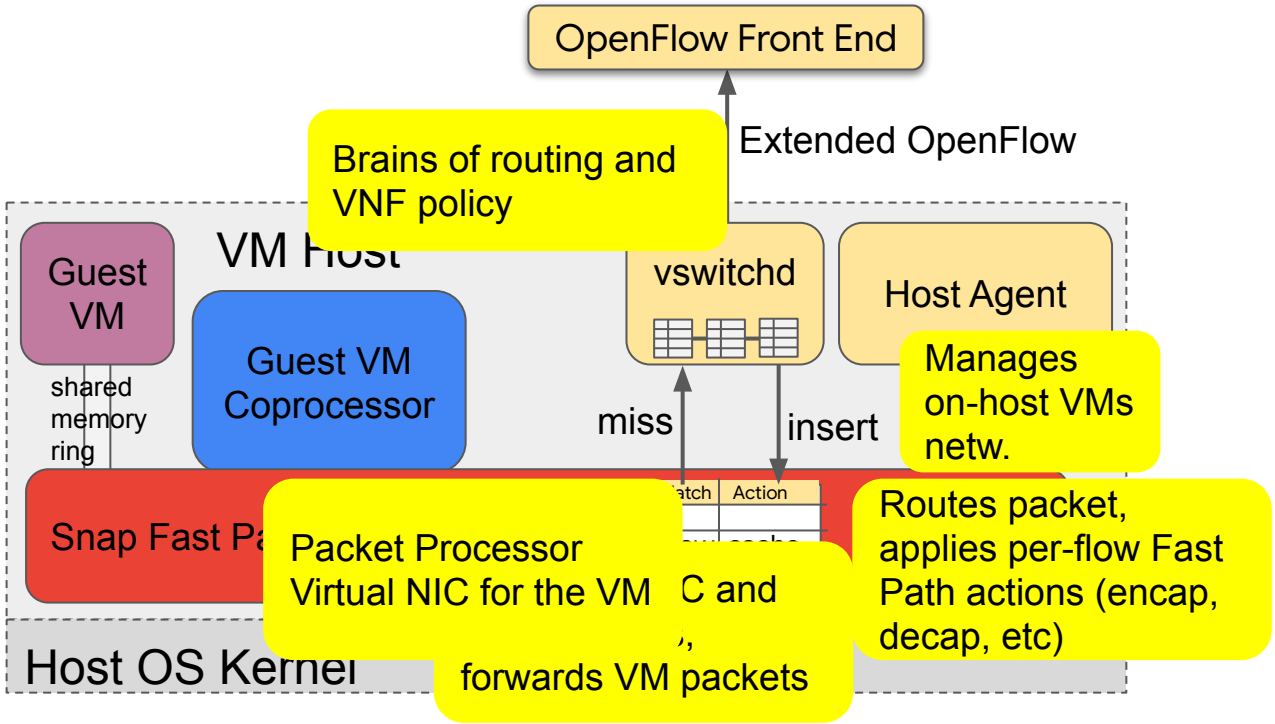
Programming linear with # VMs only

On-host Components

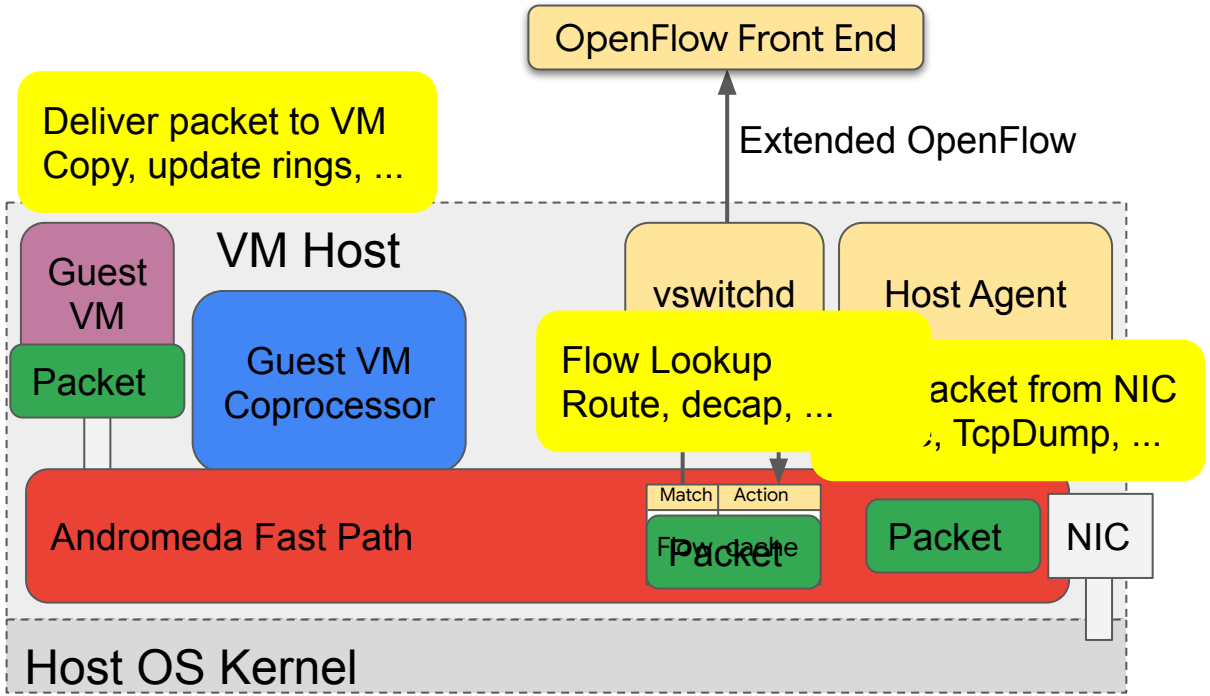
Andromeda Architecture



Deep Dive: On-Host Stack (Snap, host agent, vswitchd)



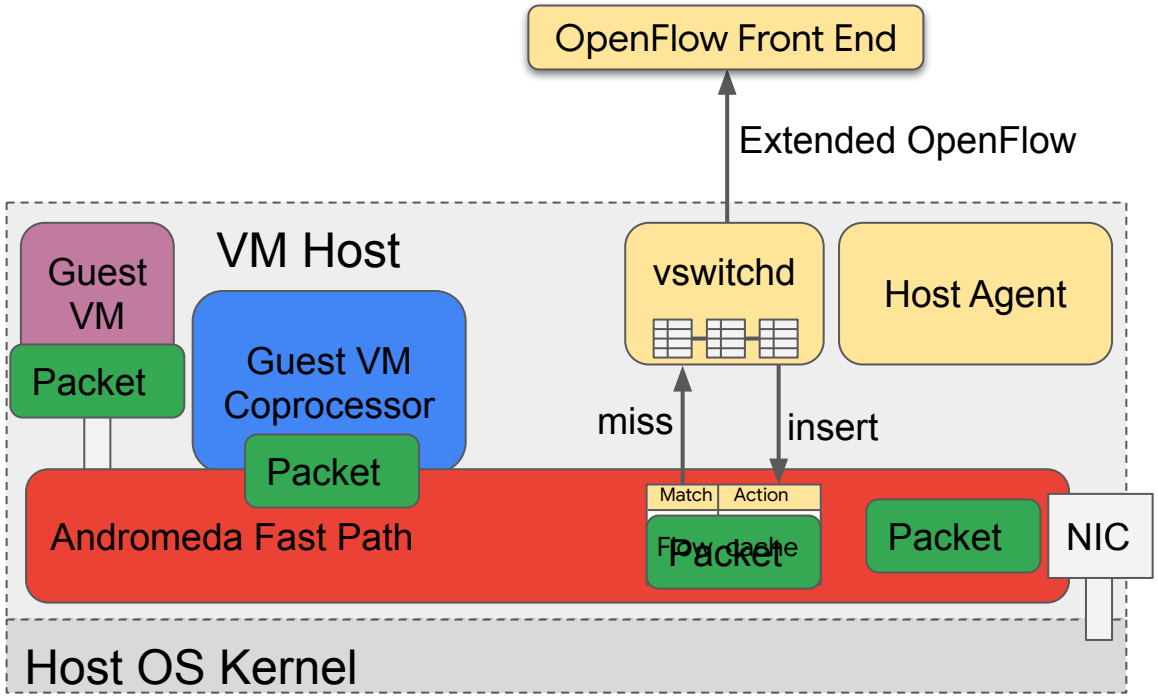
Deep Dive: On-Host Stack (Snap, host agent, vswitchd)



High performance traffic processed end-to-end on **Fast Path**

Flow Table performs routing, encap/decap, etc.

Deep Dive: On-Host Stack (Snap, host agent, vswitchd)



Coprocessors are per-VM threads CPU attributed to VM container

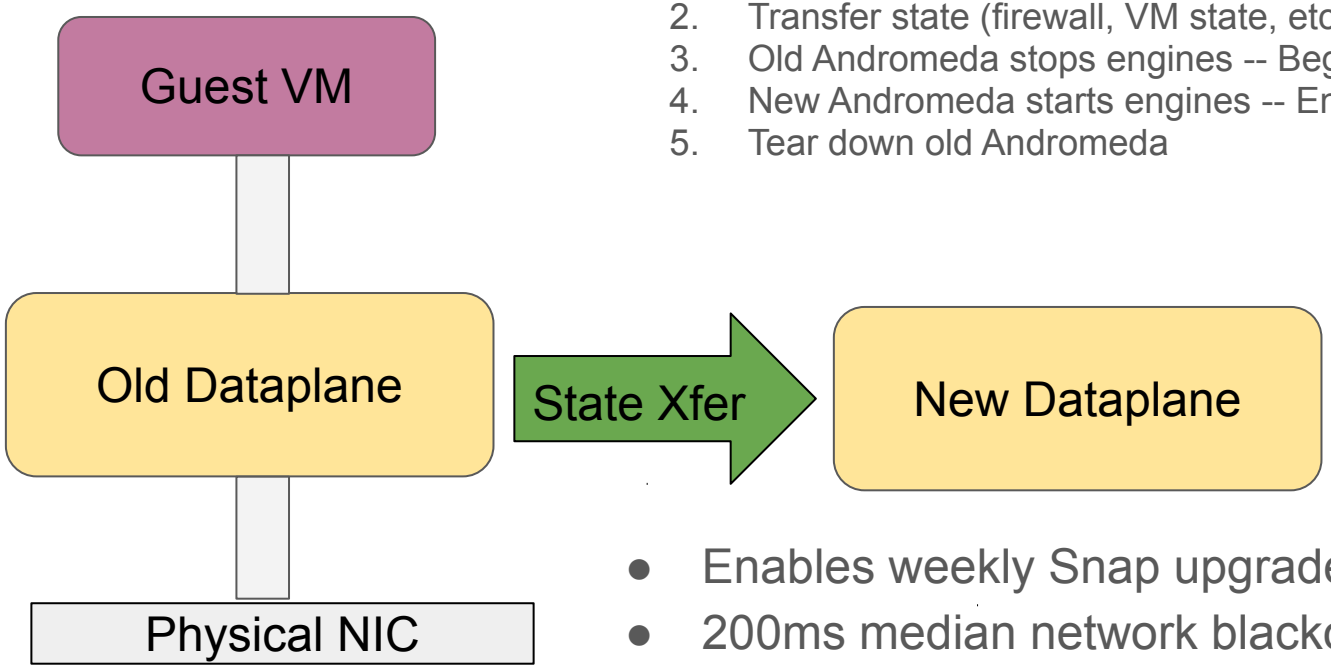
Coprocessors execute CPU-intensive packet ops such as LPM/NAT

Decouples feature growth from Fast Path speed

Velocity through Transparent Maintenance

Andromeda Hitless Upgrade

1. Bring up new Dataplane, create packet buffers, NIC queues
2. Transfer state (firewall, VM state, etc)
3. Old Andromeda stops engines -- Begin blackout
4. New Andromeda starts engines -- End blackout
5. Tear down old Andromeda



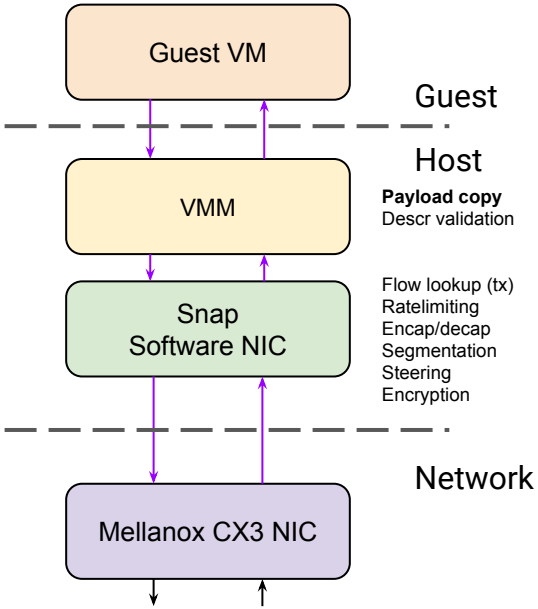
- Enables weekly Snap upgrades
- 200ms median network blackout (1s @ P99)

Offloads and performance

Snap/Andromeda 2.0 (cir. 2016)

- VMM / hypervisor
  - Implements all aspects of guest network interaction
- Snap/Andromeda is a userspace C++ binary
  - "Spins" on 1 host core
  - Owns a couple OS-bypass NIC queues
  - Implements packet virtualization

⇛ 16Gbps on a 2x10G NIC

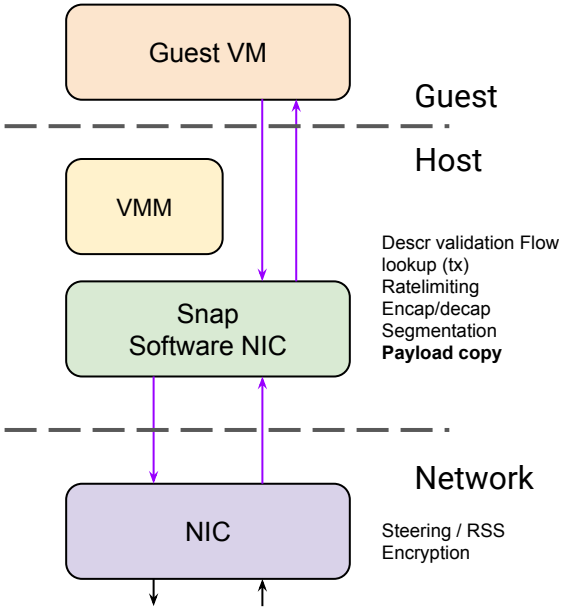


Snap/Andromeda 2.1 (cir. 2019)

- Bypass VMM for dataplane(*)
  - VMM still implements control aspects of "virtual device"
  - (*) fallback to older mode on VM migration
- Snap slowly taking advantage of NIC offloads
  - HW RSS allows Snap to "scale up" to 2 (or more) host cores
  - Rx Steering allows per-VM queues & major isolation win
  - Encryption offloaded

⇒ 32Gbps GA in Q2'19

⇒ 100G for GPUs w/ more Snap cores

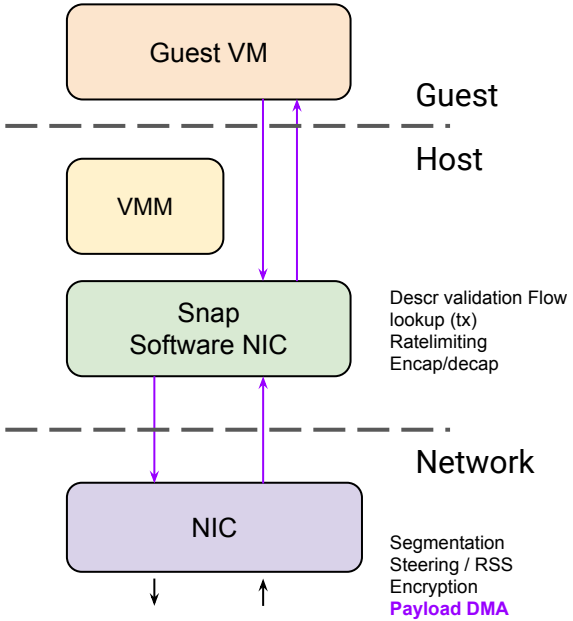


Snap/Andromeda 2.2 (Q1 2021)

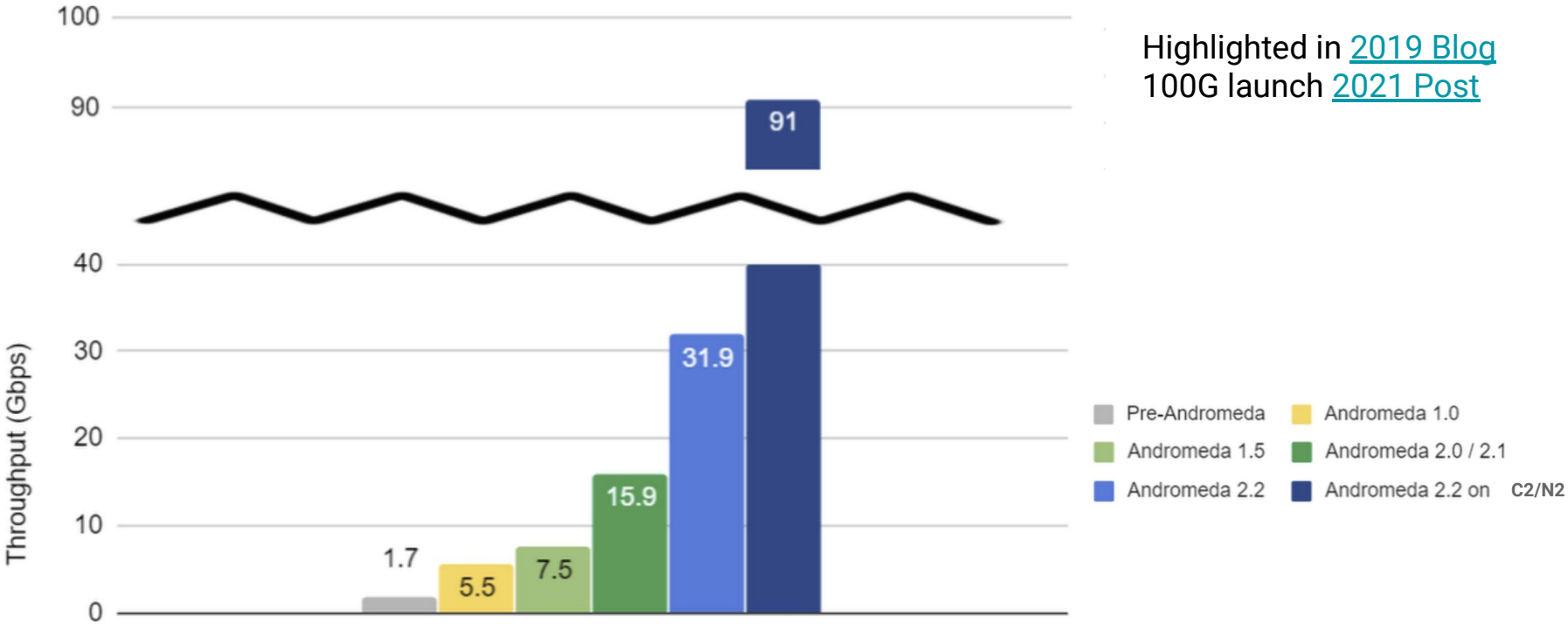
- Main goal to offload payload copies to hardware
  - With VMM Networking team, built "gVNIC" guest driver
  - Tx DMA offloaded to Google NIC with gVNIC
  - Rx copies offloaded to Intel DMA Engines
- CPU usage more efficient
  - Now spin on only 1 host core
  - Dynamically scale out to 3 more hyperthreads as needed

  ⇒ 100G VM-to-VM Preview in Q1'21

- Ideally NIC would enable "full bypass" of Snap
  - In practice ***HW bugs*** and ***changing requirements*** prevent that



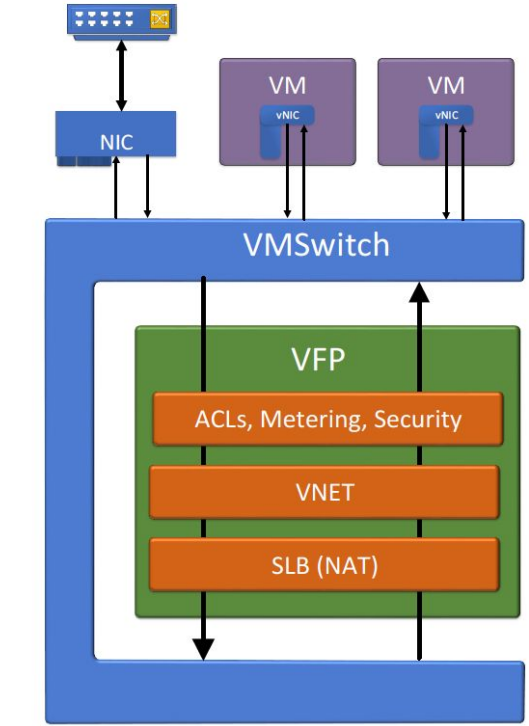
Achieved TCP Throughput for GCP VMs



Highlighted in 2019 Blog
100G launch 2021 Post

Azure VFP

Virtual Filtering Platform (VFP)
Azure's SDN Dataplane

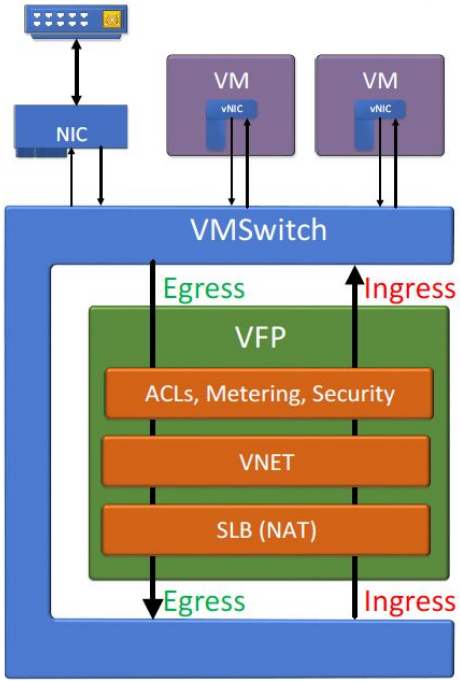


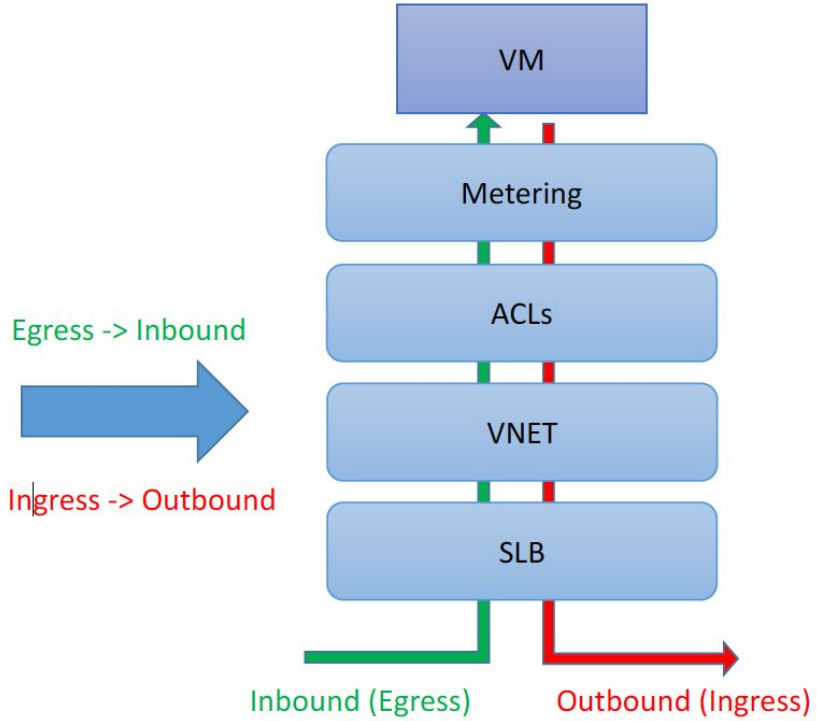
Microsoft Azure

- Plugin module for WS2012+ VMSwitch
- Provides core SDN functionality for Azure networking services, including:
  - Address Virtualization for VNET
  - VIP -> DIP Translation for SLB
  - ACLs, Metering, and Security Guards
- Uses programmable rule/flow tables to perform per-packet actions
- Programmed by multiple Azure SDN controllers, supports all dataplane policy at line rate with offloads

Source: https://www.usenix.org/sites/default/files/conference/protected-files/nsdi17_slides_firestone.pdf

VFP Translates L2 extensibility (ingress/egress to switch) to L3 extensibility (inbound/outbound to VM)

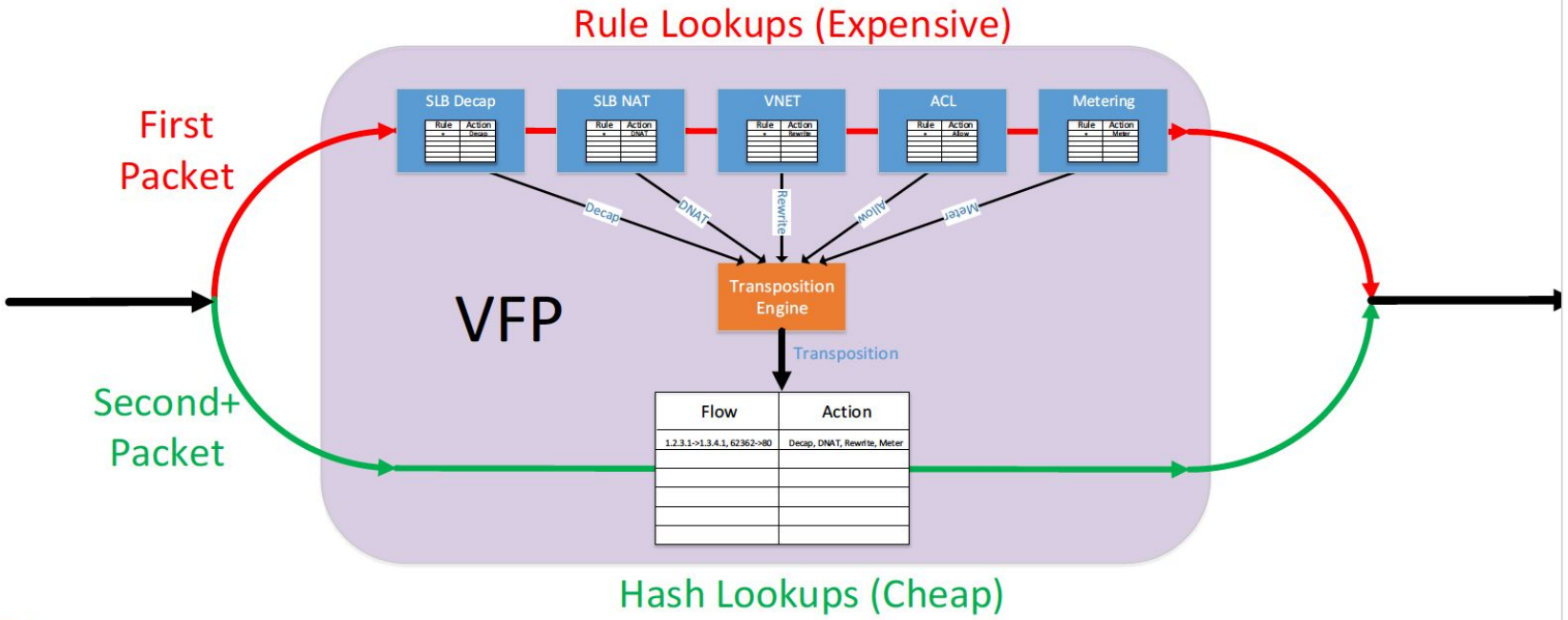




Microsoft Azure

Source: https://www.usenix.org/sites/default/files/conference/protected-files/nsdi17_slides_firestone.pdf

Unified Flow Tables – A Fastpath Through VFP



Microsoft Azure

Source: https://www.usenix.org/sites/default/files/conference/protected-files/nsdi17_slides_firestone.pdf

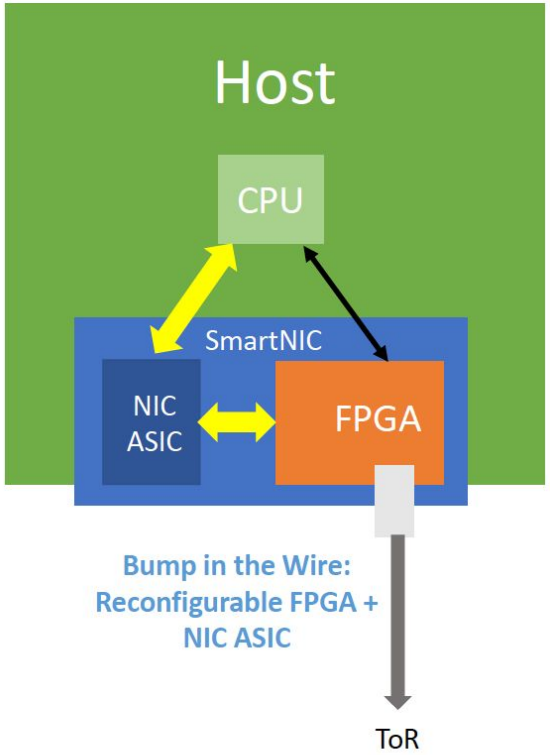
On-host SDN datapaths work great, however

- Jitter due to software not acceptable
  - Some workloads demand predictable latency and bandwidth
- Do not support virtual networking for Baremetal machines
  - Bring your own hypervisor
- Expensive to tune software to each hardware
  - Heterogeneity of the host architecture - Intel, AMD, ARM

SmartNICs

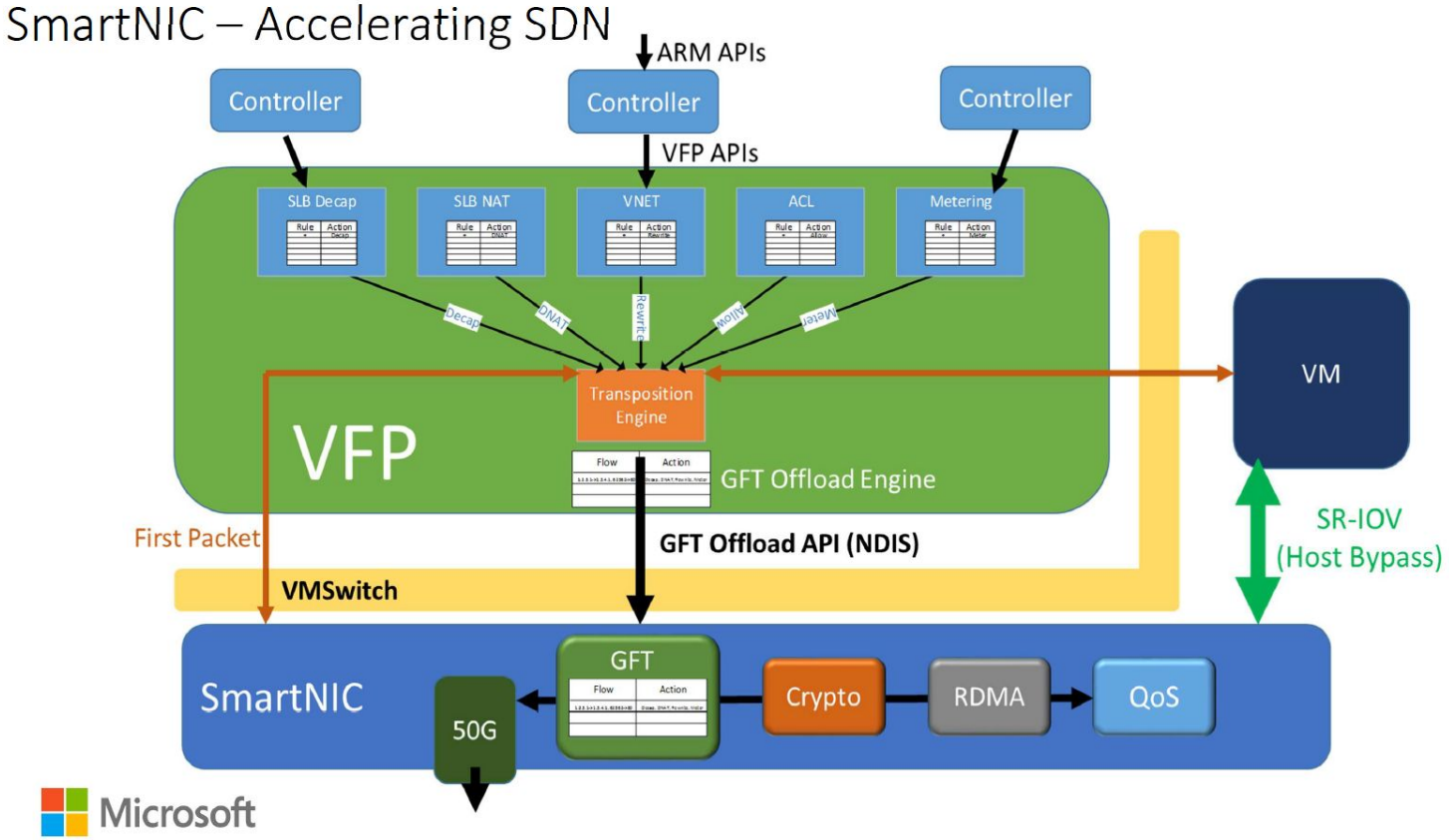
Azure SmartNIC (FPGA)

- HW is needed for scale, perf, and COGS at 40G+
- 12-18 month ASIC cycle + time to roll new HW is too slow
- To compete and react to new needs, we need agility – SDN
- Programmed using Generic Flow Tables
  - Language for programming SDN to hardware
  - Uses connections and structured actions as primitives



Microsoft

Source: https://www.usenix.org/sites/default/files/conference/protected-files/nsdi18_slides_firestone.pdf

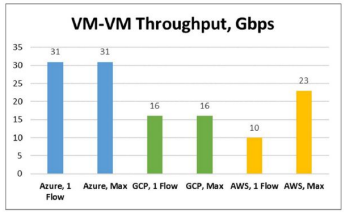


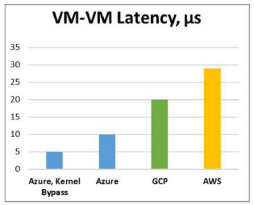
Source: https://www.usenix.org/sites/default/files/conference/protected-files/nsdi18_slides_firestone.pdf

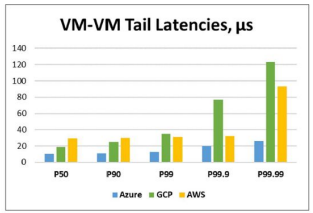
2018

2021

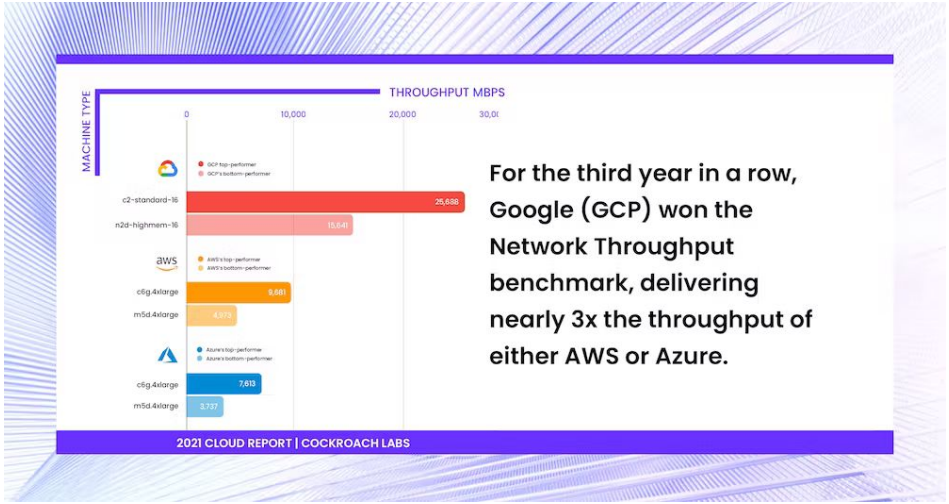
AccelNet Comparative Results







Microsoft



Source: https://www.usenix.org/sites/default/files/conference/protected-files/nsdi18_slides_firestone.pdf

Source: https://www.cockroachlabs.com/guides/2021-cloud-report/

Common themes

- Control plane
  - Hierarchical Controllers
  - High feature velocity
  - OpenFlow too inflexible for cloud scale
- Data plane
  - Match-action tables
  - Fastpath, slowpath separation
  - Hitless upgrades
  - Offloads

Future Directions

Challenges/opportunities

- Keeping up with speeds and feeds (Tbps networking)
  - Performance versus programmability
- Dataplane portability across multiple SmartNICs
  - Is there a flexible enough language to express offloads
- Container networking offloads
  - Can we enable networking for a container in milliseconds
- High-performance networking workloads
  - Multicast, RDMA, time synchronization etc
- Explosion of heterogeneity
  - CPU arch, VM types, workload types, NIC types, GPUs, Baremetal, multi-NIC
- How to maintain reliability and velocity in this complex space?

Questions (½)

Andromeda

- Can you speak to the **scheduling aspect of running 10s of thousands of VMs at scale** in a data center? None of the papers so far have emphasized this as much as I would have expected, but this seems to be a critical and nontrivial piece. – Borg.
- I believe this paper is something that came out of industry (Google). I think this allowed the authors to build Andromeda out over five years, and discuss experiences and challenges of the process informed by a half-decade's work. I think it is less likely that academic research environments provides students with the ability to build out a project over such a long time, so I'm curious if the scope of academic research in networking is generally smaller? Are different applications and problems tackled in academia than in industry so as to not necessitate 5 years of evolution, evaluation, and deployment? – Focus on the primitives, Collaborate with the industry
- If the operators want to offload more functionalities to hardware, do they need to redesign/change the hardware? - Depends
- Programmable switches like RMT support SDN and virtualization *within the network*, but they do not solve the issues related to partitioning a host's networking resources to support this virtualization. RMT and AccelNet solve different but related problems. What are the possible co-optimizations that could be leveraged to improve a cloud's virtualization system as a whole? - RMT not flexible enough to support all use-cases.
- Do Google and Microsoft still use Andromeda and Accelnet, respectively, or has one of the systems won out as the clearly better option? - no single best solution, it's a trade-off.
- As more archetypal flow types start to emerge (beyond basic attributes of "CPU-intensive" or "mostly idle" to more application-specific attributes), would the authors support the development of additional types of data paths to specialize for each kind of flow, given that this was a key benefit of their approach? Is there a cost to subdividing these flows too much and specializing to the needs of each kind of flow, or is that the future of network virtualization? - RDMA is a good example, RPC offloads?
- The Azure paper stresses how expensive it is to use physical cores for host networking. How does Andromeda deal with this issue as they seem to still use CPU cores for all packet processing? - efficient software
- In lecture, it was mentioned recently that datacenter networks tend to be "brittle", where admins arrive at an optimal topology through a complex process that sometimes involves trial and error, and any modification tends to be very complex, with high operational cost. Does the Andromeda architecture's Hoverboard mechanism help remedy this issue, at least partly? - Physical networks vs virtual networks
- This question is more for my curiosity about the broader field of computer networking but -- a lot of papers we read this quarter seems to cover systems that are very easy to identify (e.g., Azure is clearly Microsoft platform likely run by those in that institution). How does the anonymization process work for reviewing these papers? - industry track does not require anonymization
- Also, how generalizable are the findings that are aimed at designing large-scale networking systems (presumably, they do not get built all that often)? - benefits to adjacent spaces HW designers, storage, ML
- How could we reasonably compare the efficacy of Andromeda and AccelNet?

Questions (2/2)

AccelNet

- Have offloading optimizations like those above been incorporated for cloud GPUs yet, to your knowledge? It feels like they haven't yet given the relatively poor cost proposition of cloud GPUs vs. Cloud CPUs (for cloud infra vs. buying the infra directly).
- What are the limitations of using FPGA? Would it restrict the innovations on the functionalities that are offloaded to FPGA because the hardware design is fixed?
- Just a bit confused on the Hoverboard architecture.
- It seems like the authors are completely indifferent to the problem of wasting CPU cycles on the networking stack. This was the main motivation for the authors of the last paper. Why is there this difference in opinions between direct competitors? It seems like Microsoft got it right, since FPGAs probably have strictly better performance and allow match-action pipelining, unless I am missing something. If I am reading the experiment results right, AccelNet achieves ~5us ping between VMs whereas Andromeda achieves ~30.
- To what extent are cloud customers aware of these underlying network architecture differences among cloud vendors? To what extent have these different “bets” on network architecture manifested in concrete shifts in the competitive landscape of these top companies?
-