Disaggregating Stateful Network Functions

Deepak, Gerald, Rishabh, Michal, James, Silvano, Mario, Krishna, Arun, Arunkumar, Balakrishnan, Avijit, Sachin, Deven, Evan, Pranjal, Rishiraj, Neeraj, Soumya, Stewart, Ranveer, Srikanth

Microsoft Research

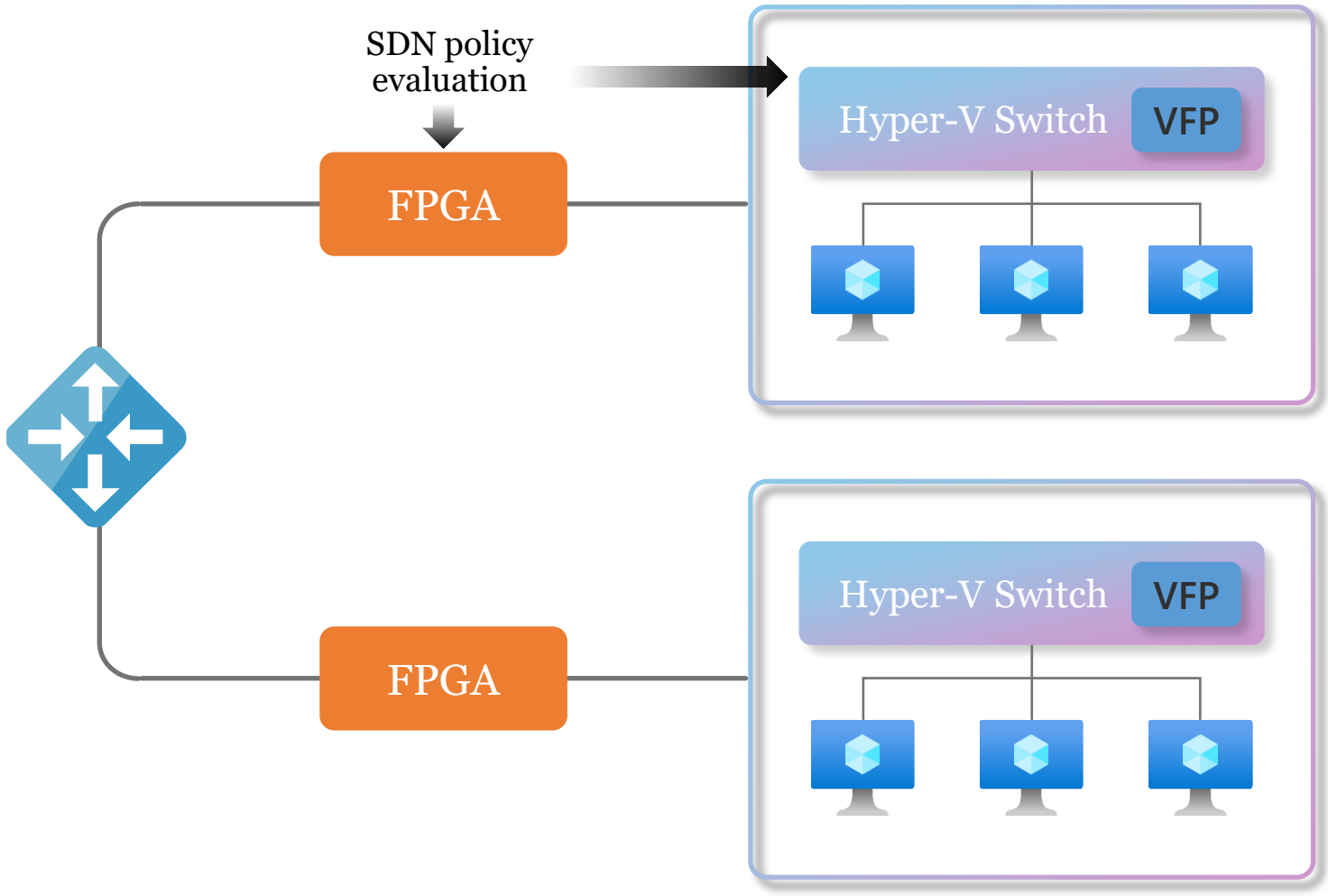
Azure SDN

Network policy processing today is performed on host node in virtual switch

---

Demanding customers use high number of SDN rules

SDN policy evaluation can be resource intensive when configurations are complex



Key motivation to Disaggregate the SDN

Enable full feature parity at high scale for **non VM workloads**

- Hardware storage appliances (NETAPP), Super computers (Cray), on-prem, hypothetical NICs corresponding to storage accounts.

Key motivation to Disaggregate the SDN

Enable full feature parity at high scale for **non VM workloads**

- Hardware storage appliances (NETAPP), Super computers (Cray), on-prem, hypothetical NICs corresponding to storage accounts.

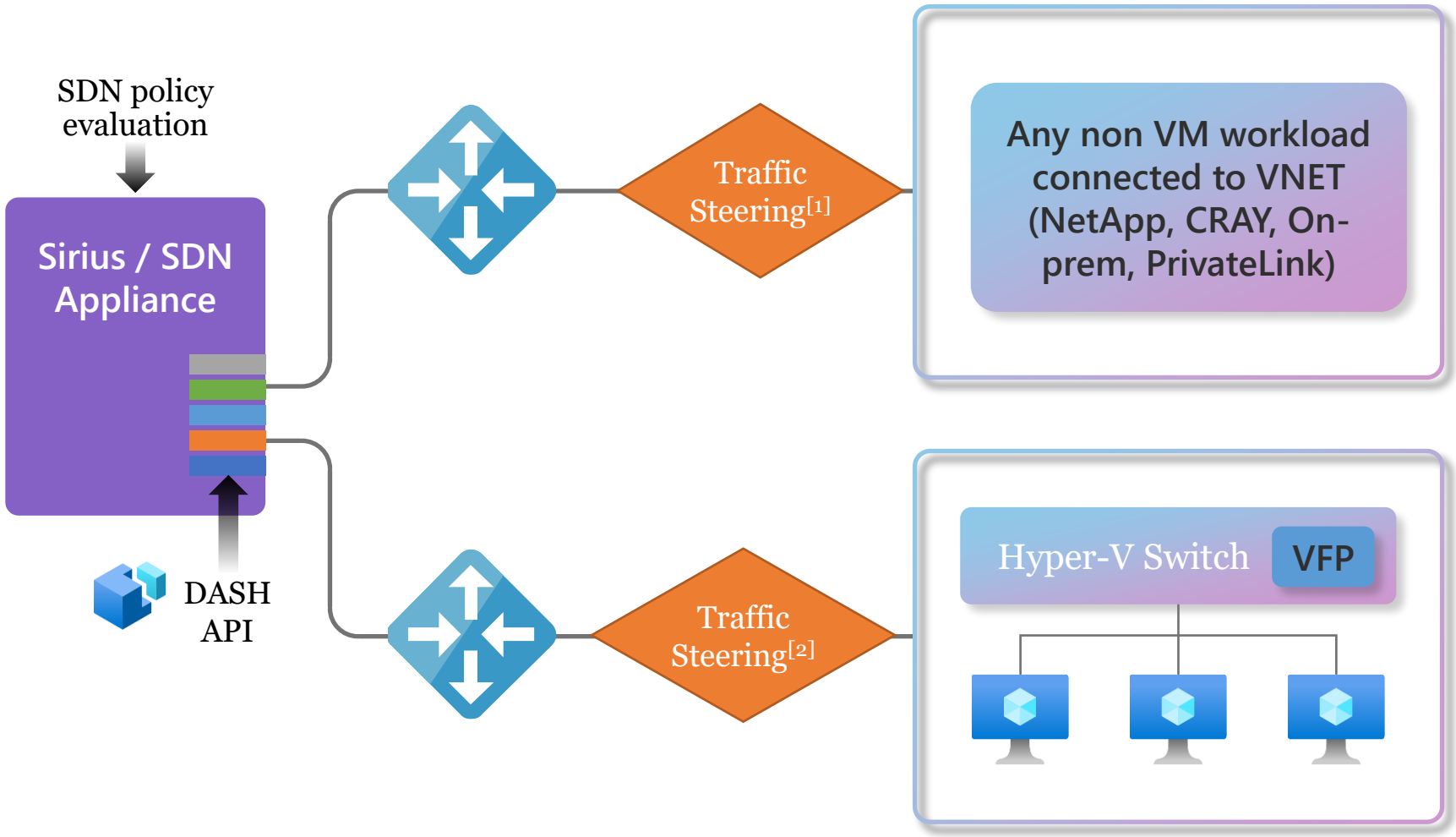
Enable **network optimized VMs** which require higher flow scale than can be supported on one host.

Azure SDN with Sirius (SDN appliances)

Offload SDN policy evaluation to disaggregated appliances

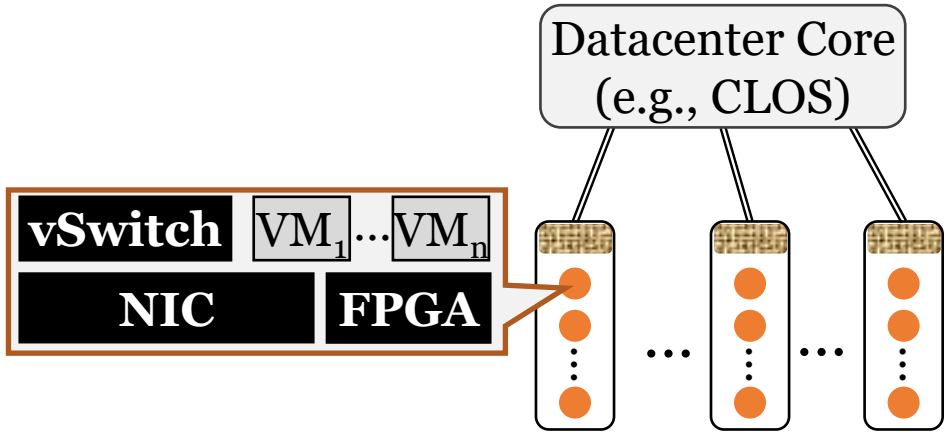
Greater agility, policy scale, flow scale and CPS

Enables rich scenario set beyond VM use cases.



[1] VXLan Tags at switch
[2] As above, with an FPGA match-action rule

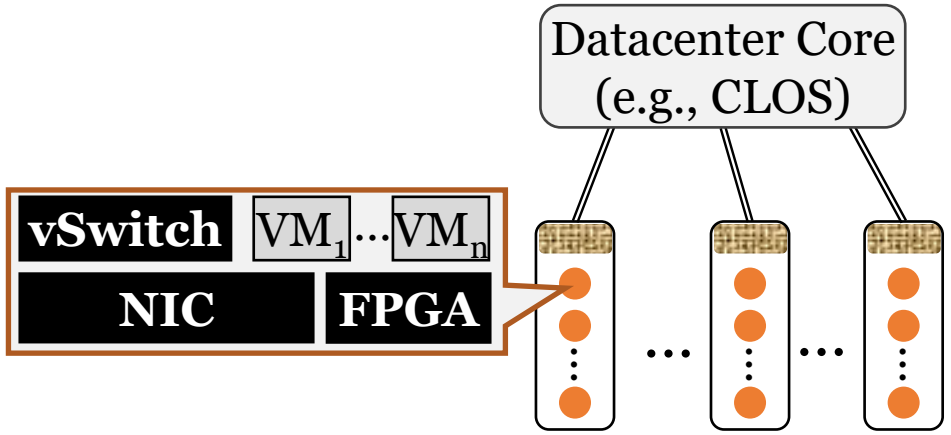
All Cloud Providers implement stateful NFs



All Cloud Providers implement stateful NFs

Example Stateful NFs:

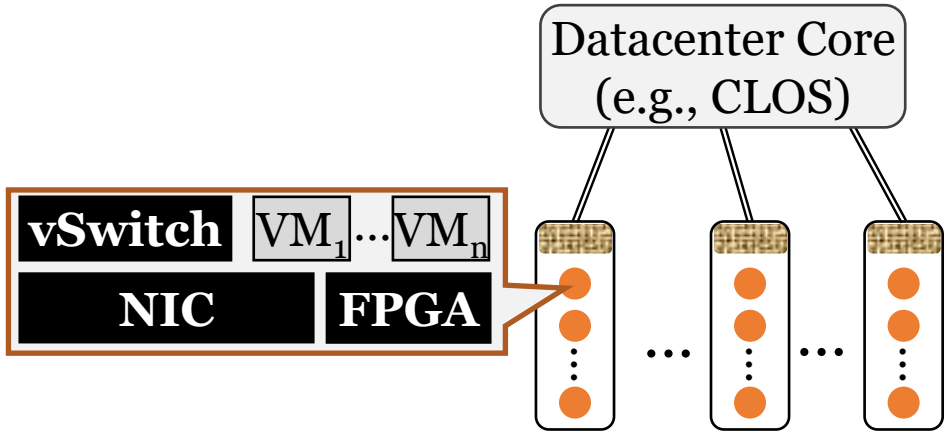
- Connection-tracking firewall
- Azure PrivateLink
- Stateful load balancers …



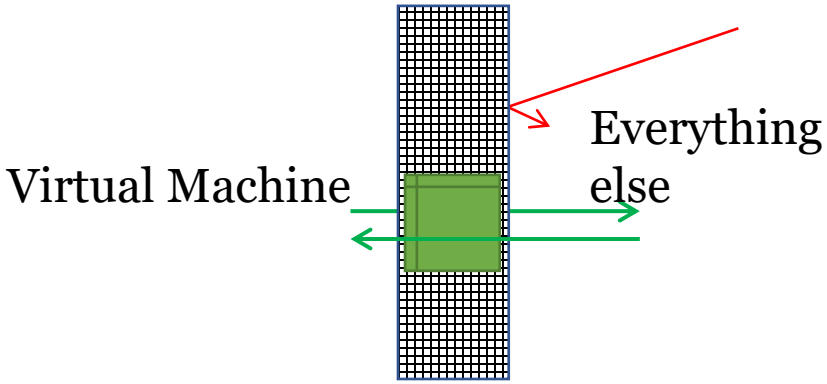
All Cloud Providers implement stateful NFs

Example Stateful NFs:

- Connection-tracking firewall
- Azure PrivateLink
- Stateful load balancers …



Connection-tracking firewall

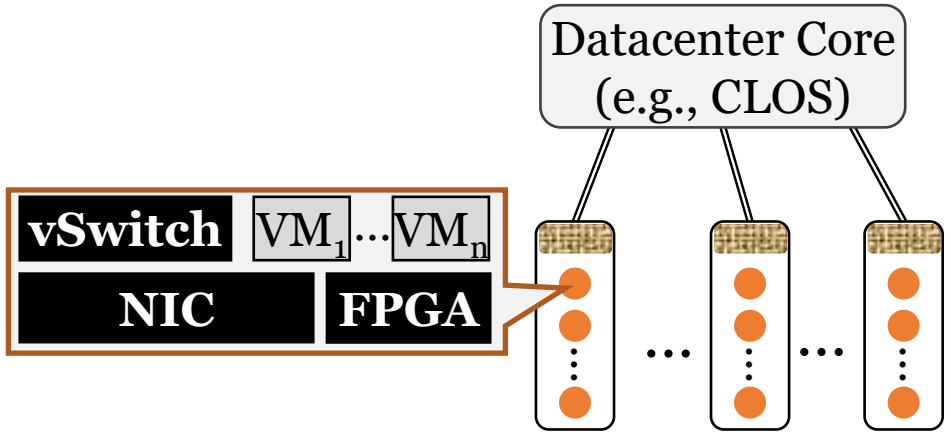


Functionality:
Block incoming packets except those that **belong to connections initiated by the VM**

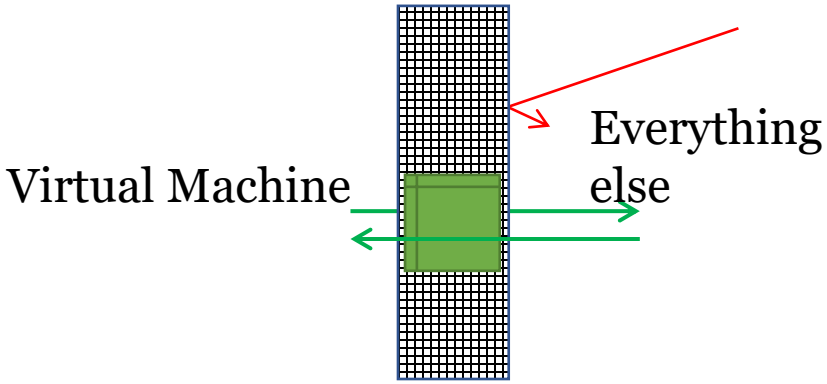
All Cloud Providers implement stateful NFs

Example Stateful NFs:

- Connection-tracking firewall
- Azure PrivateLink
- Stateful load balancers …



Connection-tracking firewall



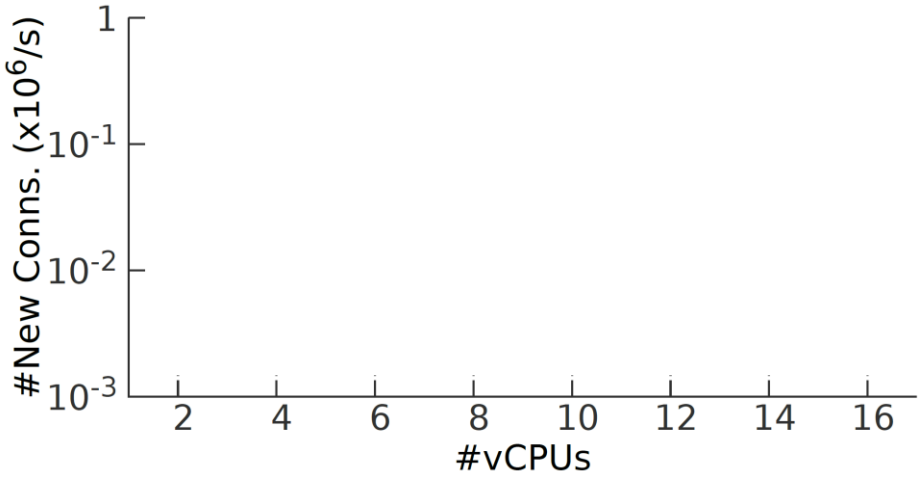
Functionality:
Block incoming packets except those that **belong to connections initiated by the VM**

State:
Proportional to **#Connections**

Perf impact:
Connection **scale** and **latency**

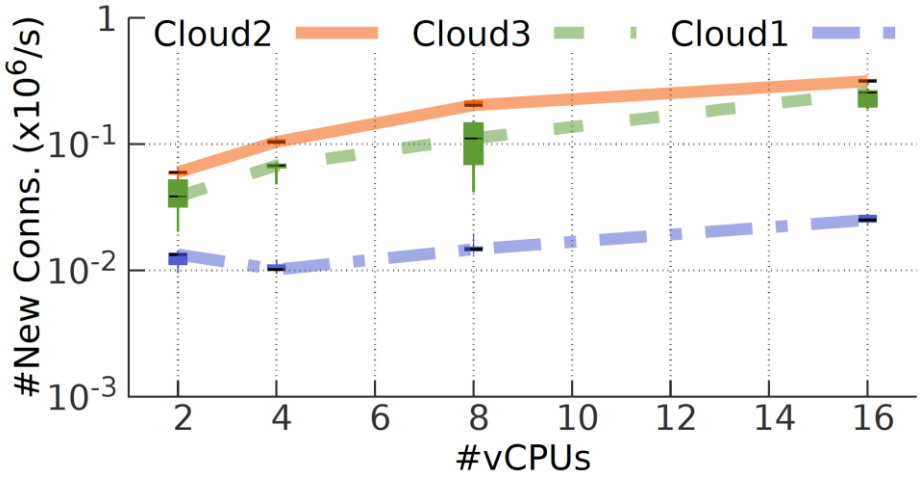
Today, stateful NFs are a performance bottleneck.

Today, stateful NFs are a performance bottleneck.



Method:
Pair of VMs.
TCP connections as-fast-as-you-can with one connection-tracking firewall rule

Today, stateful NFs are a performance bottleneck.

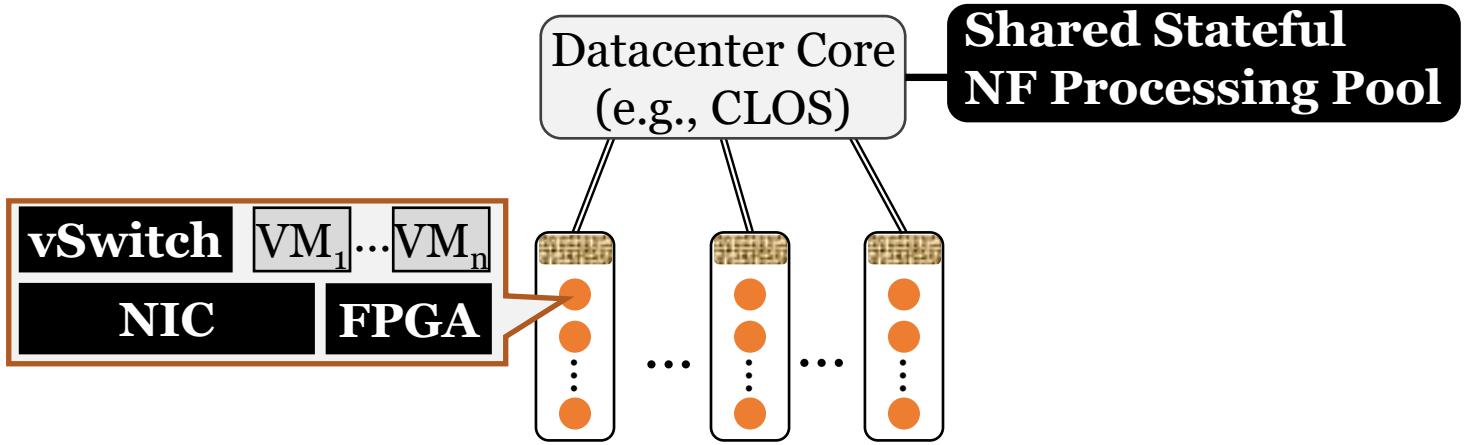


Method:
Pair of VMs.
TCP connections as-fast-as-you-can with one connection-tracking firewall rule

No cloud achieves more than 300K CPS

Disaggregating stateful NF processing

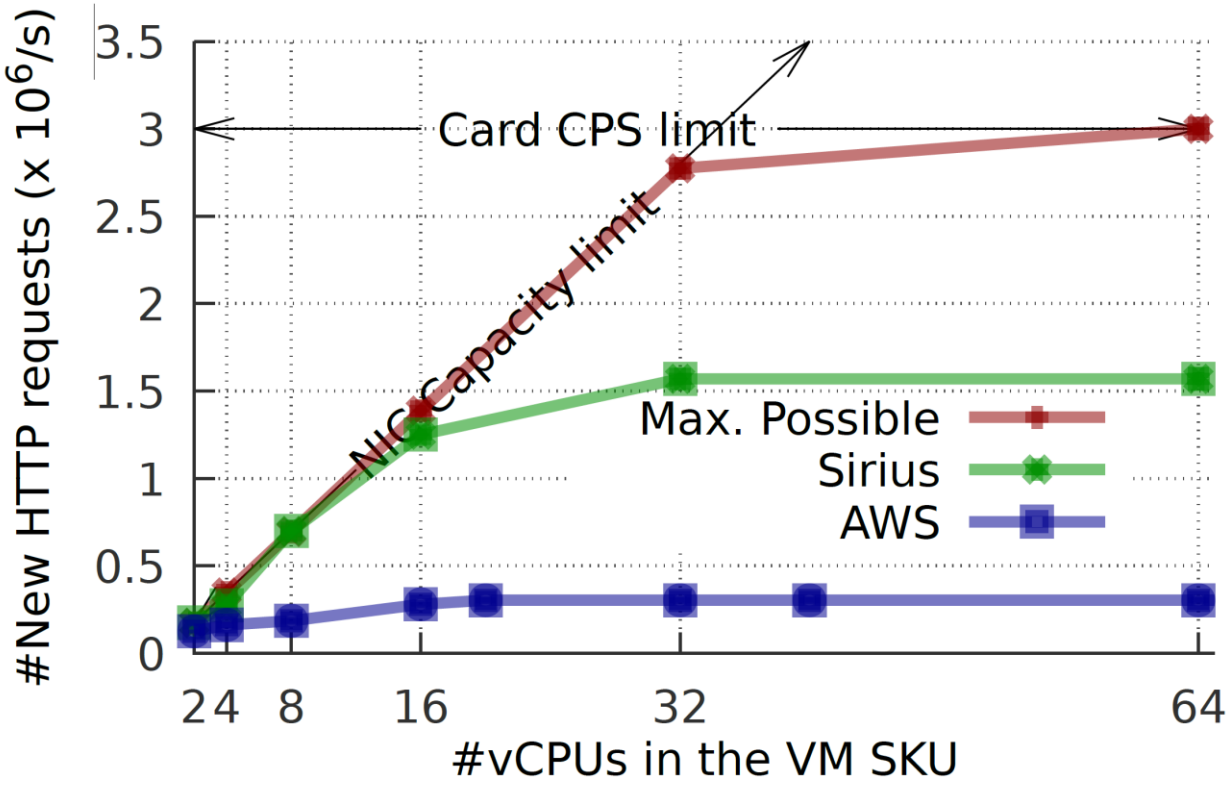
Disaggregating stateful NF processing



Floating NICs that onboard onto Sirius have their stateful NFs processed in a disaggregated pool of (Pensando) cards

With Sirius, connection scale increases to over 5X the closest other cloud

With Sirius, connection scale increases to over 5X the closest other cloud



TRex, ~250B request and response.
Note: on AWS, we use c5n.18xlarge

Disaggregating Stateful NFs has value beyond perf.

Agility	New features from Sirius can be used by VMs on any hardware SKU.
Cost	Inexpensive FPGAs at host may suffice since disaggregated pool handles all the spillover load.
Vendor-neutral	Can use best-of-class implementations of the appliance.

Key technical challenges

1. Performant stateful NF processing on the cards

Key technical challenges

1. Performant stateful NF processing on the cards

P4 programmable MPU pipelines + ARM cores + coherent large DRAM

Per card: 3M CPS, 16M conn., 2x100Gbps

Key technical challenges

1. Performant stateful NF processing on the cards

P4 programmable MPU pipelines + ARM cores + coherent large DRAM

Per card: 3M CPS, 16M conn., 2x100Gbps

2. Ensure high availability of state inspite link, switch and **card failures**

Key technical challenges

1. Performant stateful NF processing on the cards

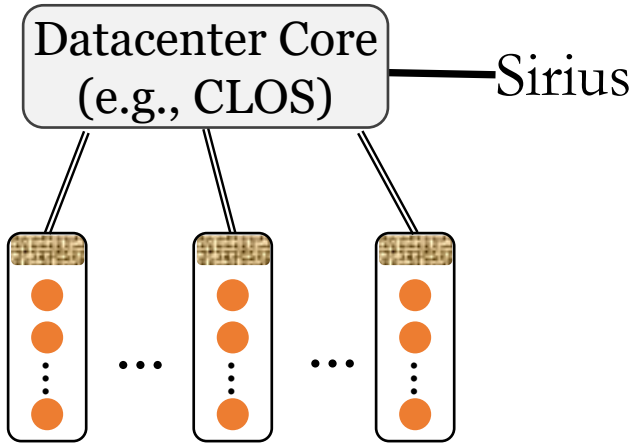
P4 programmable MPU pipelines + ARM cores + coherent large DRAM

Per card: 3M CPS, 16M conn., 2x100Gbps

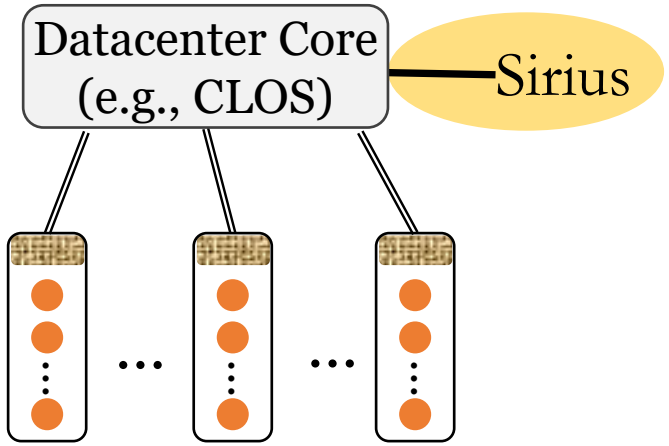
2. Ensure high availability of state inspite link, switch and **card failures**

Redundant paths, cards, … and
in-line state replication
(Paxos-style does not scale to packet state)

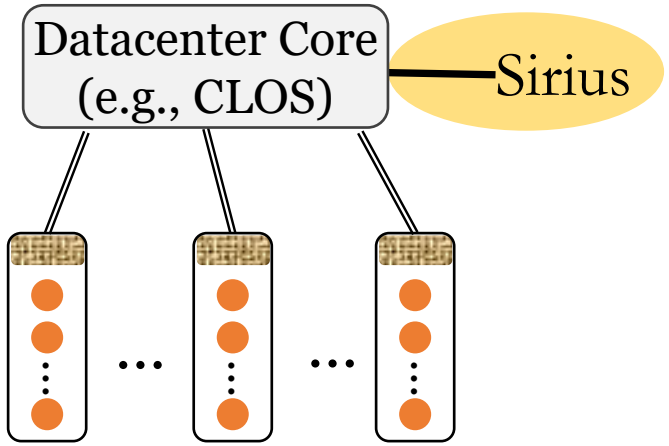
Disaggregation Architecture

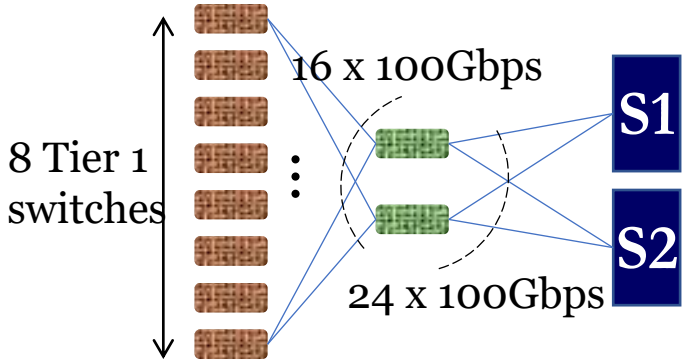


Disaggregation Architecture

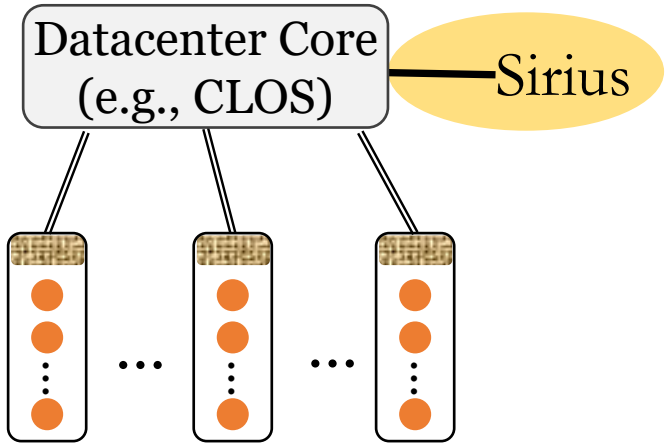


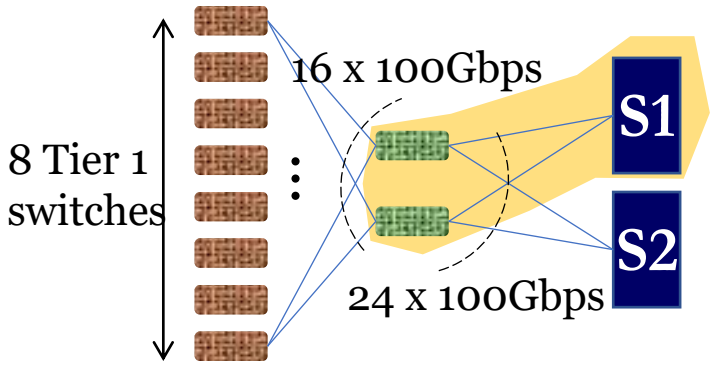
Disaggregation Architecture



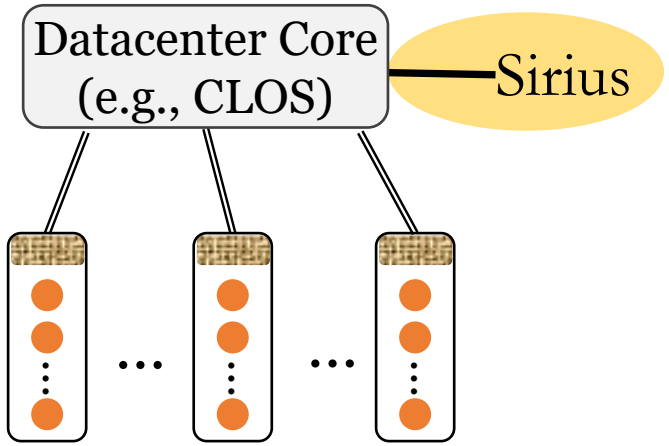


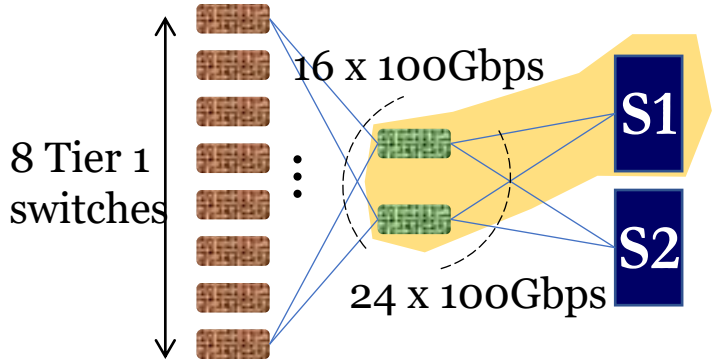
Disaggregation Architecture

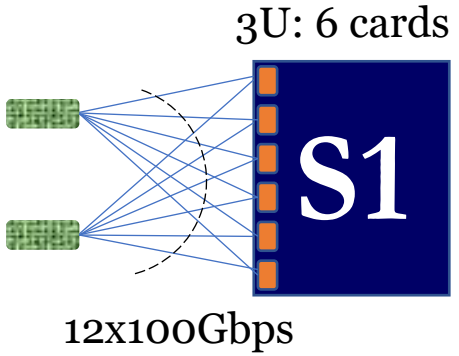




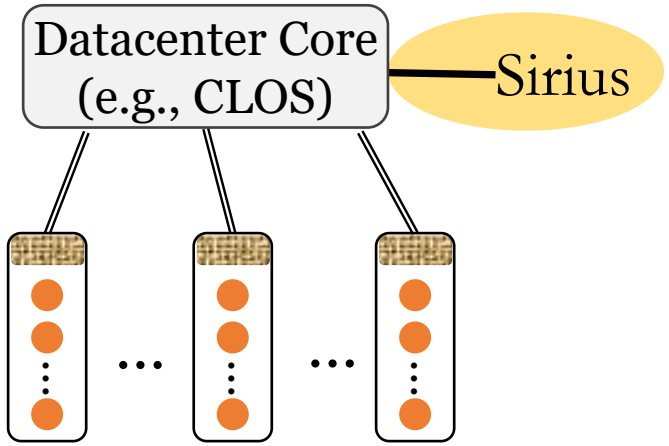
Disaggregation Architecture

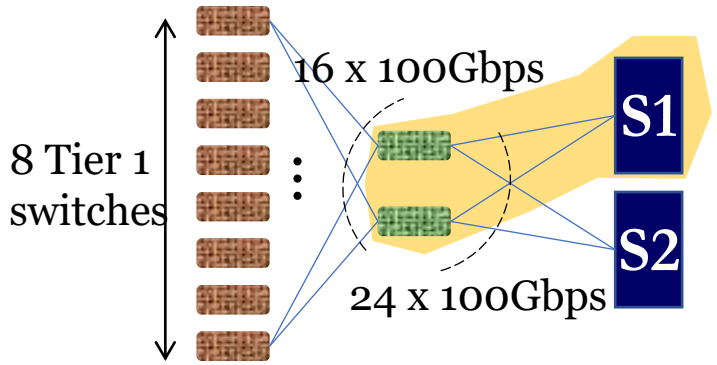


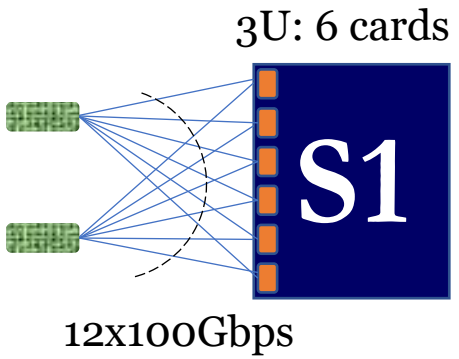




Disaggregation Architecture

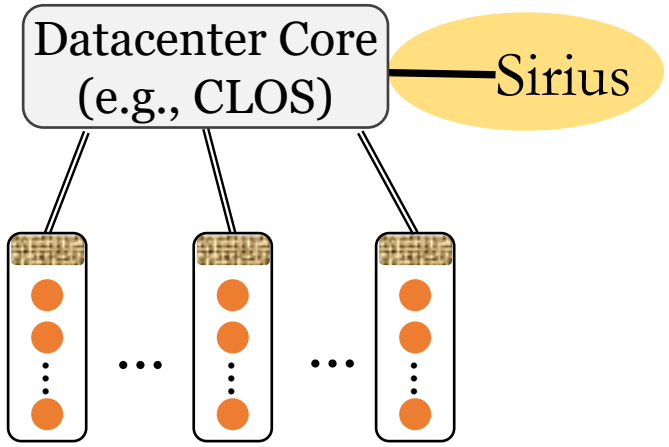


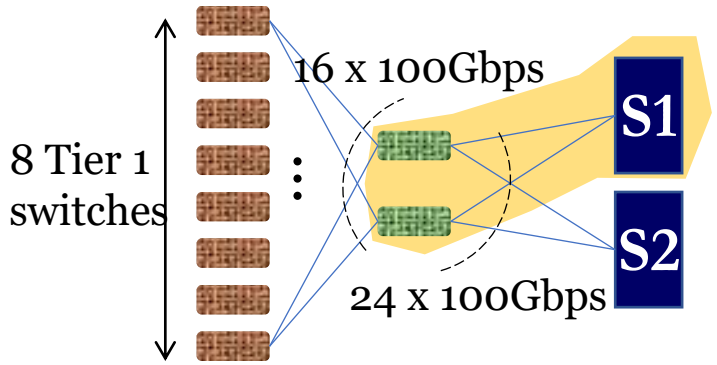


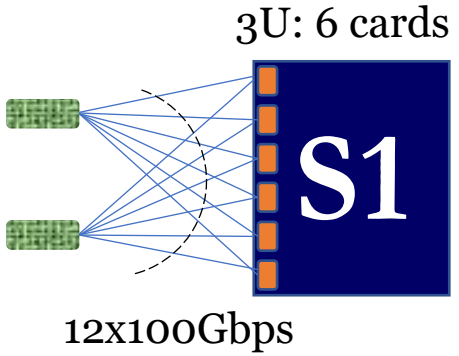


Availability — Access to state (at the cards) and NF processing capability remains as long as:

Disaggregation Architecture





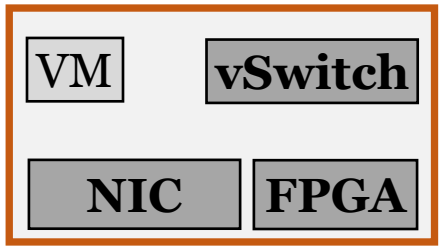


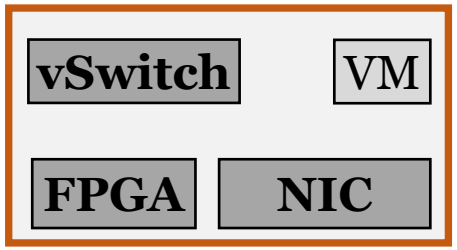
Availability — Access to state (at the cards) and NF processing capability remains as long as:

1. At most one of the two links connecting each card to green switches fail
2. At most one of the green switches fail
3. At most half of the links connecting green switches to the red switches fail
4. At most half of the red switches fail

Disaggregation Datapath Versus Direct Datapath

Direct (AccelNet)





State in FPGA
First packets go to vswitch

Disaggregation Datapath Versus Direct Datapath

Direct (AccelNet)

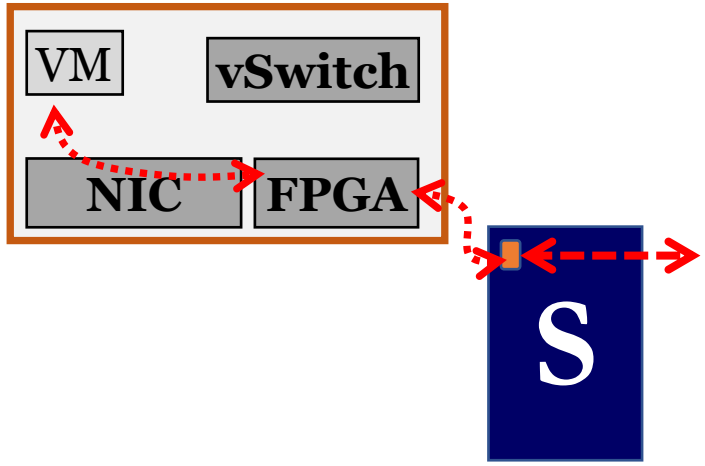
State in FPGA
First packets go to vswitch

Disaggregation Datapath Versus Direct Datapath

Direct (AccelNet)

State in FPGA
First packets go to vswitch

Disagg. (Sirius)



Sirius card announces address of fNIC

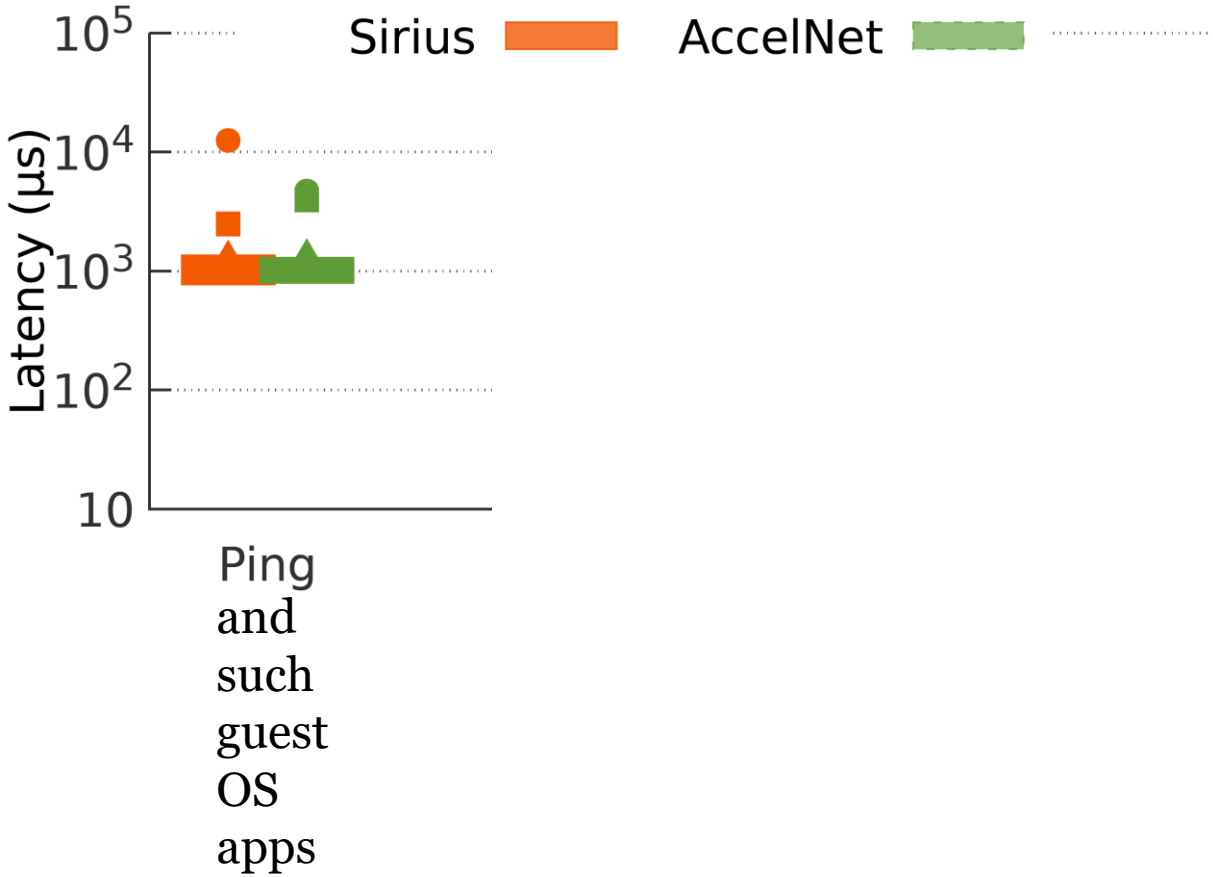
FPGA tunnels packets to the Sirius card

Datapath performance

VM pairs with both AccelNet and Sirius NICs

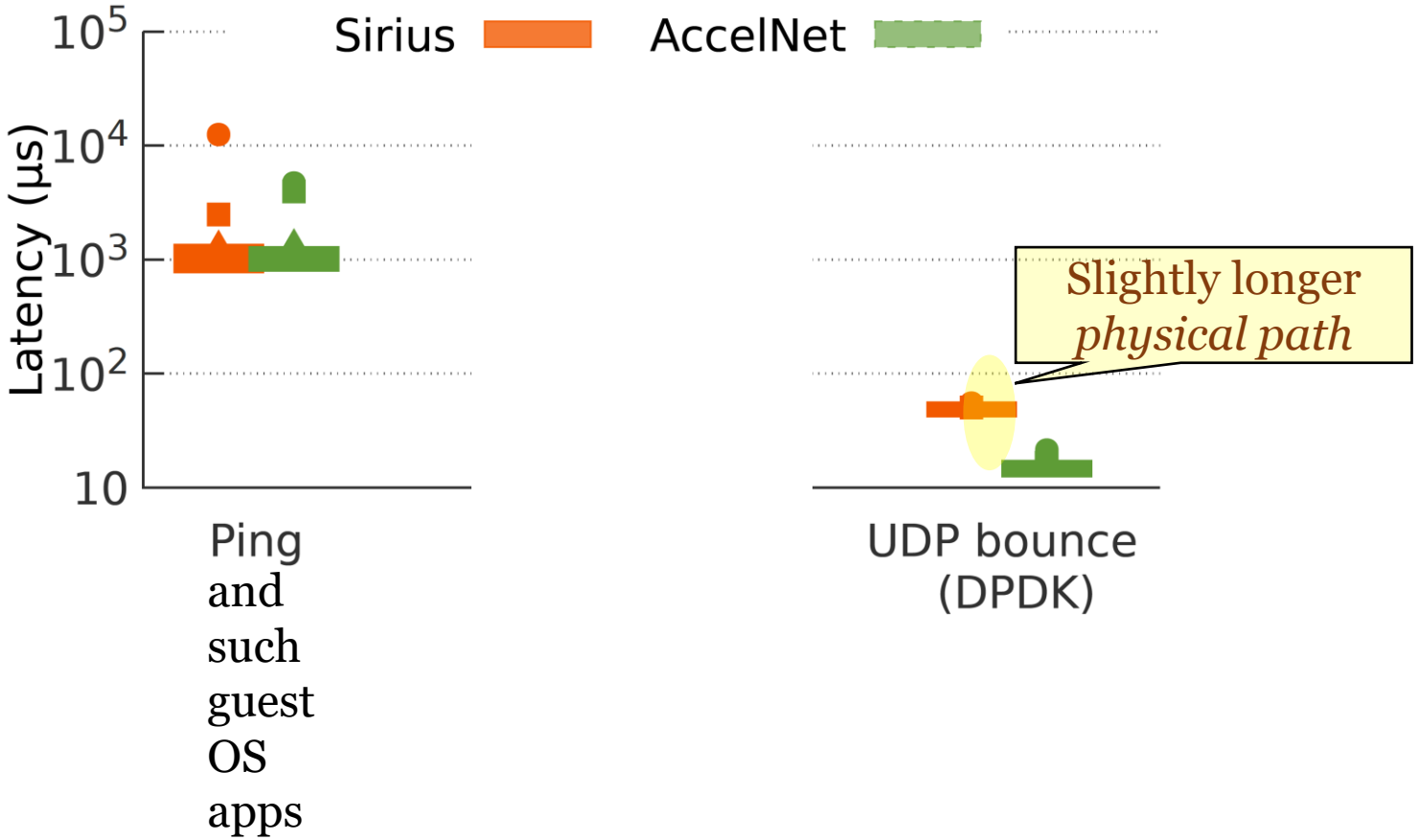
Datapath performance

VM pairs with both AccelNet and Sirius NICs



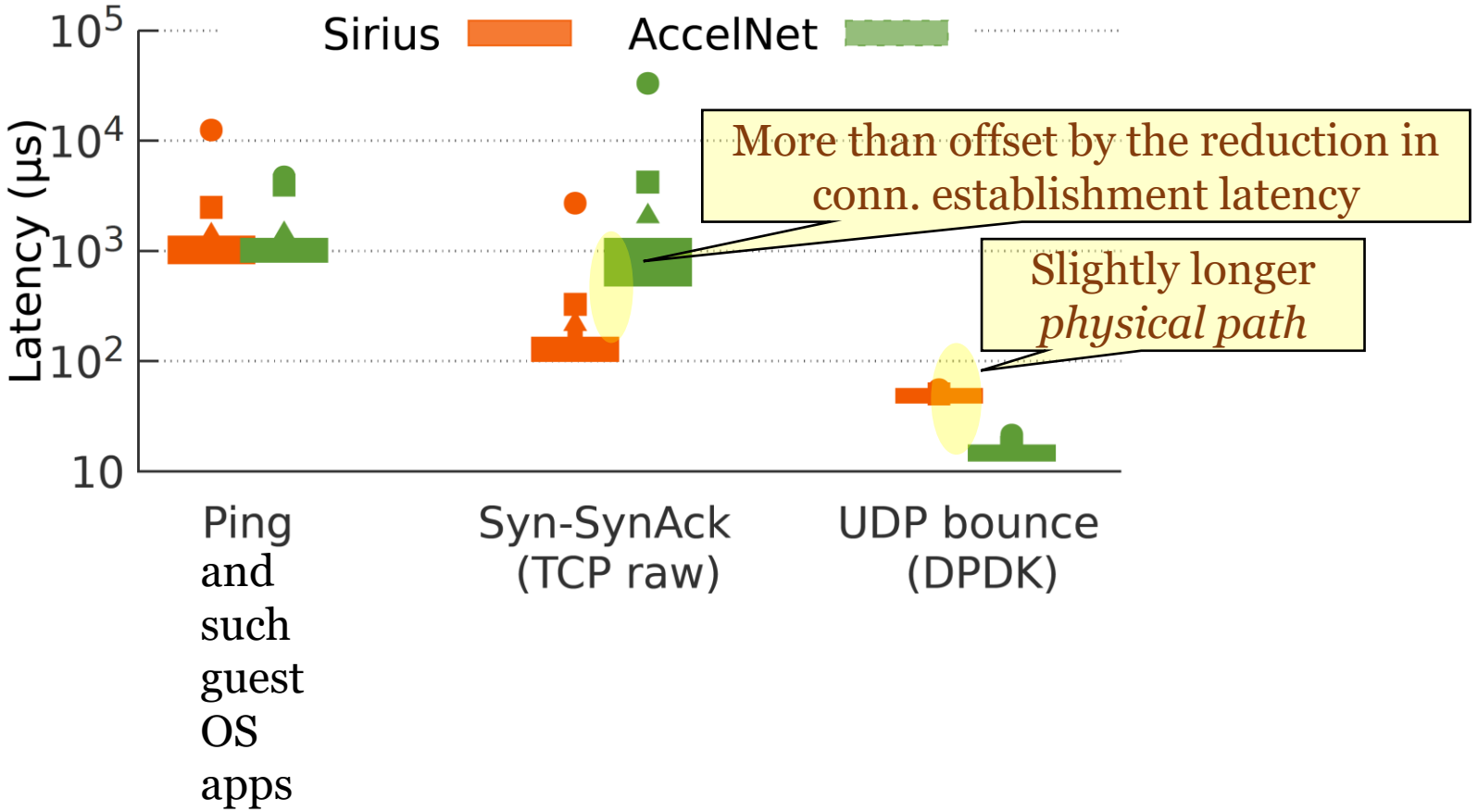
Datapath performance

VM pairs with both AccelNet and Sirius NICs



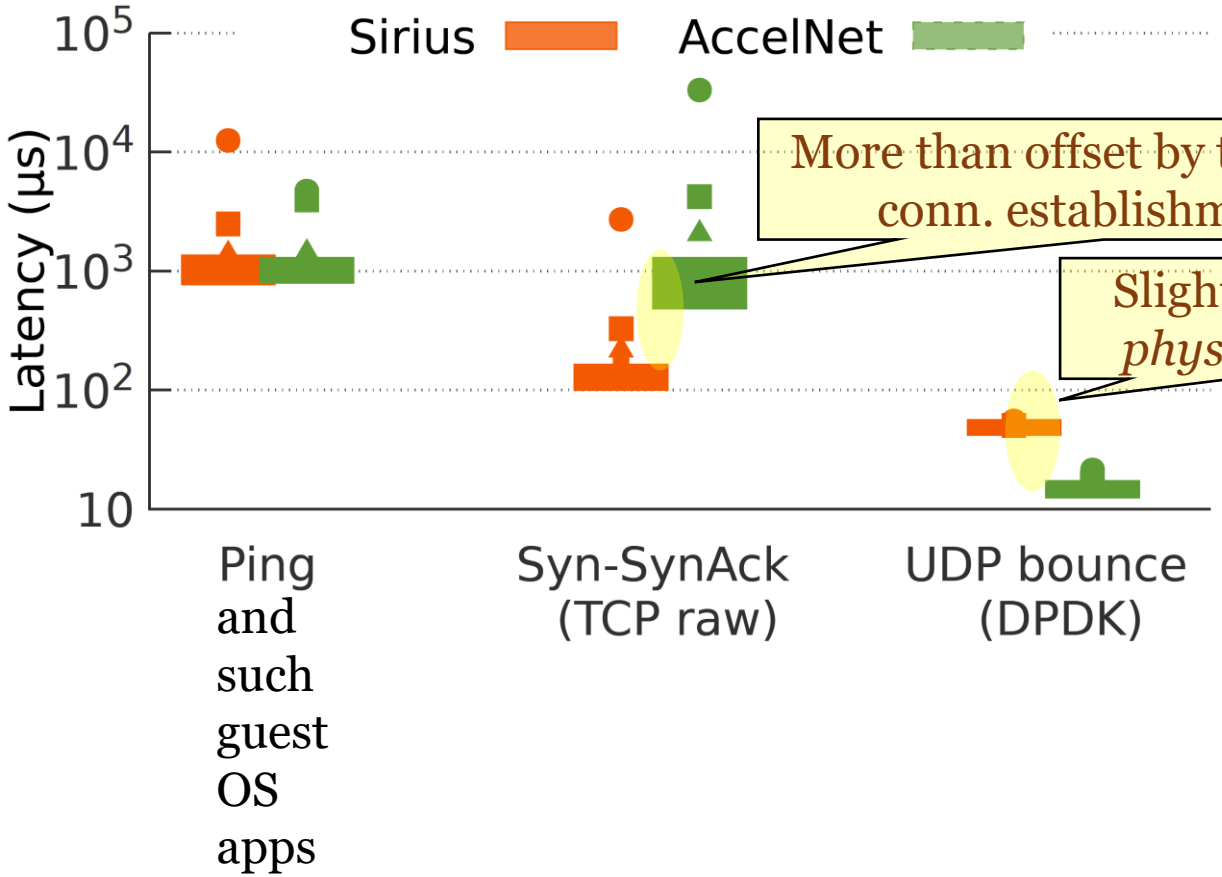
Datapath performance

VM pairs with both AccelNet and Sirius NICs



Datapath performance

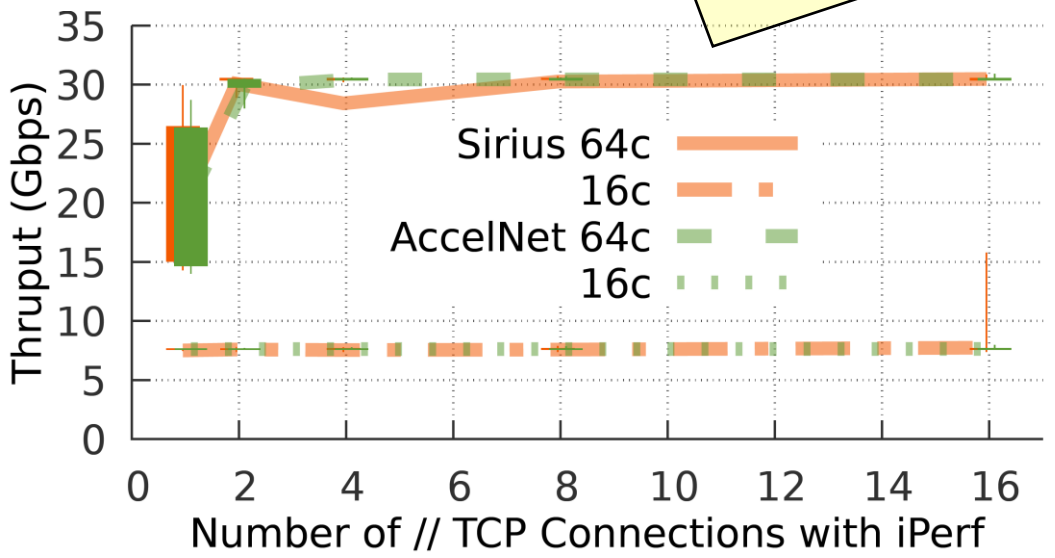
VM pairs with both AccelNet and Sirius NICs



More than offset by the reduction in conn. establishment latency

Slightly longer *physical path*

Thruput is statistically indistinguishable

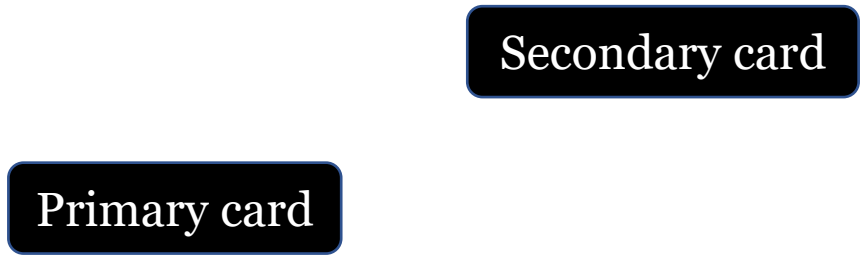


Protecting NF state on cards

If a card fails, state on that card will be lost.
So, we replicate state.

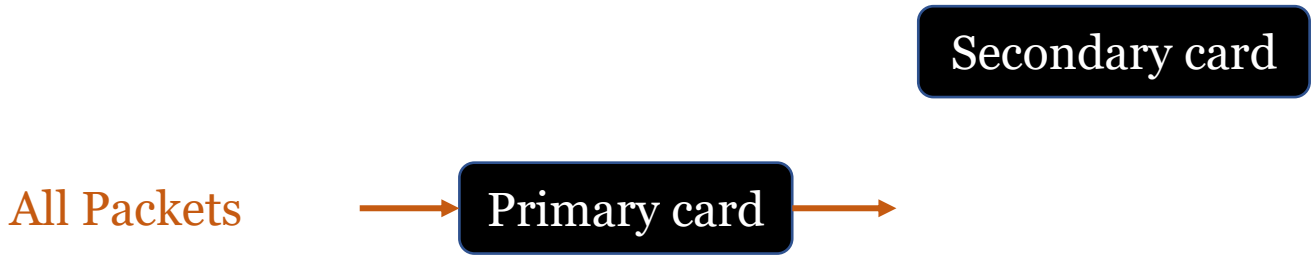
Protecting NF state on cards

If a card fails, state on that card will be lost.
So, we replicate state.



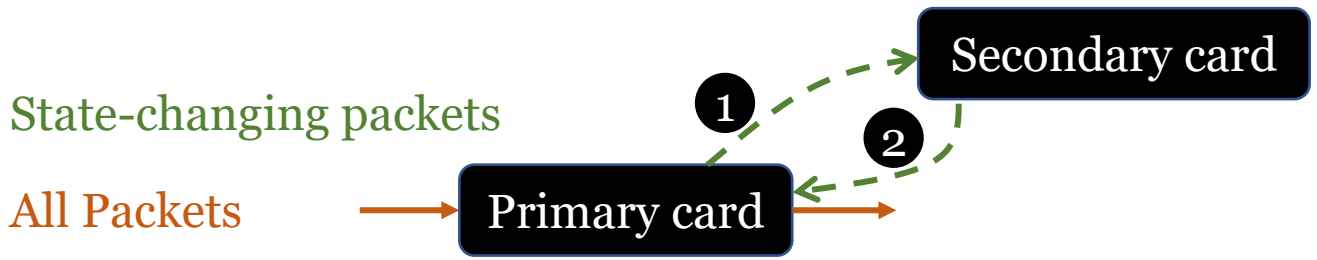
Protecting NF state on cards

If a card fails, state on that card will be lost.
So, we replicate state.



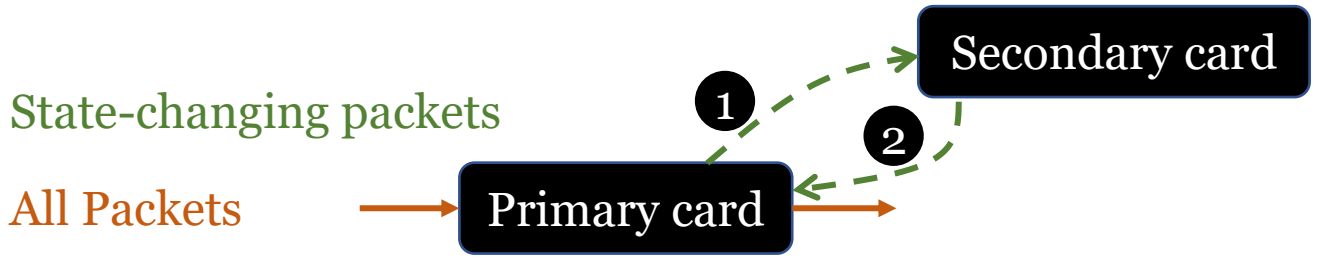
Protecting NF state on cards

If a card fails, state on that card will be lost.
So, we replicate state.



Protecting NF state on cards

If a card fails, state on that card will be lost.
So, we replicate state.

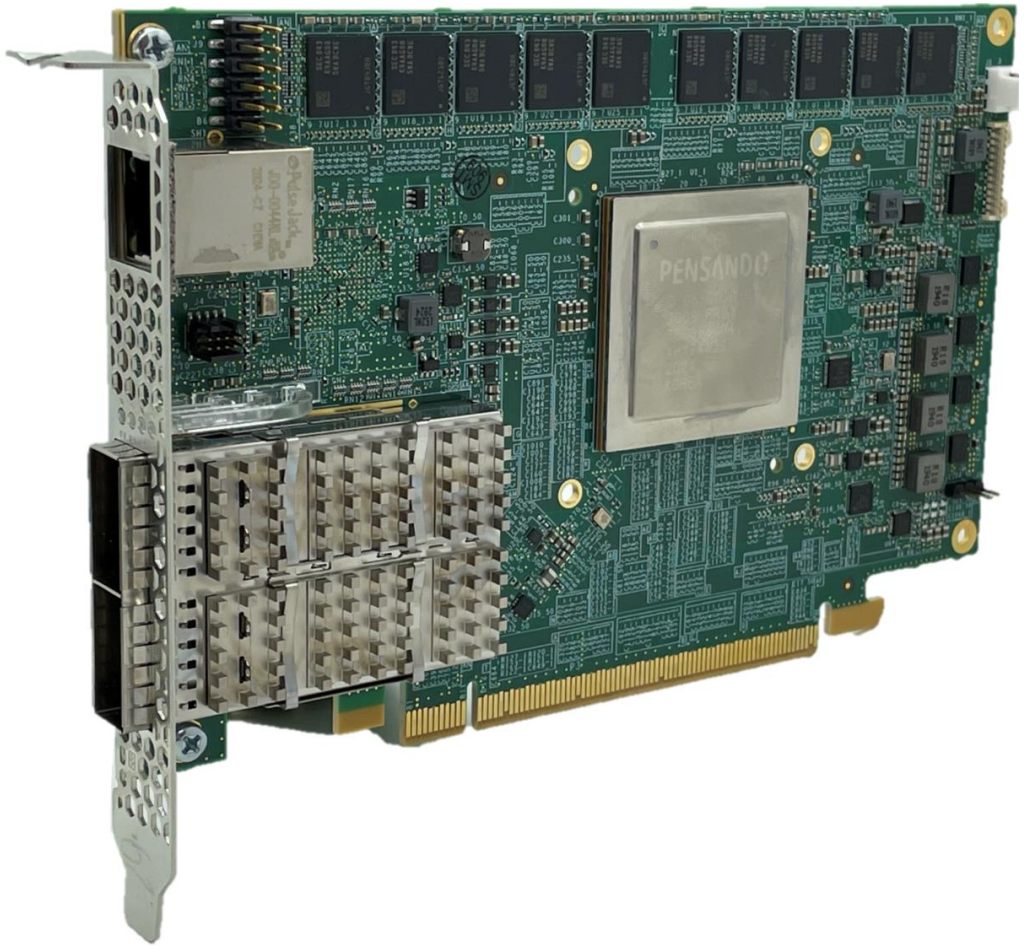


In-line State Replication

Nuances in the paper
Key properties:

- Safety
- Performant
  - No additional buffering of packets during replication
  - Card pairs are located on nearby appliances

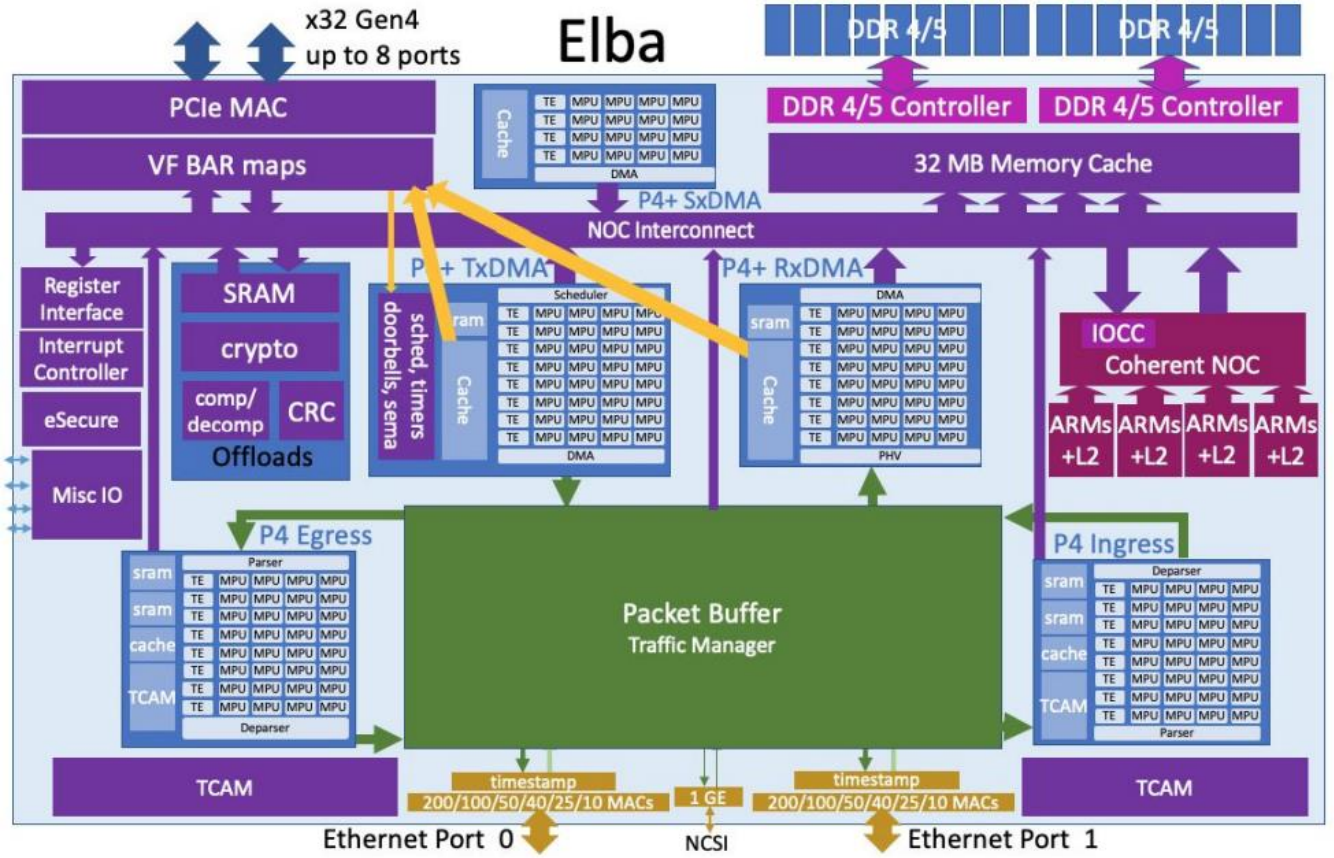
Performant stateful NF processing



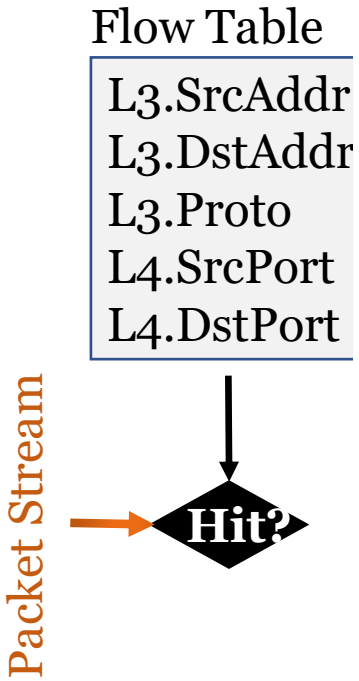
Performant stateful NF processing

P4 programmable MPUs for normal packet processing

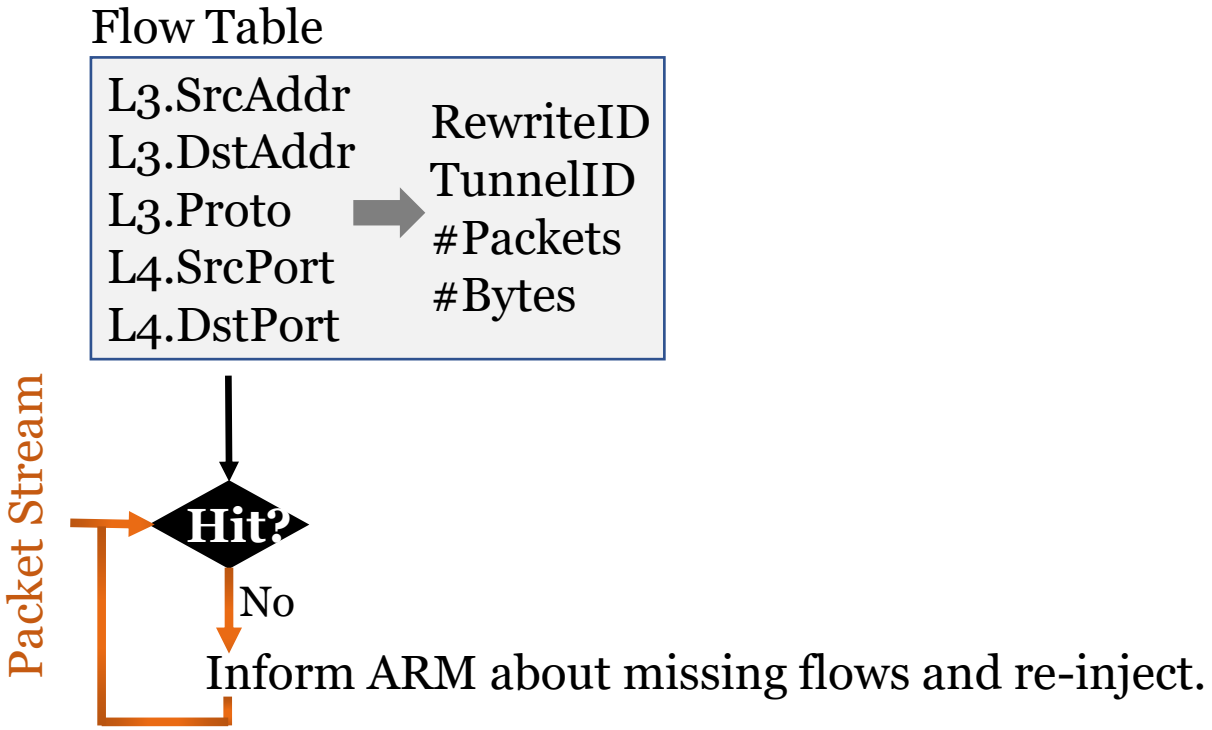
Novel MPUs that use state in the DRAM in packet processing



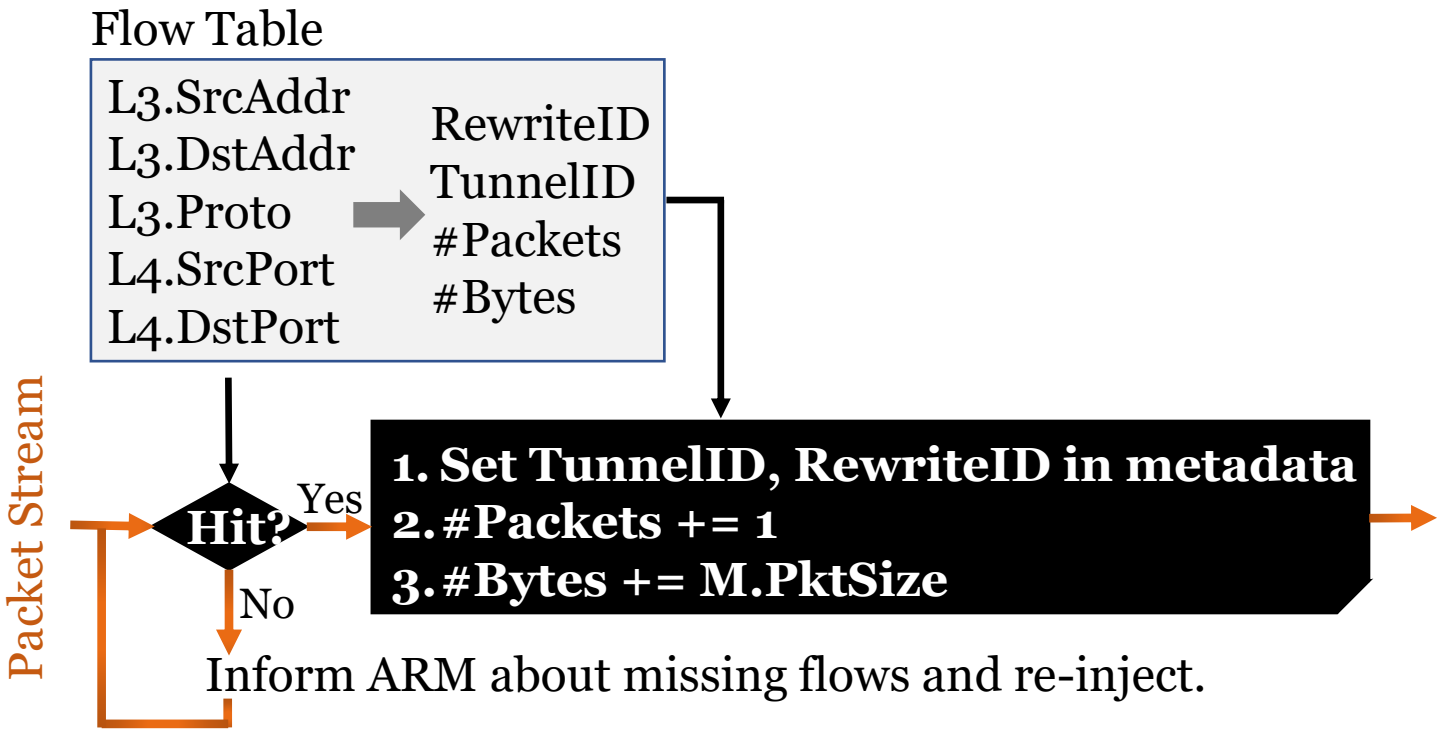
Example: Stateful Load Balancer with NAT



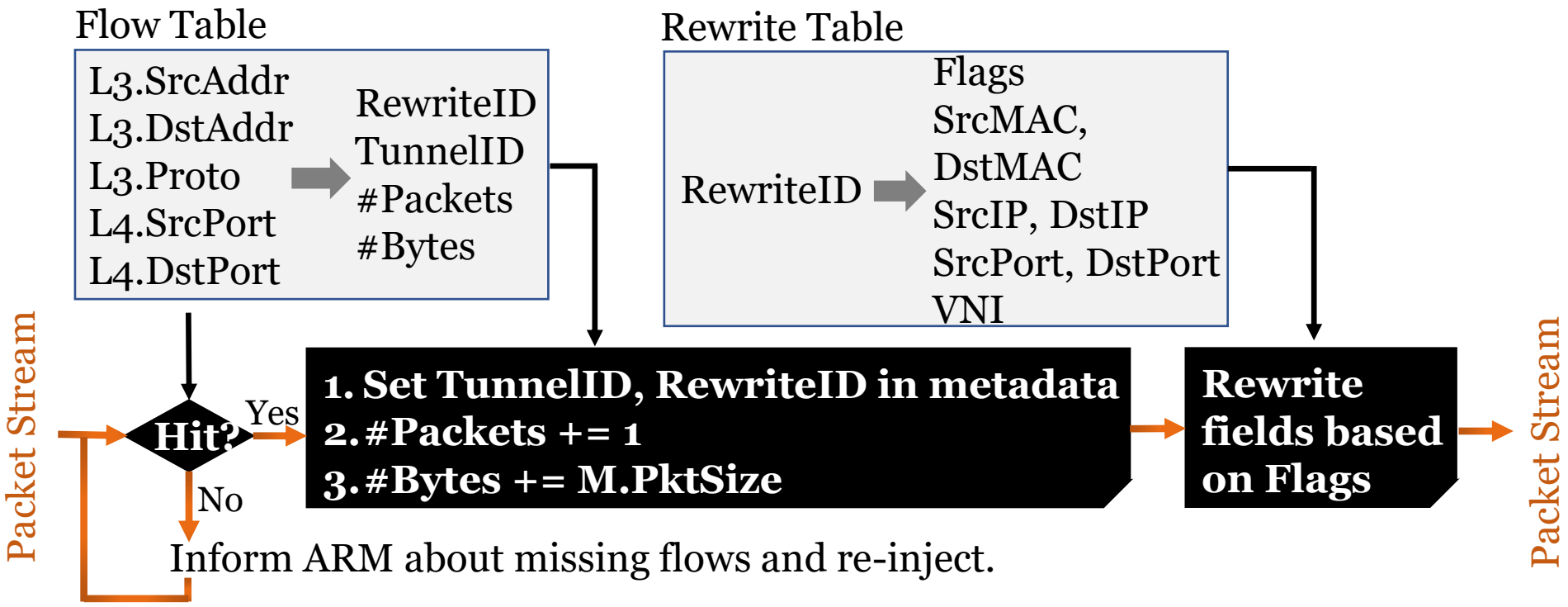
Example: Stateful Load Balancer with NAT



Example: Stateful Load Balancer with NAT



Example: Stateful Load Balancer with NAT



Other aspects

- Measurements of production NF load at Azure
- Control plane:
  - Failing over to secondary when primary is unreachable
  - Allocating fNICs to cards
- Splitting an fNIC's load between multiple cards
- On-demand spillover of fNIC from FPGA to Sirius (e.g., at load)
- Open APIs (DASH) to invite other implementations

Status and Demo

Sirius is in Public Preview at Azure and on track for GA

```
-Per port stats table
      ports |               1
 -----------------------------------------
   opackets |             308
     obytes |          35584
   ipackets |             412
     ibytes |          55288
    ierrors |
    oerrors |
      Tx Bw |          436.9

-Global stats enabled
 Cpu Utilization : 0.3
 Platform_factor : 1.0
 Total-Tx        :
 Total-Rx        :
 Total-PPS       :
 Total-CPS       :

 Expected-PPS    :
 Expected-CPS    :
 Expected-L7-BPS :

 Active-flows    :
 Open-flows      :
 drop-rate       :
 current time    : 149.
 test duration   : 0.0
```

```
Connecting to publisher

[SUCCESS]

Acquiring ports [1]:

Server_Info:

Server version:   v3.02
Server mode:      Advanc
Server CPU:       16 x I
Ports count:      1 x 50

-=TRex Console v3.0=-

Type 'help' or '?' for s

trex>start -f ./astf/htt

Loading traffic at acqui

Starting traffic.

6.89 [ms]

trex>
```

```
 opackets |         412871355
   obytes |       55288146648
```

-nan

trex>

https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/072e20e6-2aae-460b-a60e-eb66e0616224/resourceGroups/sk_sirius_dec7/providers/Microsoft.Compute/virtualMachines/dec64xV1/overview

Microsoft Azure (Preview)

Search resources, services, and docs (G+/)

SC-tl834@microsoft.com
MICROSOFT

Home > SK_SIRIUS_DEC7 >

dec64xV1

Virtual machine

Search

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings

- Networking
- Connect (Preview)
- Disks
- Size
- Microsoft Defender for Cloud
- Advisor recommendations
- Extensions + applications
- Continuous delivery
- Availability + scaling
- Configuration
- Identity
- Properties
- Locks

Operations

- Bastion
- Auto-shutdown
- Backup

Connect | Start | Restart | Stop | Capture | Delete | Refresh | Open in mobile | Feedback | CLI / PS

Advisor (1 of 1): Endpoint protection should be installed on machines →

Essentials

View Cost | JSON View

Resource group (move)	: sk_sirius_dec7	Operating system	: Linux (ubuntu 20.04)
Status	: Running	Size	: Standard D64s v3 (64 vcpus, 256 GiB memory)
Location	: West Central US	Public IP address	: 20.69.19.24
Subscription (move)	: MSR Redmond Networking Research Group	Virtual network/subnet	: sk_sirius_dec7_vnet/default
Subscription ID	: 072e20e6-2aae-460b-a60e-eb66e0616224	DNS name	: Not configured
Tags (edit)	: fastpathenabled : True		

Properties | Monitoring | Capabilities (7) | Recommendations (1) | Tutorials

Virtual machine

Computer name	dec64xV1
Health state	-
Operating system	Linux (ubuntu 20.04)
Publisher	Canonical
Offer	UbuntuServer
Plan	18_04-lts-gen2
VM generation	V2
VM architecture	x64
Agent status	Ready
Agent version	
Host group	
Host	
Proximity pl	

Networking

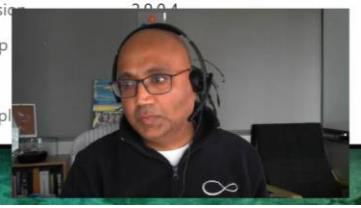
Public IP address	20.69.19.24 (Network interface dec64xv1510)
Public IP address (IPv6)	-
Private IP address	10.0.0.4
Private IP address (IPv6)	-
Virtual network/subnet	sk_sirius_dec7_vnet/default
DNS name	Configure

Size

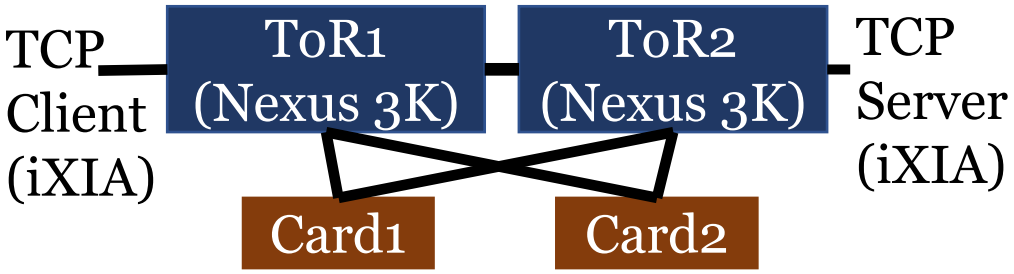
Size	Standard D64s v3
vCPUs	64
RAM	256 GiB

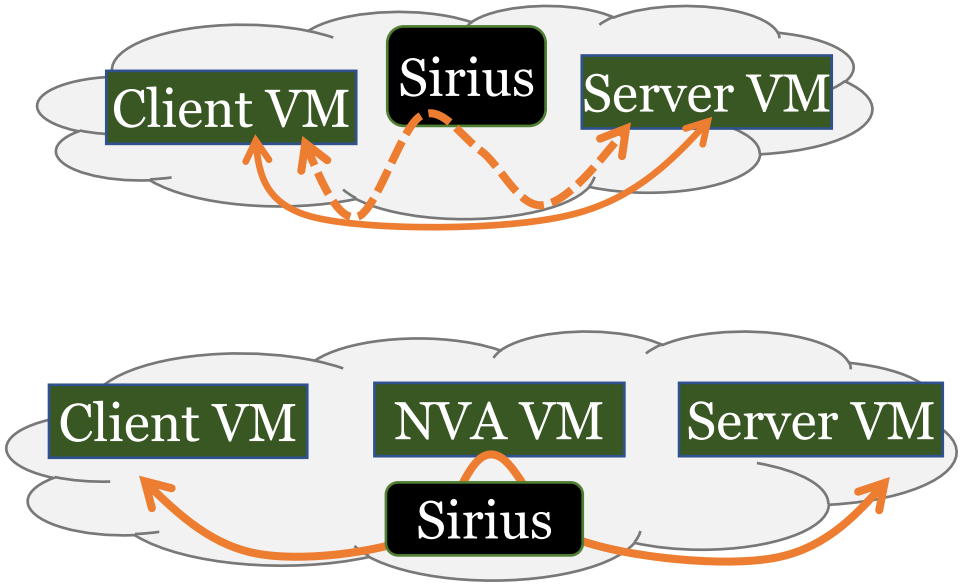
Disk

OS disk	dec64xV1_OsDisk_1_f0917a72ba944f129daa87e549791210



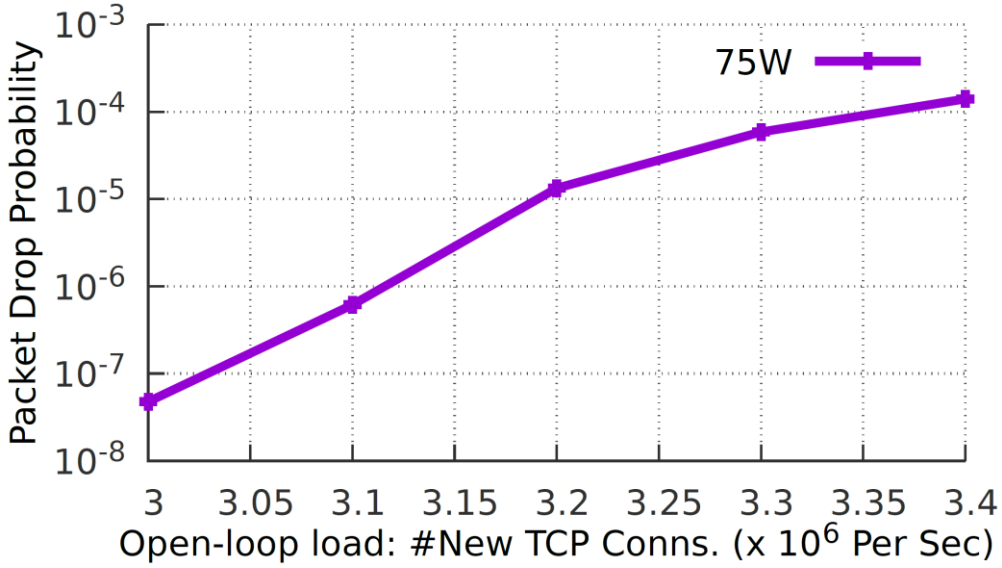
Evaluation

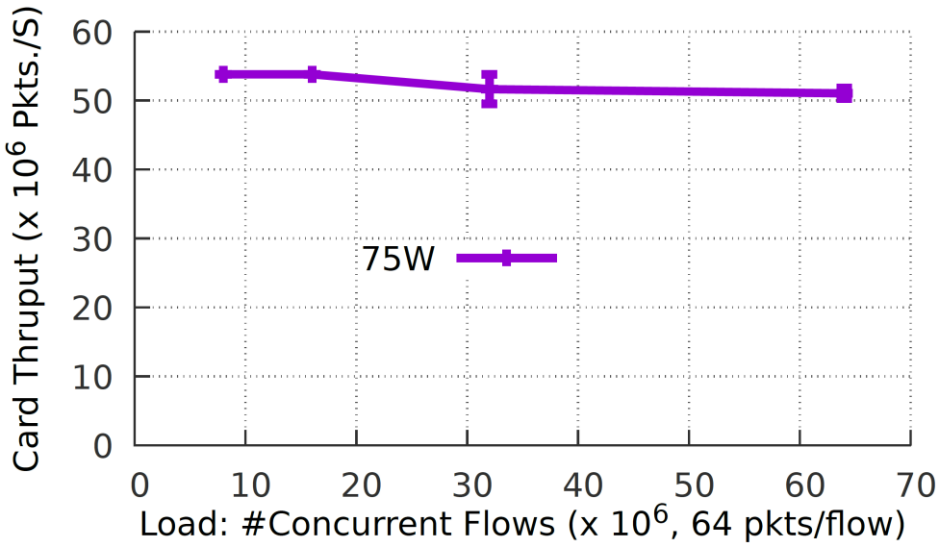




In Azure

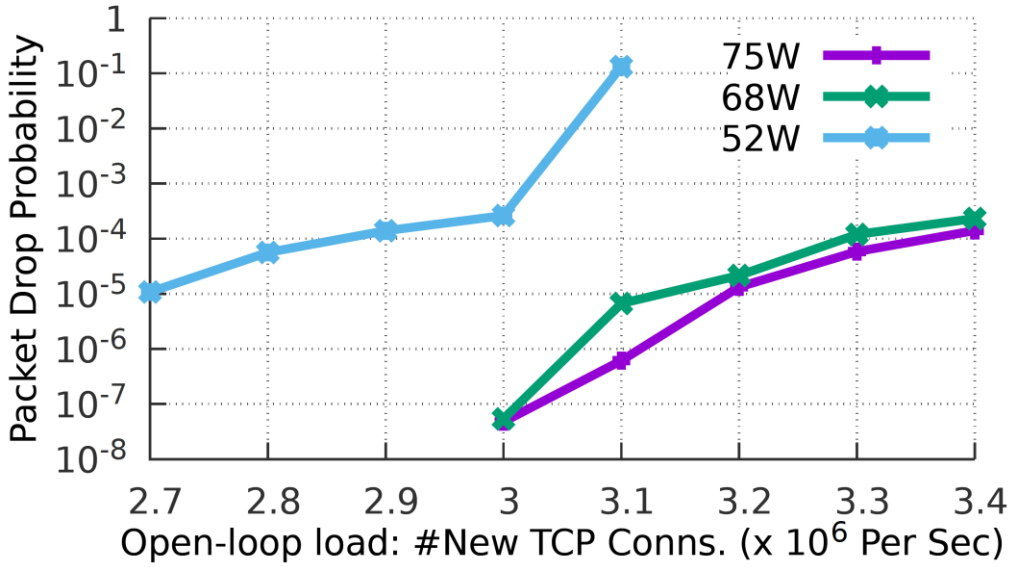
Performance in the lab

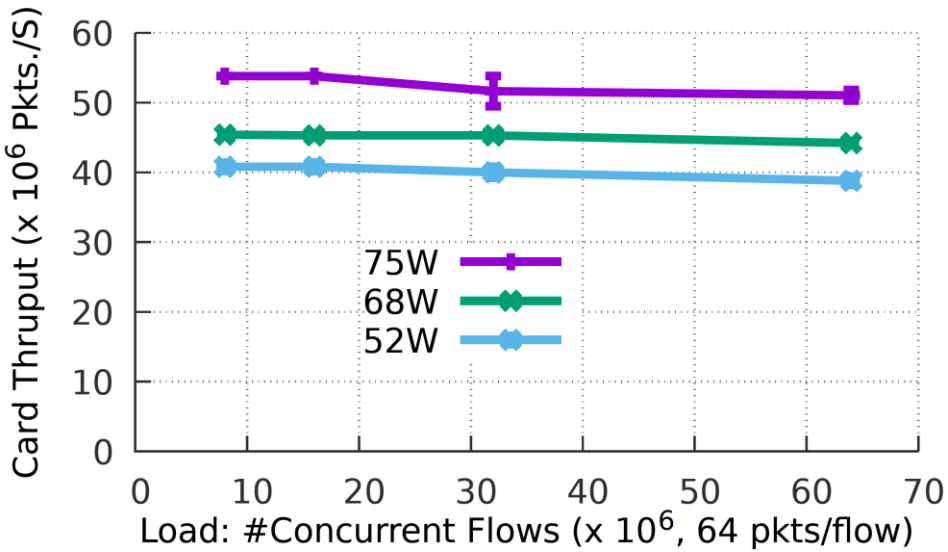




Each card supports high CPS, large numbers of concurrent flows (and PPS)

Performance in the lab

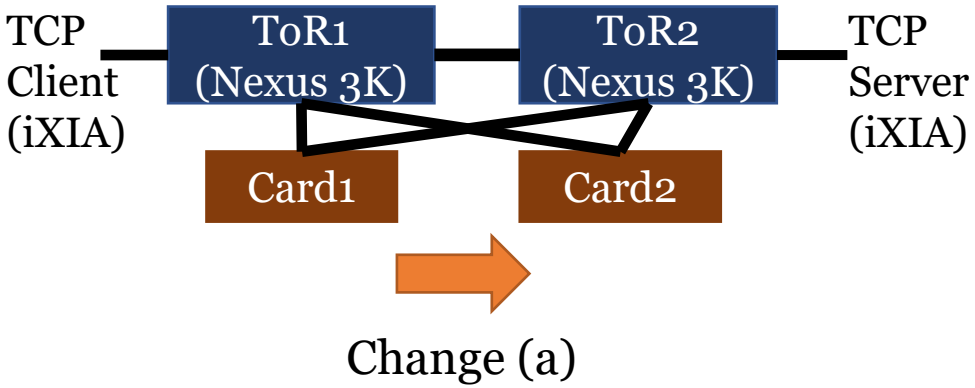




Each card supports high CPS, large numbers of concurrent flows (and PPS)

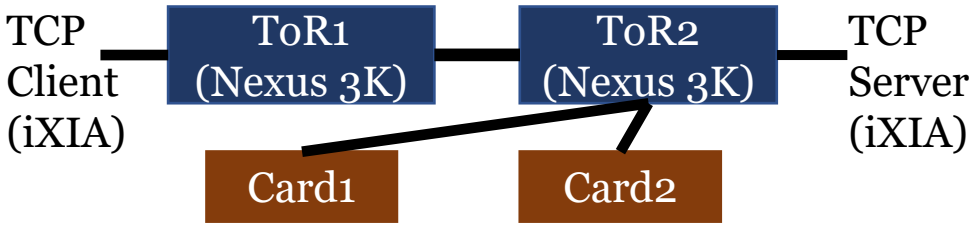
Reducing power appears possible

Performance under faults



Change
(a) Planned switchover
(b) ToR1's links to both cards are down
(c) Card1's links to both ToRs are down

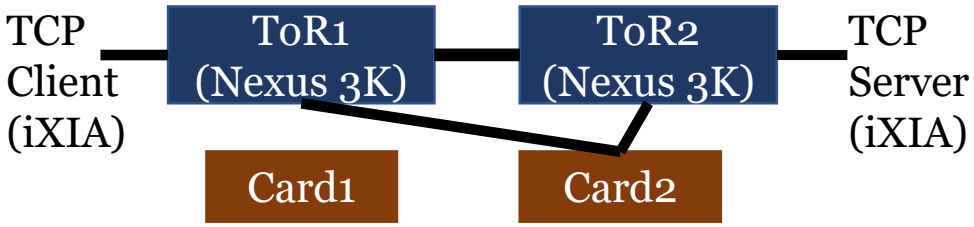
Performance under faults



Change (b)

Change
(a) Planned switchover
(b) ToR1's links to both cards are down
(c) Card1's links to both ToRs are down

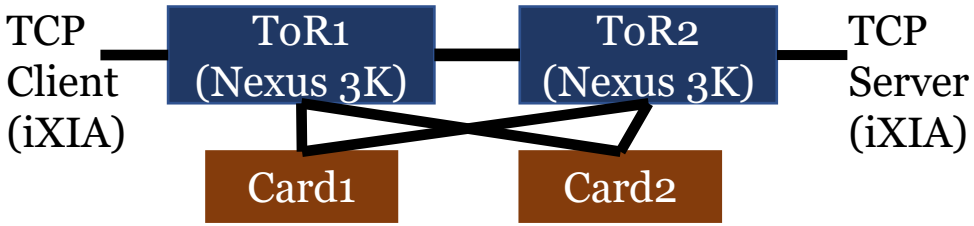
Performance under faults



Change (c)

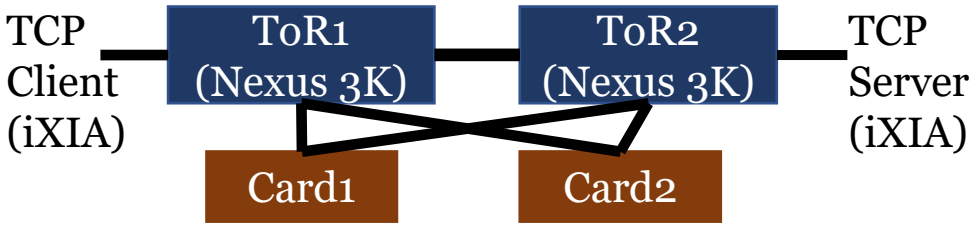
Change
(a) Planned switchover
(b) ToR1's links to both cards are down
(c) Card1's links to both ToRs are down

Performance under faults



Change	#Flow breaks	% of pkts dropped		Recovery Latency
		All	At Cards	
(a) Planned switchover				
(b) ToR1's links to both cards are down				
(c) Card1's links to both ToRs are down				

Performance under faults



Change	#Flow breaks	% of pkts dropped		Recovery Latency
		All	At Cards	
(a) Planned switchover	0	0.00316%	0	1.89ms
(b) ToR1's links to both cards are down	0	0.00929%	0.0000227%	5.75ms
(c) Card1's links to both ToRs are down	0	0.00835%	0.0000201%	5.01ms

Sirius disaggregates Stateful NF Processing

Stateful NF processing is resource-intensive and a key bottleneck

Disaggregating stateful NF processing

saves $, offers higher perf, allows laissez-faire deployment

Sirius disaggregates Stateful NF Processing

Stateful NF processing is resource-intensive and a key bottleneck

Disaggregating stateful NF processing

saves $, offers higher perf, allows laissez-faire deployment

Novel techniques

- Splitting functions between ARMs, packet-MPUs and MPUs that access DRAM
- In-line state replication
- Efficient failover
- …

Sirius disaggregates Stateful NF Processing

Stateful NF processing is resource-intensive and a key bottleneck

Disaggregating stateful NF processing

saves $, offers higher perf, allows laissez-faire deployment

Novel techniques

- Splitting functions between ARMs, packet-MPUs and MPUs that access DRAM
- In-line state replication
- Efficient failover
- …

State-of-the-art performance available at Azure in preview.

sirius-aznet@microsoft.com