Enabling Flexible Network FPGA Clusters in a Heterogeneous Cloud Data Center

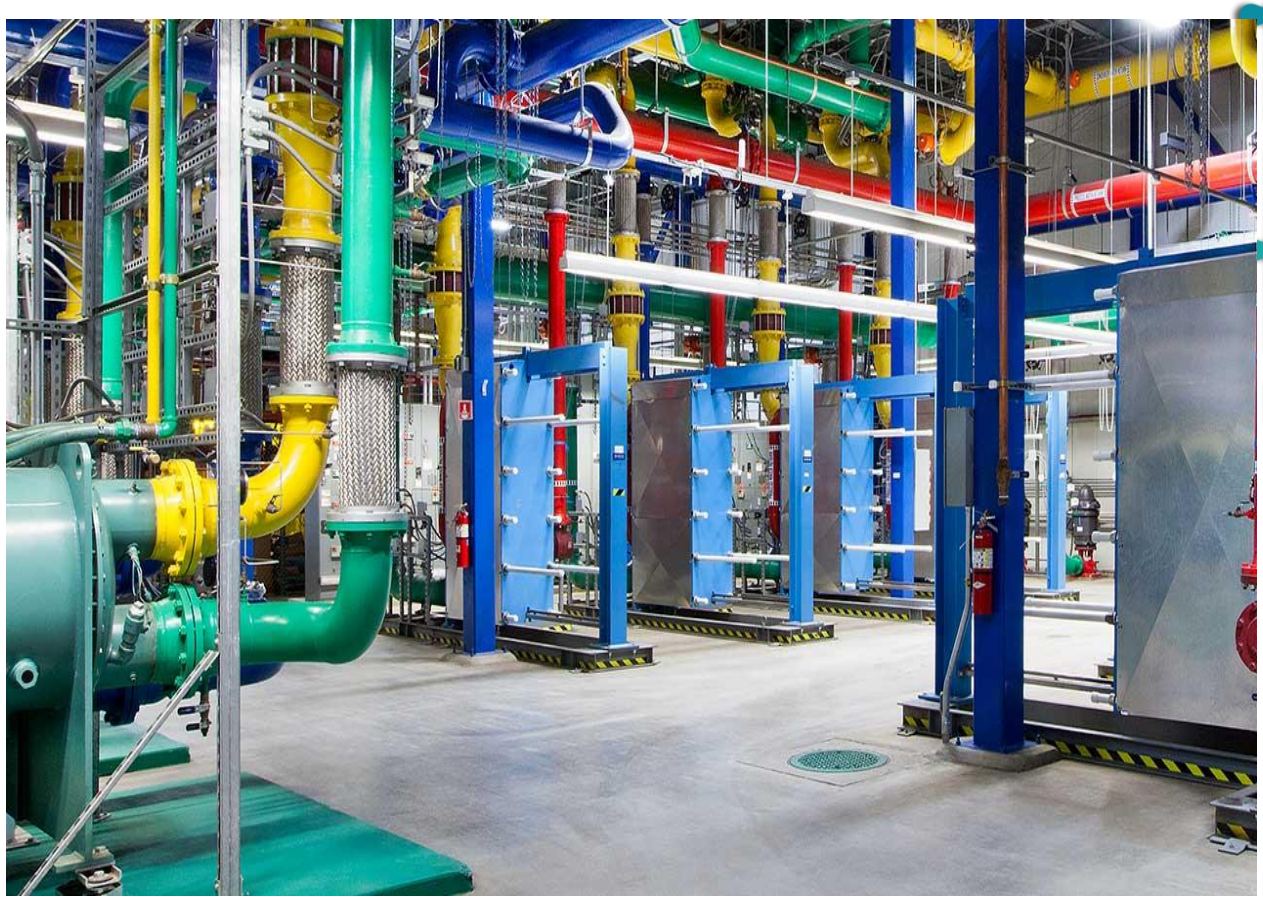
Naif Tarafdar, Thomas Lin, Eric Fukuda,

Hadi Bannazadeh, Alberto Leon-Garcia, Paul Chow

University of Toronto

Cloudy with a chance of FPGAs?

- Big data needs more compute power: How about FPGAs in datacenters?
- Datacenters are approximately 200 billion dollar industry
- Datacenter applications are large
- No large scale datacenter multi-FPGA fabric deployed until 2014

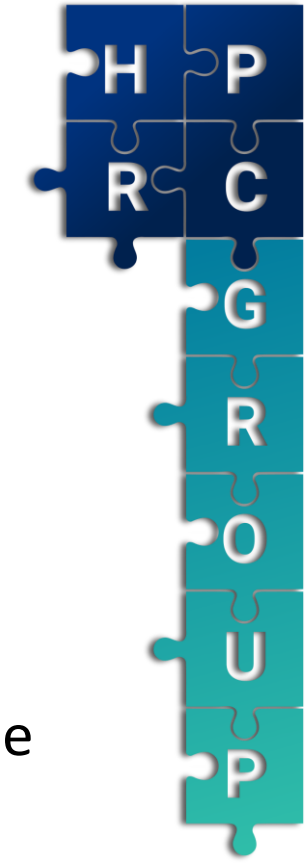


Cloudy with ~~a Chance of~~ FPGAs?

- Microsoft Catapult
  - 1632 Servers
  - Bing search engine
  - 10 % more power, 95 % more throughput
- Intel acquisition of Altera in December 2015
- More than just a *chance*

Large Clusters Difficult To Manage

- Large resources of clusters difficult to manage
- Expensive
- Solution:
  - Allow users framework to create their own clusters from a pool of available resources



Related Work

- Byma et al:
  - FPGA broke into partial FPGAs
  - Multiple users share portion of FPGA

Related Work

- Byma et al:
  - FPGA broke into partial FPGAs
  - Multiple users share portion of FPGA
- IBM Supervessel:
  - FPGA tightly coupled with virtual machine CPU
  - Connected to CPU via shared memory
  - Network connection through CPU

Related Work

- Byma et al:
  - FPGA broke into partial FPGAs
  - Multiple users share portion of FPGA
- IBM Supervessel:
  - FPGA tightly coupled with Virtual Machine CPU
  - Connected to CPU via shared memory
  - Network connection through CPU
- Amazon:
  - FPGA(s) tightly coupled with virtual machine CPU
  - Up to 8 FPGAs connected via high performance network link

Related Work

- Byma et al:
  - FPGA broke into partial FPGAs
  - Multiple users share portion of FPGA
- IBM Supervessel:
  - FPGA tightly coupled with Virtual Machine CPU
  - Connected to CPU via shared memory
  - Network connection through CPU
- Amazon:
  - FPGA(s) tightly coupled with virtual machine CPU
  - Up to 8 FPGAs connected via high performance network link
- IBM Hyperscale
  - Network connected FPGAs
  - Modified Openstack to accept bitstream and then returns IP address and programmed FPGA to user

Problems We Target

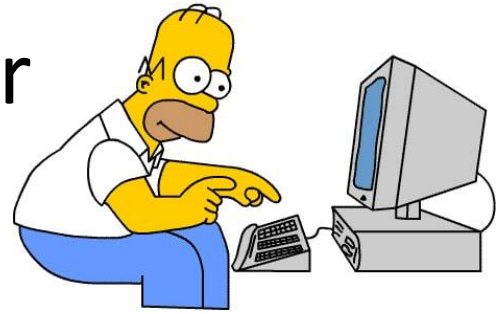
- Large multi-FPGA systems
  - Create abstraction between FPGAs in multi-FPGA systems
  - Easy scalability of system

Problems We Target

- Large multi-FPGA systems
  - Create abstraction between FPGAs in multi-FPGA systems
  - Easy scalability of system
- Network capabilities
  - FPGA cluster directly accessible by any other network device in the datacenter

Overall System View

User



Input From User

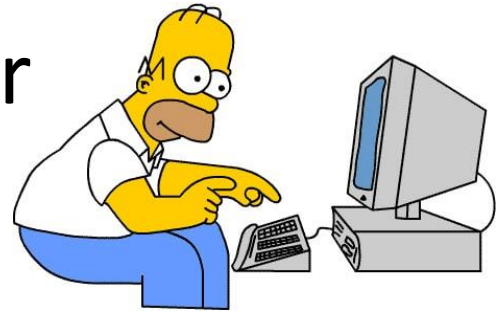
FPGA Mapping File

Logical Cluster Description

FPGA Cluster Generator

Overall System View

User



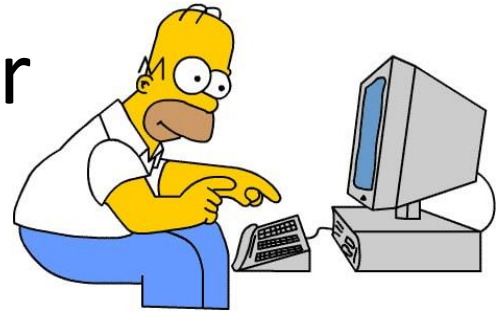
FPGA Cluster Generator

Output to VM with FPGA Tools

Individual FPGA Projects

Overall System View

User



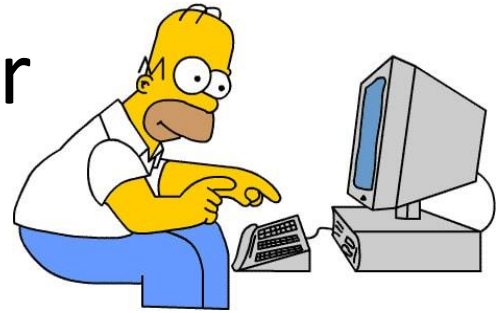
FPGA Cluster Generator

Output to Cloud Manager

- Command For Resource Allocation
- Commands For Connecting FPGAs to Network

Overall System View

User



Output To User

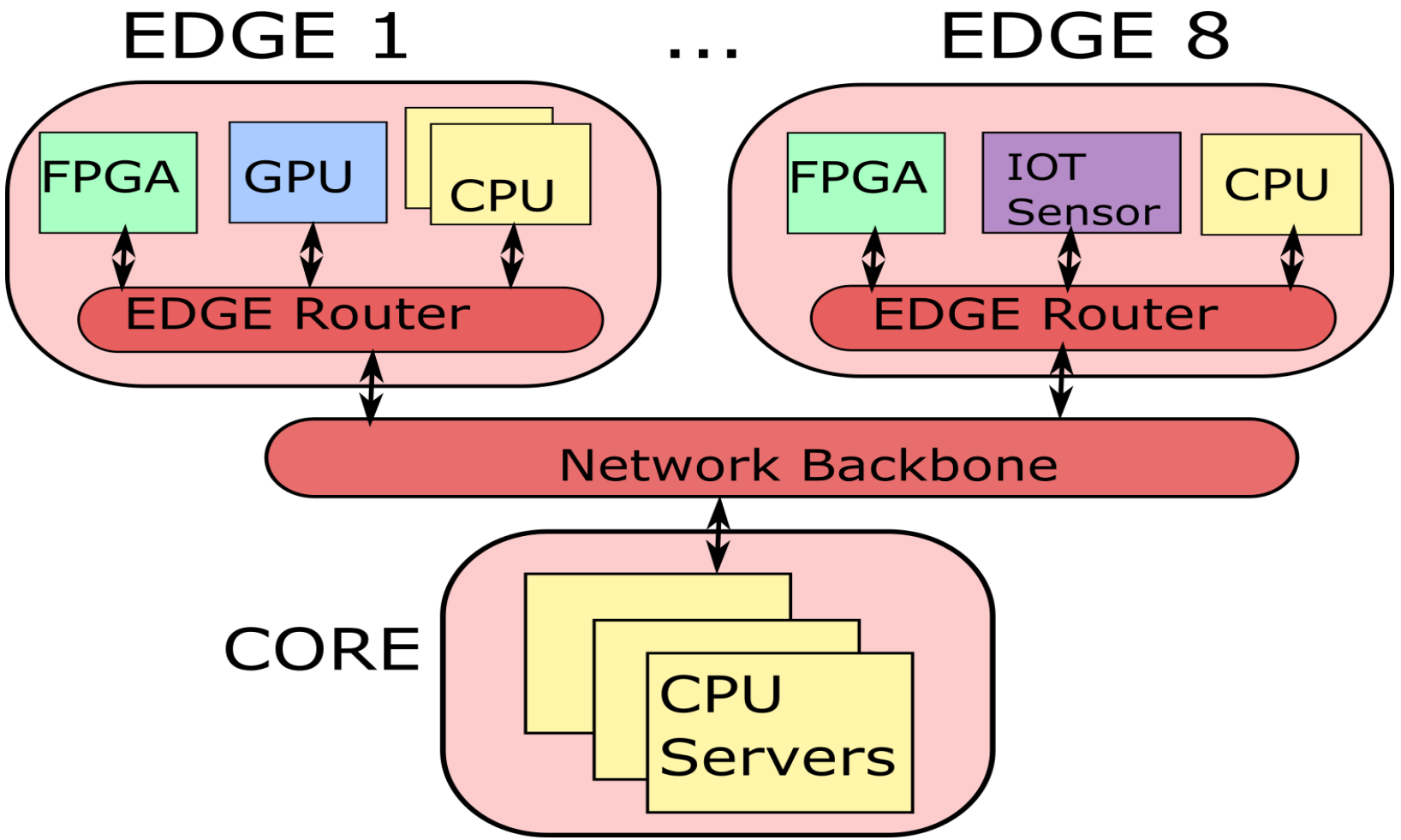
MAC addresses of FPGAs in Multi-FPGA Cluster

FPGA Cluster Generator

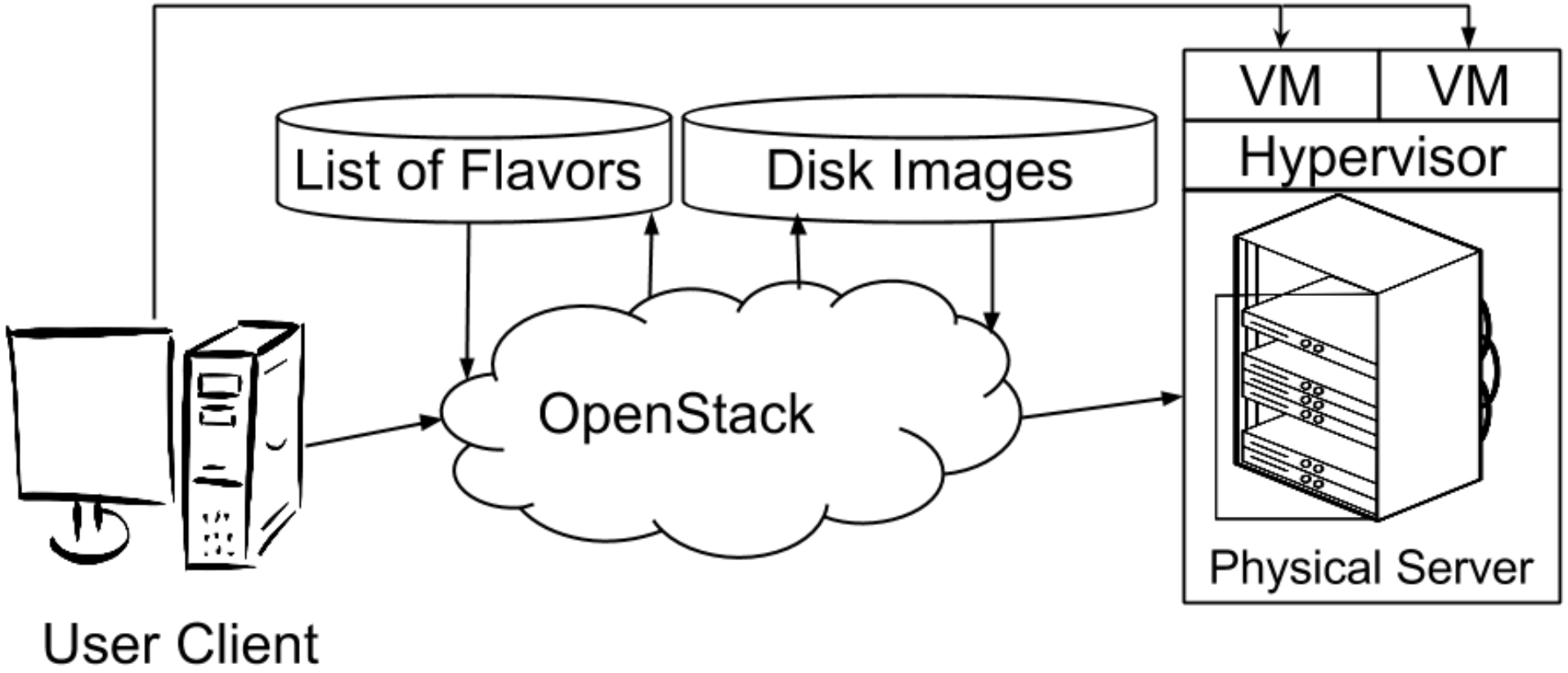
Baseline Infrastructure

- SAVI (Smart Applications on Virtualized Infrastructure)
- OpenStack (Cloud Managing Software)
- Xilinx SDAccel (FPGA Hypervisor)

SAVI (Smart Applications on Virtualized Infrastructure)

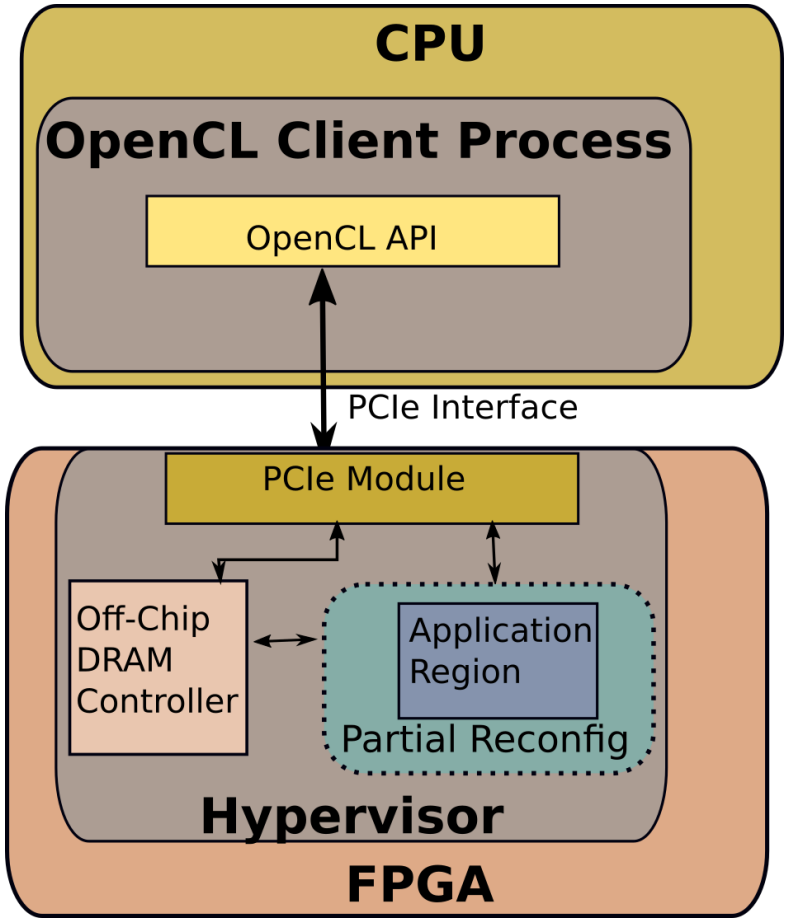


Cloud Managing Software: OpenStack



FPGA Hypervisor: Xilinx SDAccel

- Abstracts physical hardware on FPGA and provides software interface for these modules
- Publicly available through Xilinx
- No network interface



Contributions

1. Non-network FPGA from cloud

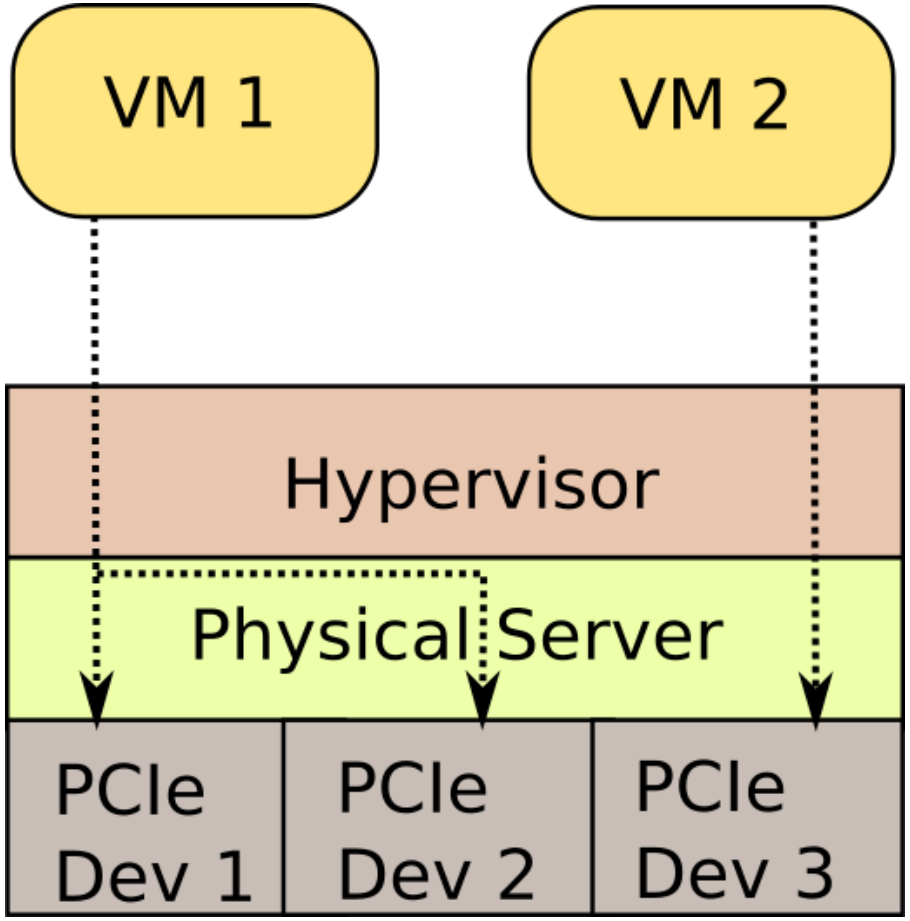
Contributions

1. Non-network FPGA from cloud
2. Networking infrastructure for FPGAs to communicate in heterogeneous network
   - Modified FPGA hypervisor for networking support
   - FPGAs MAC addresses, accessible by any network device in datacenter

Contributions

1. Non-network FPGA from cloud
2. Networking infrastructure for FPGAs to communicate in heterogeneous network
   - Modified FPGA hypervisor for networking support
   - FPGAs MAC addresses, accessible by any network device in datacenter
3. FPGA cluster generator

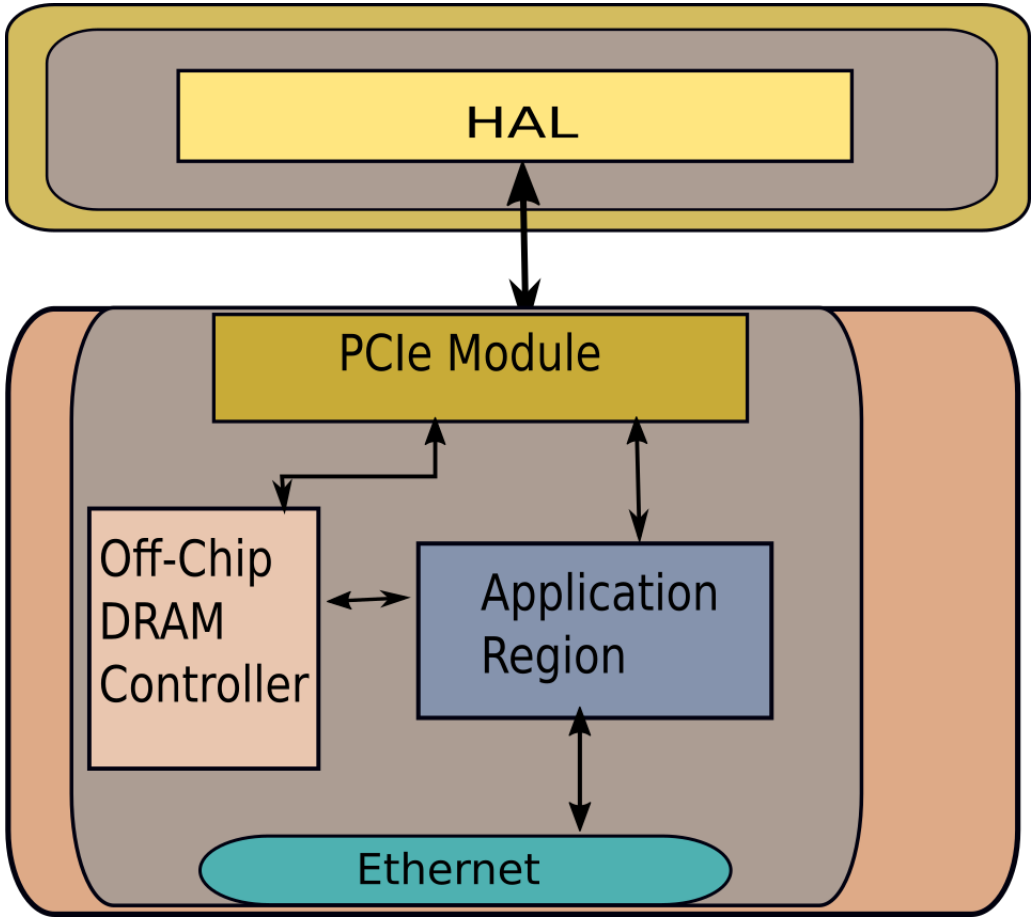
Non-network FPGA from Cloud



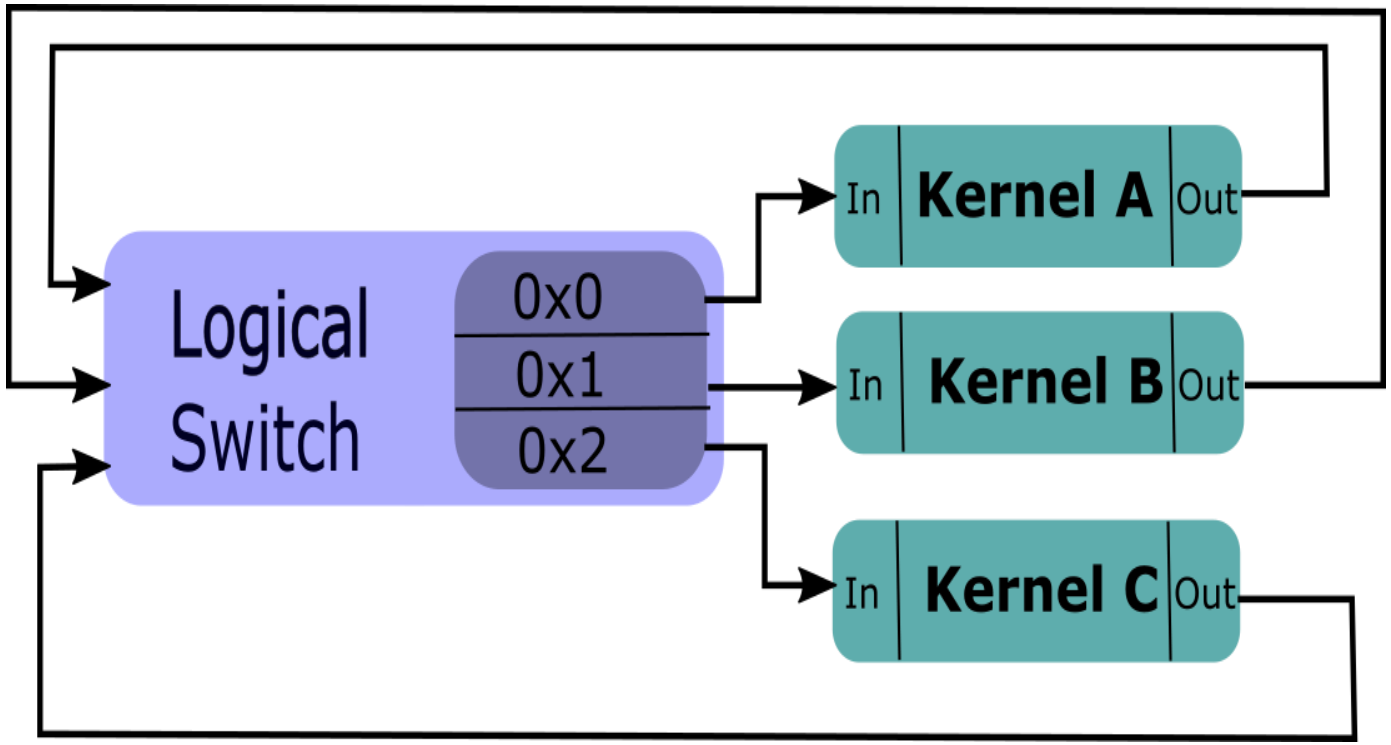
- Deployment Flow
  1. User develops their application on a VM without an FPGA.
  2. Save VM snapshot
  3. Upload VM snapshot to OpenStack
  4. Create new VM with snapshot and FPGA

FPGA Hypervisor: Networking Hypervisor

- Customized shell with:
  - PCIe module
  - Off Chip Memory controller
  - 1 GB Ethernet
- Note: No partial reconfiguration

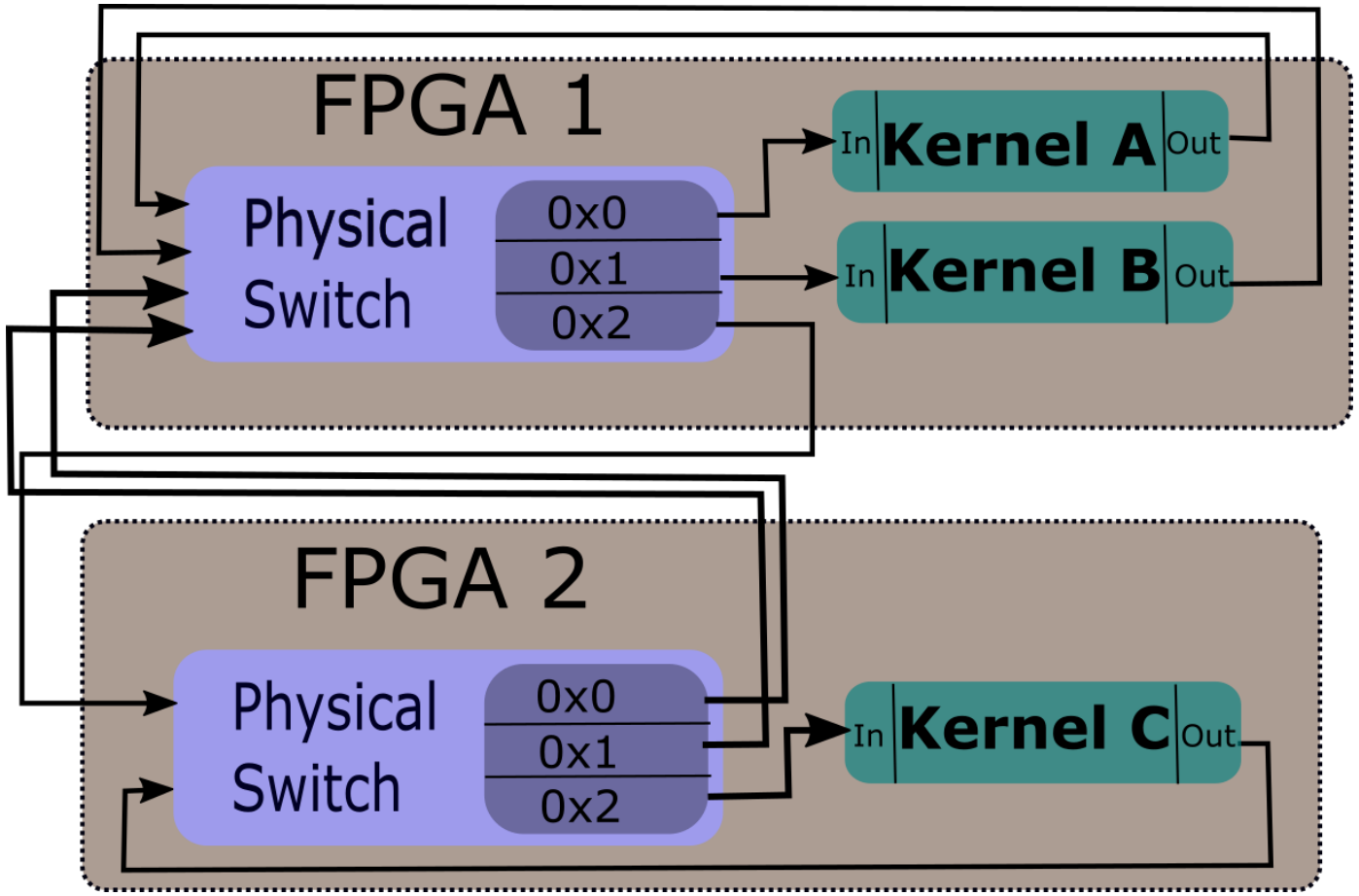


Logical Cluster Description



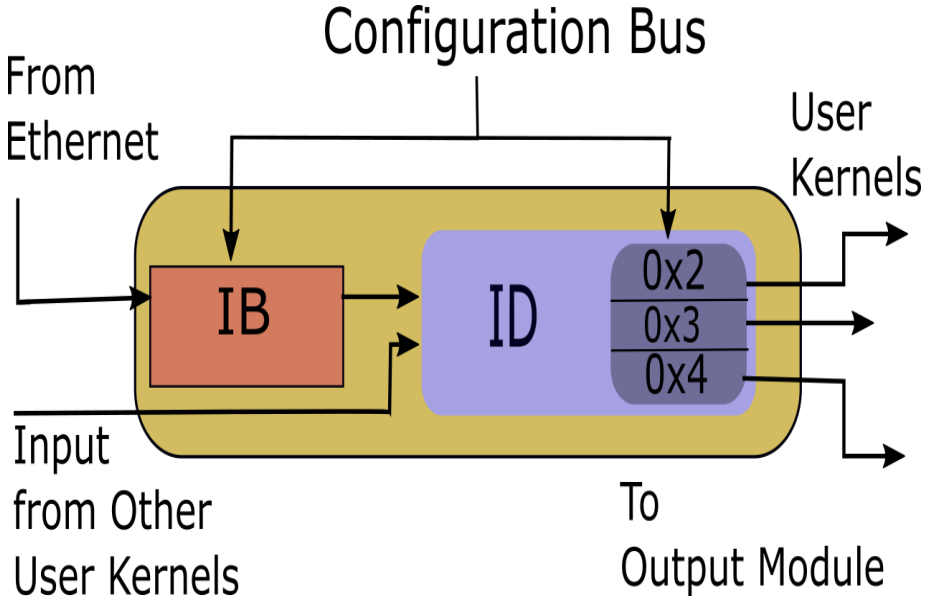
FPGA Mapping File	
Kernel A	FPGA 1
Kernel B	FPGA 1
Kernel C	FPGA 2

Physical Mapping

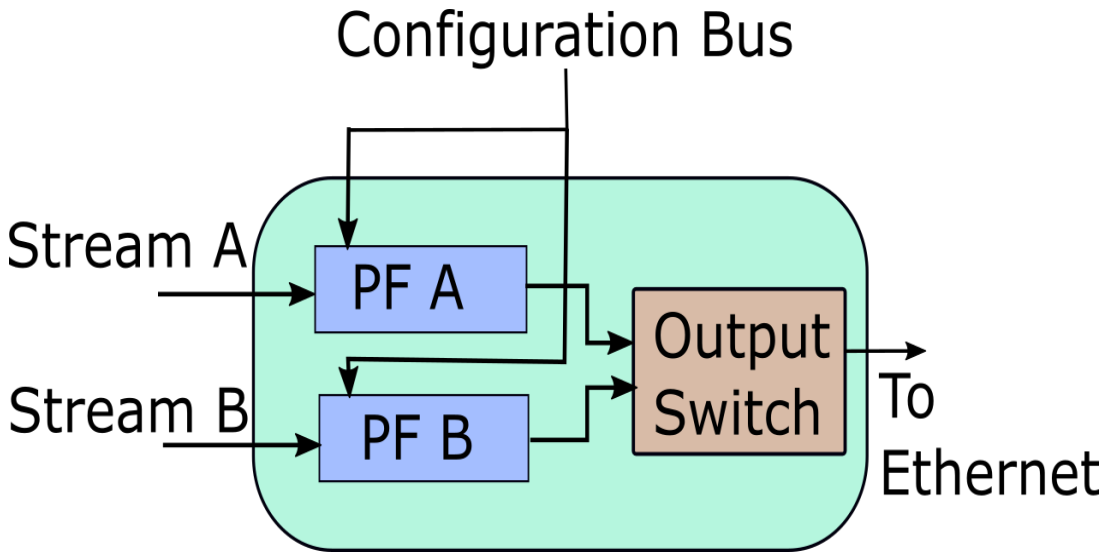


I/O to FPGAs in Cluster

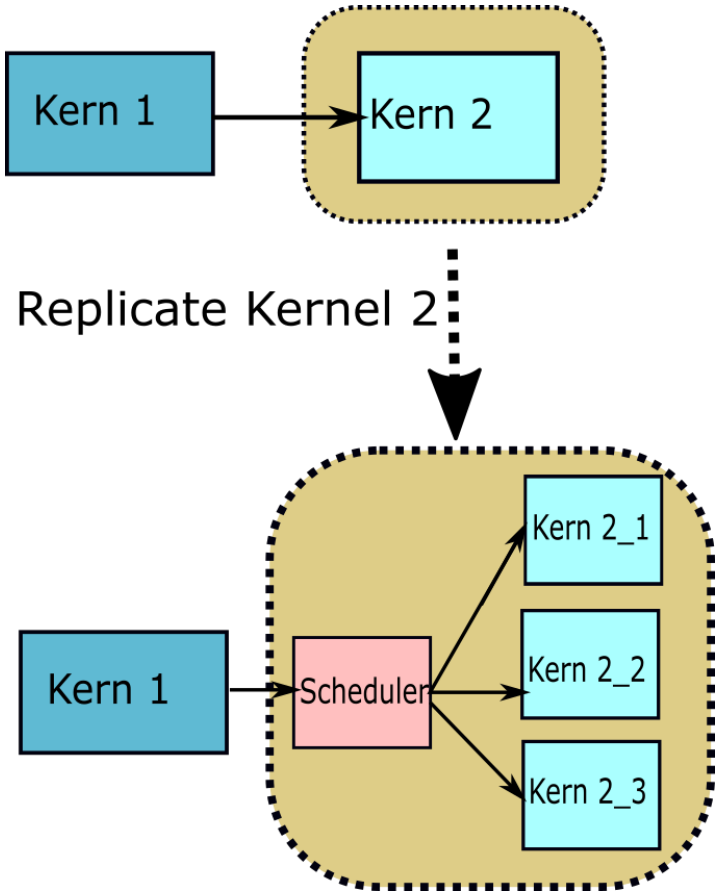
Input

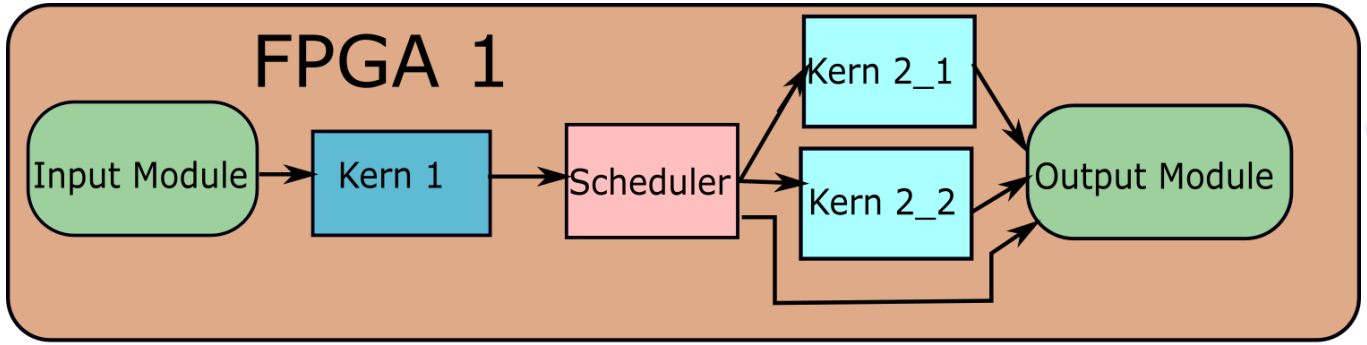


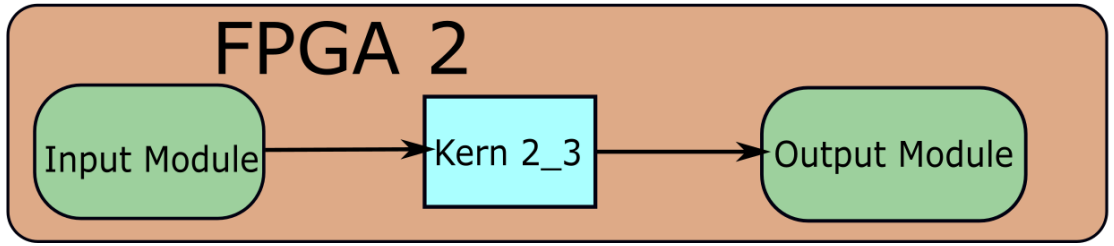
Output



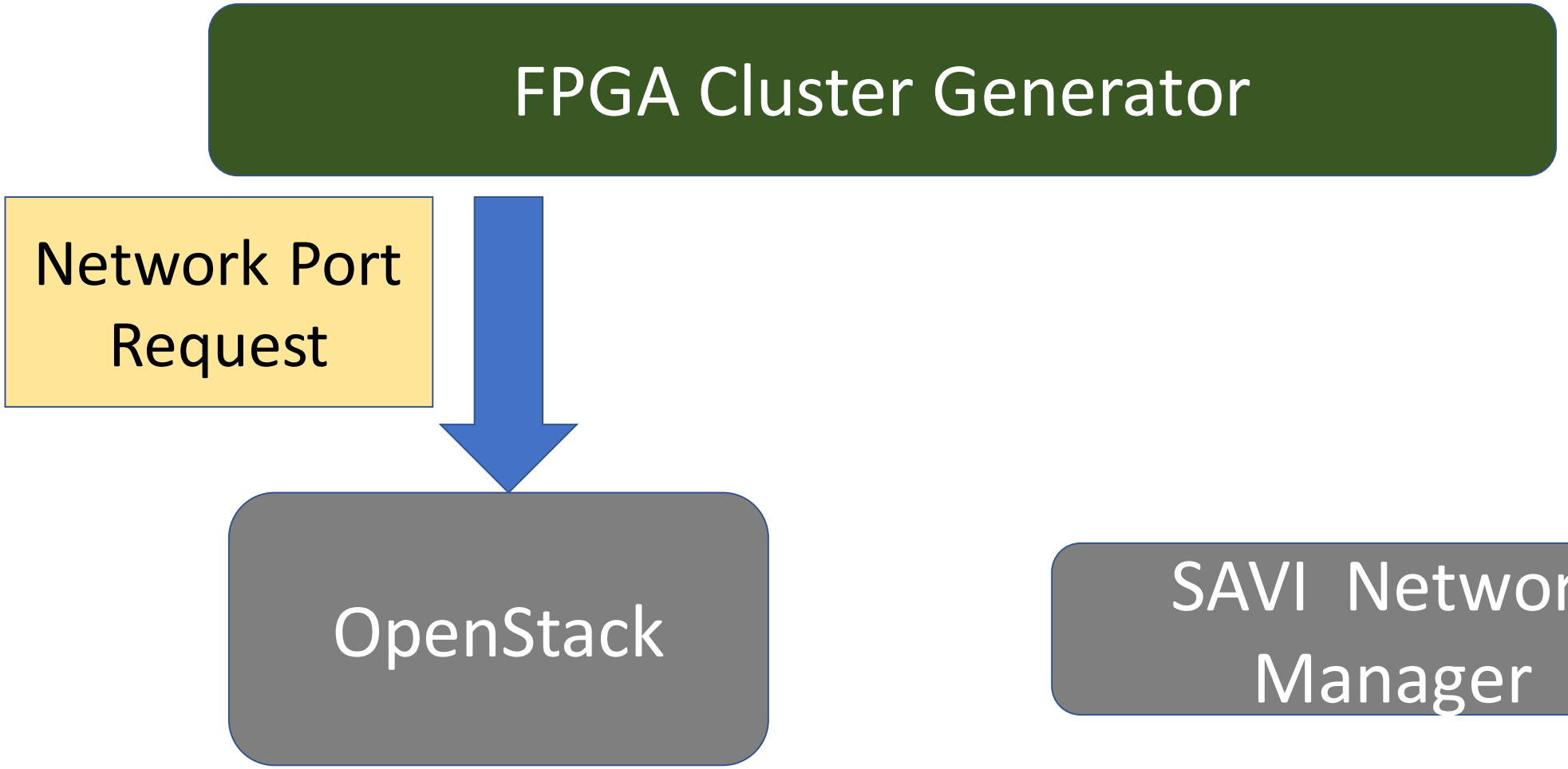
Scaling Up the Clusters







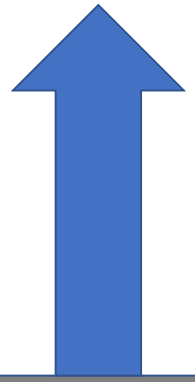
Networking Backend



Networking Backend

FPGA Cluster Generator

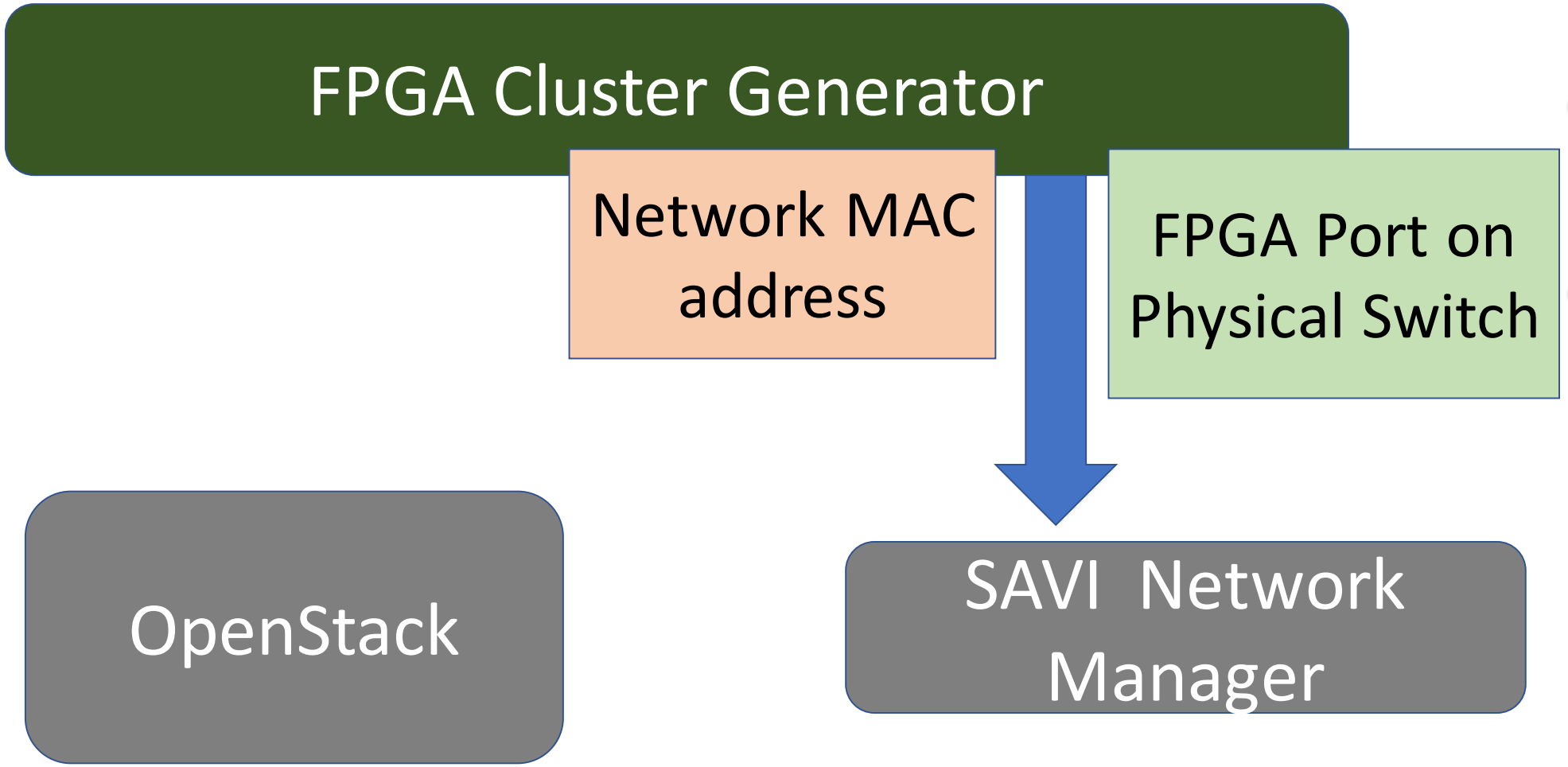
Network MAC address



OpenStack

SAVI Network Manager

Networking Backend

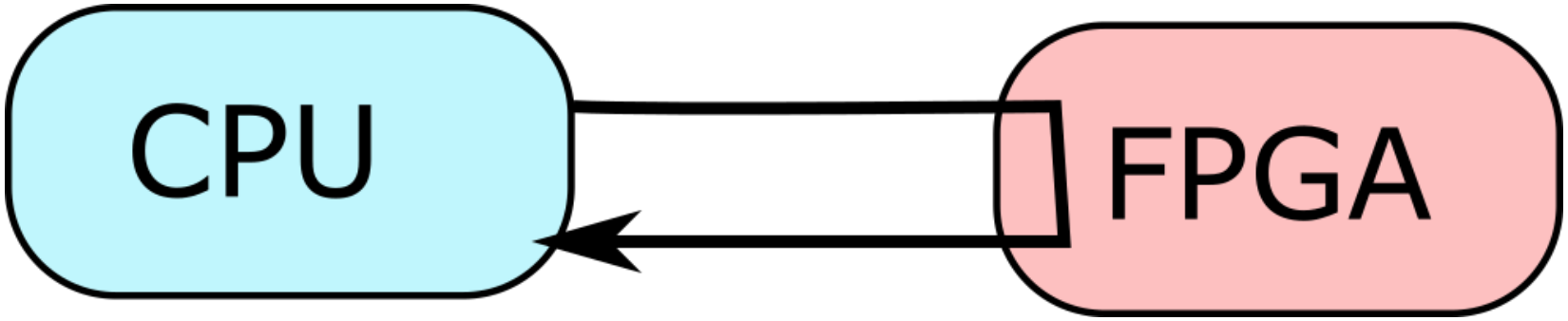


Resource Utilization

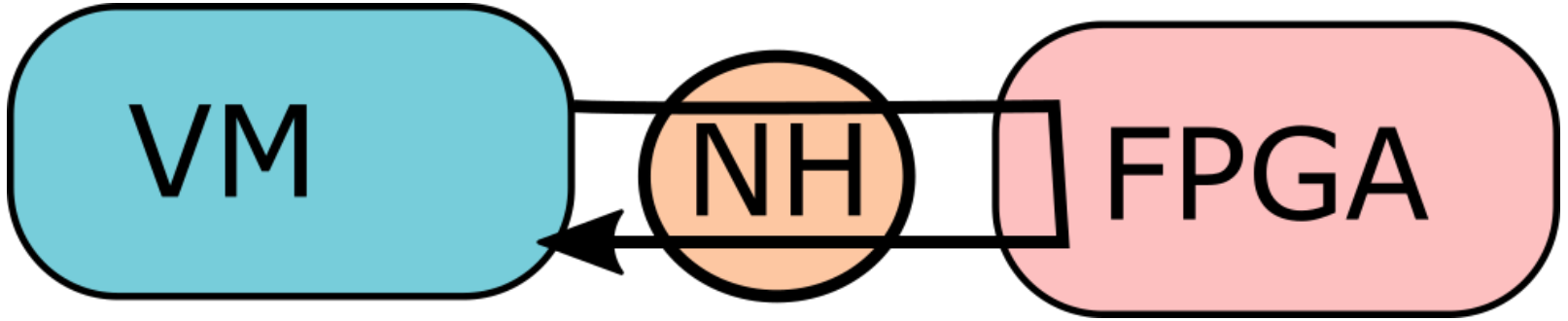
Hardware Setup	LUTS	Flip-Flops	BRAM
SDAccel Base	53346 (12.3 %)	64550 (7.45 %)	228 (15.5 %)
Ethernet Support	8998 (2.1 %)	11574 (1.34 %)	0 (0 %)
Input Module	169 (0.039 %)	294 (0.033 %)	2 (1.36 %)
Output Module	773(0.178 %)	402 (0.059%)	4 (2.72 %)
Total Available	233200	866400	1470

Testing Latency and Throughput

- Directly Connected CPU to FPGA

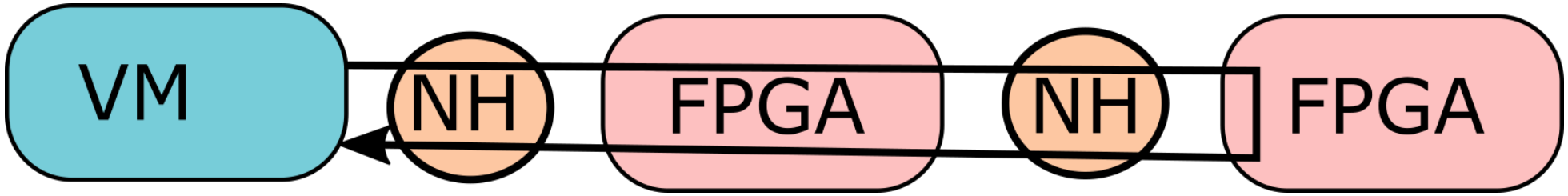


- VM to one FPGA in SAVI

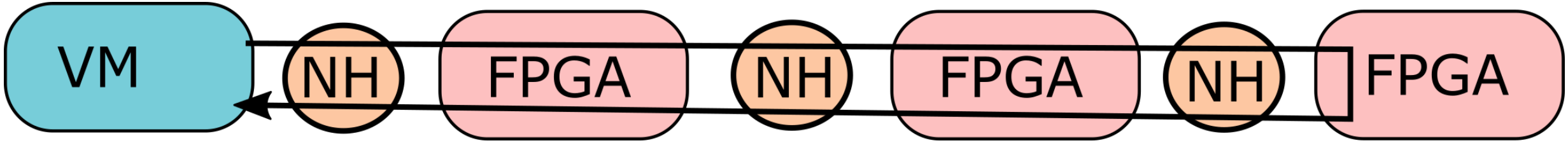


Testing Latency and Throughput

- VM to two FPGA chain in SAVI



- VM to three FPGA chain in SAVI



Round-trip Latency

Test	Latency (ms)
CPU + FPGA	0.0650
VM + 1 FPGA	0.500
VM + 2 FPGA	0.645
VM + 3 FPGA	0.790

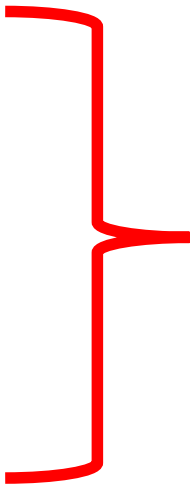
Round-trip Latency

Test	Latency (ms)
CPU + FPGA	0.0650
VM + 1 FPGA	0.500
VM + 2 FPGA	0.645
VM + 3 FPGA	0.790

- CPU > VM
- Extra network Hop

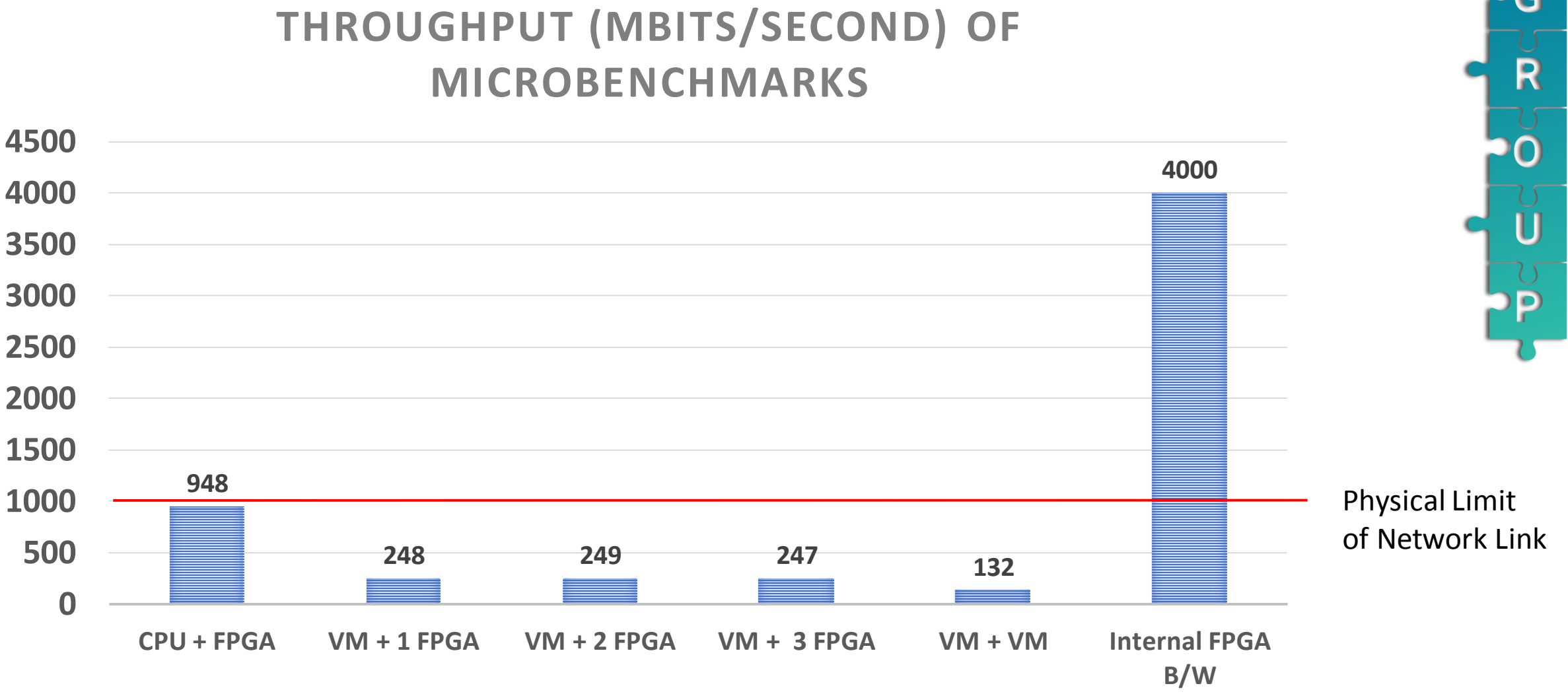
Round-trip Latency

Test	Latency (ms)
CPU + FPGA	0.0650
VM + 1 FPGA	0.500
VM + 2 FPGA	0.645
VM + 3 FPGA	0.790

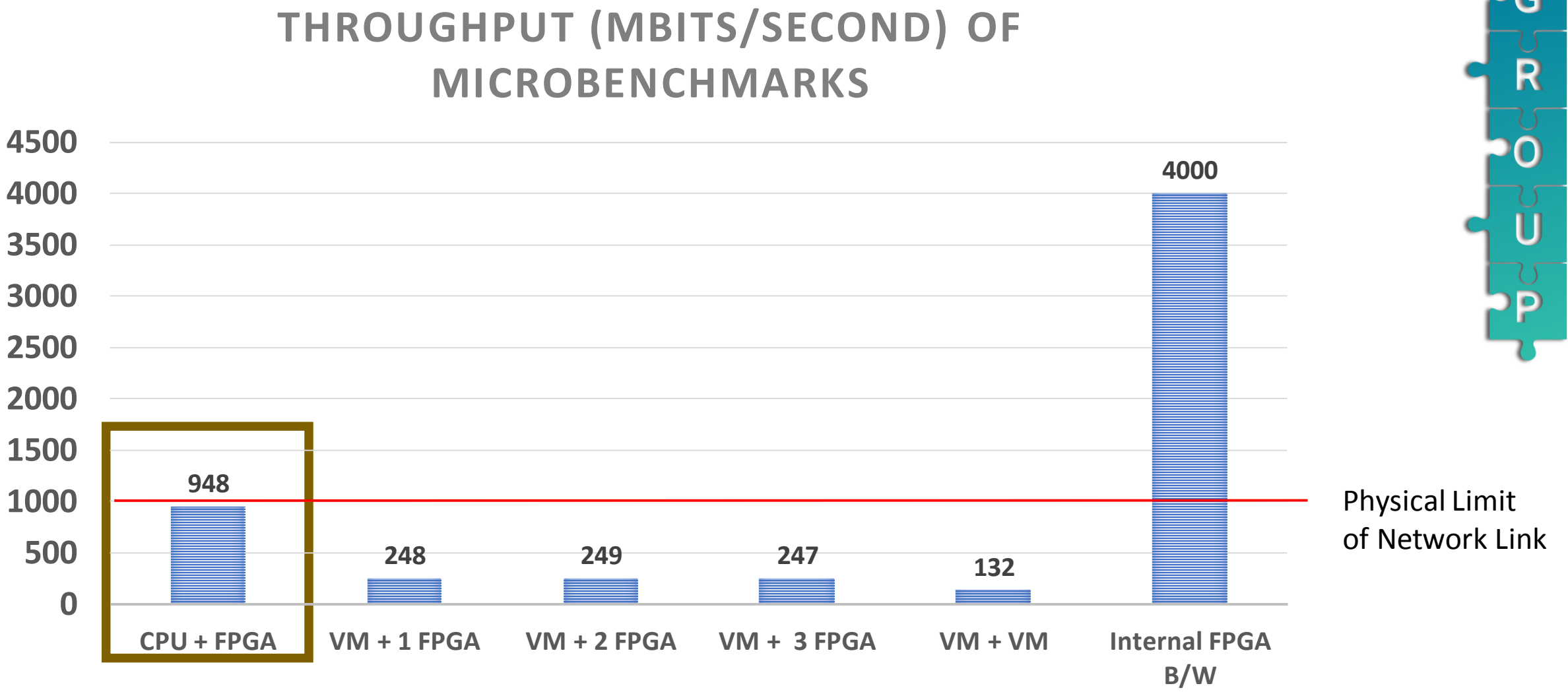


Linear

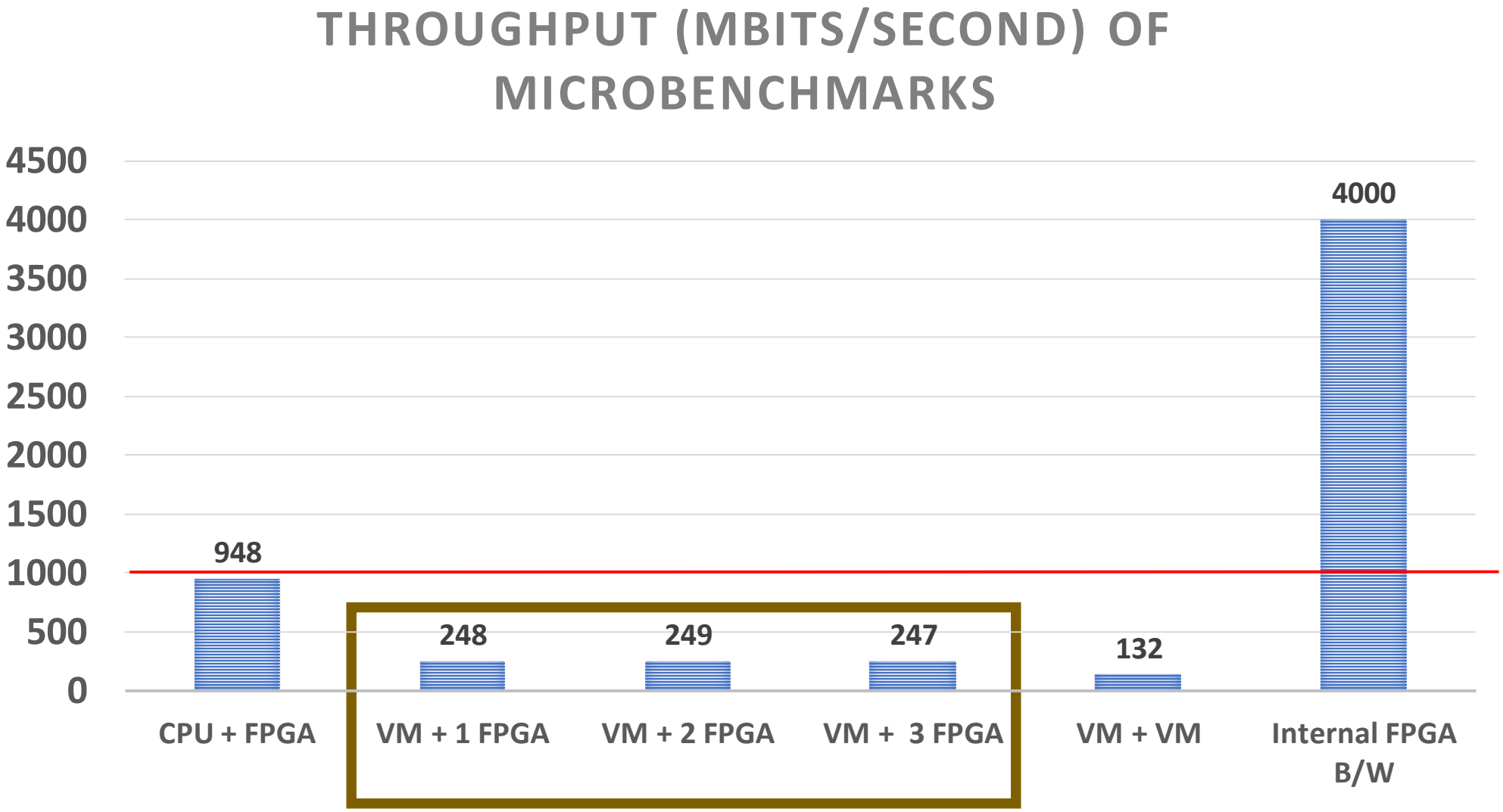
Microbenchmark Throughput



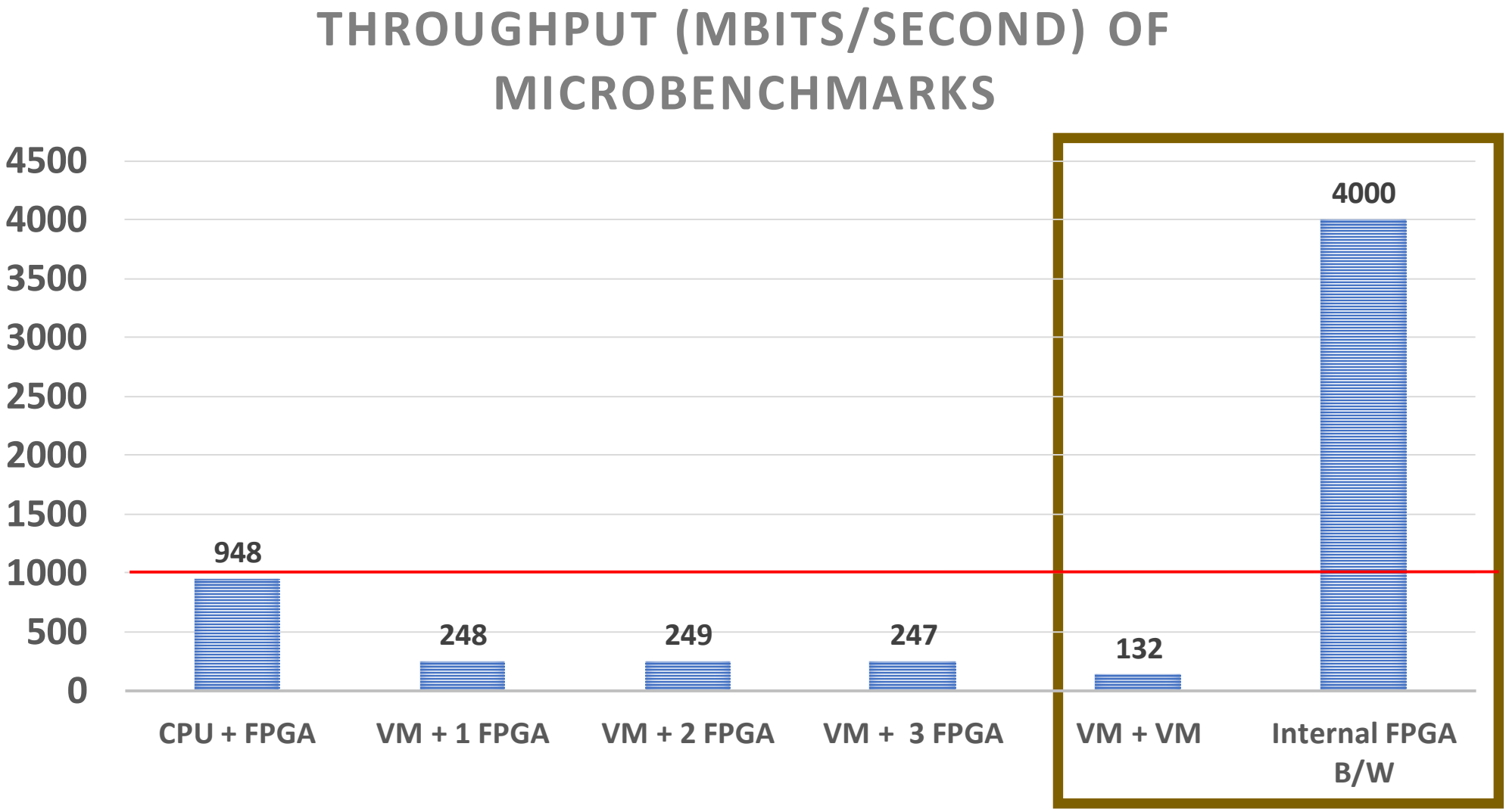
Microbenchmark Throughput



Microbenchmark Throughput

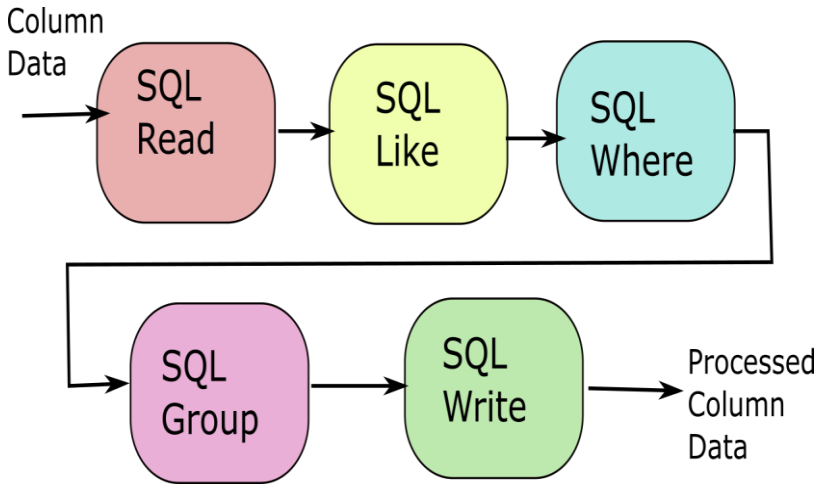


Microbenchmark Throughput

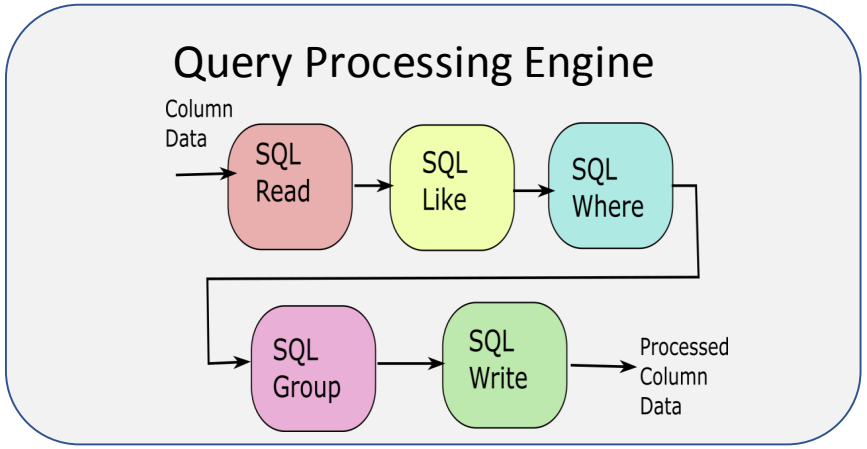


Case Study: Scalability of Query Processing Engine

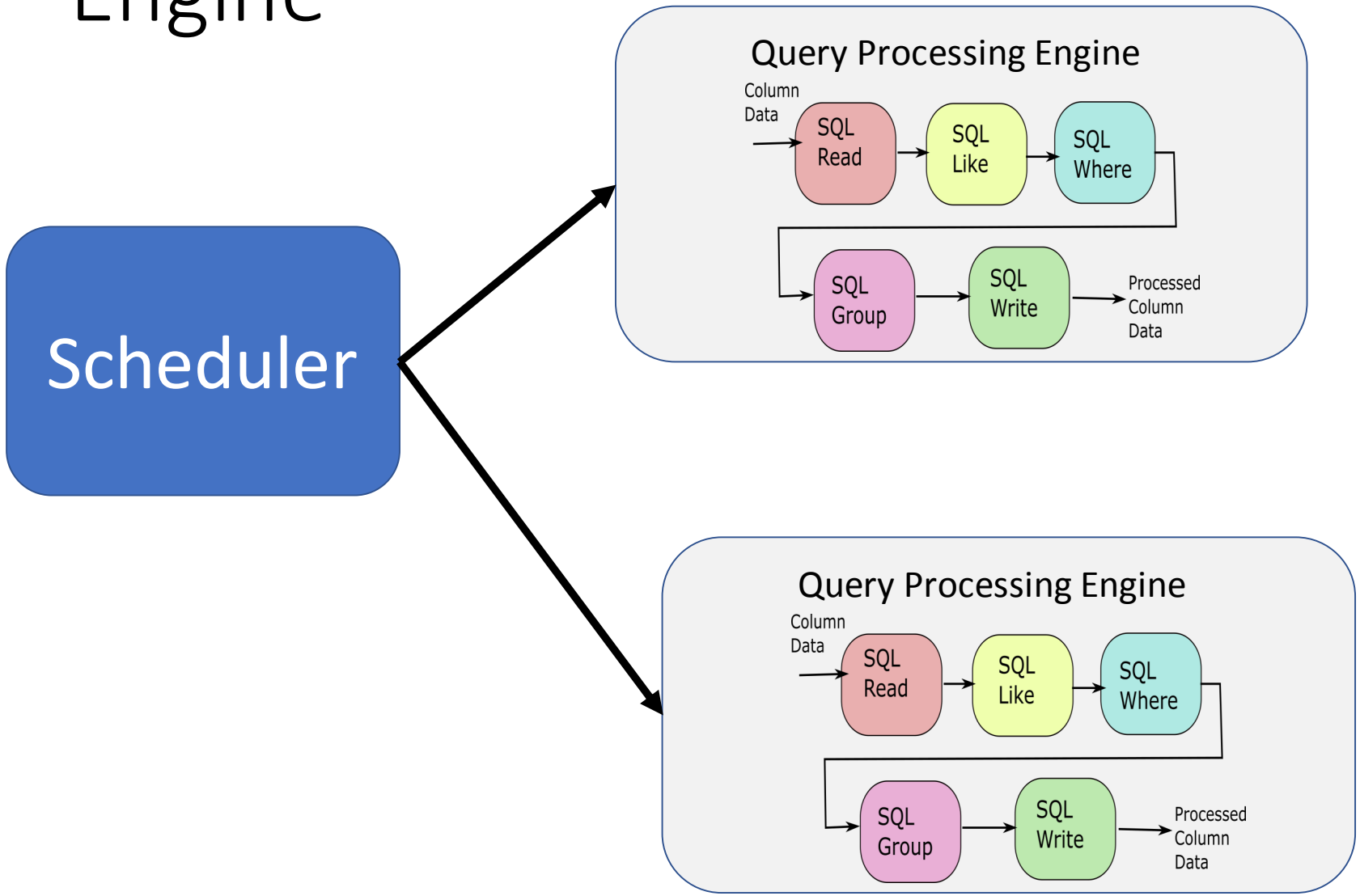
- Representative Case study: Database Streaming Query Processing Engine
  - Size
  - Streaming
- Scalable



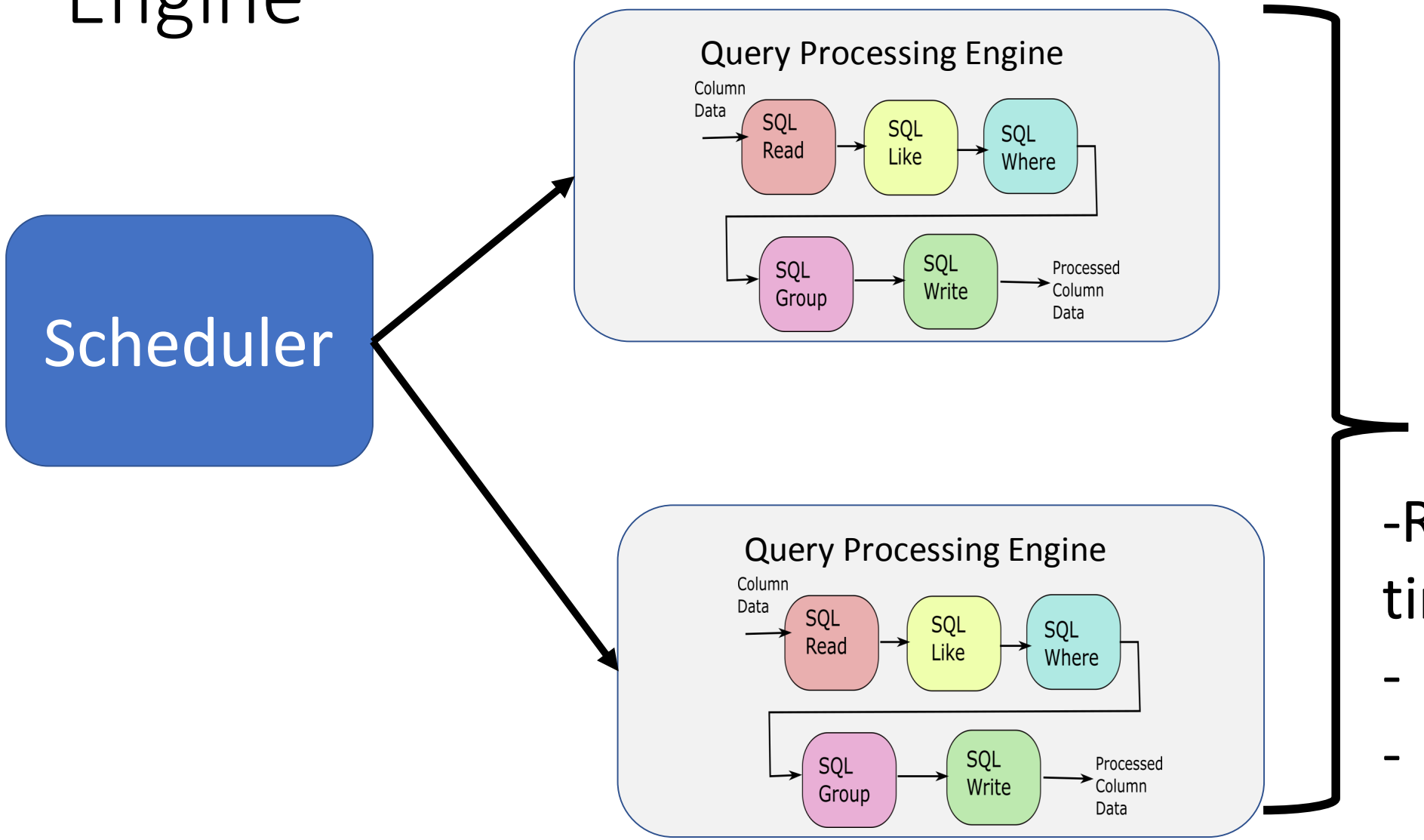
Case Study: Scalability of Query Processing Engine



Case Study: Scalability of Query Processing Engine

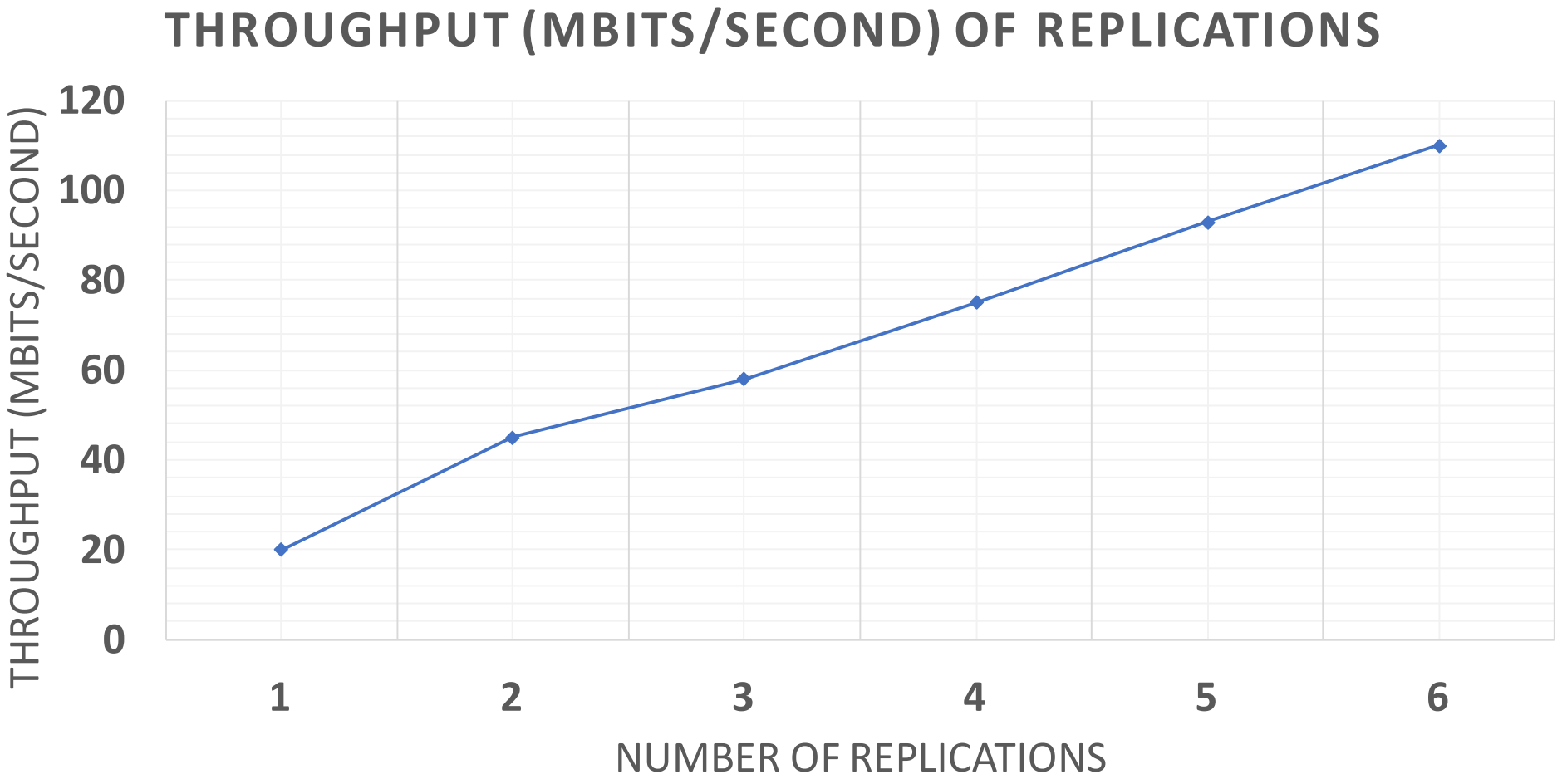


Case Study: Scalability of Query Processing Engine



-Replicated 6 times
- 3 FPGAs
- 2 units /FPGA

Case Study: Scalability of Query Processing Engine



Conclusion and Summary

- Users can create elastic FPGA clusters from cloud easily
  - Inter-FPGA fabric automatically generated
  - FPGAs provided network interface
- Little overhead
- Easy to scale

Future Work

- Infrastructure Upgrade
  - 10G
  - Partial Reconfiguration

Future Work

- Infrastructure Upgrade
  - 10G
  - Partial Reconfiguration
- Automatic Partitioning/Scheduling
  - HLS Model (Scheduler): Behavioral
  - Circuit Partitioning

Future Work

- Infrastructure Upgrade
  - 10G
  - Partial Reconfiguration
- Automatic Partitioning/Scheduling
  - HLS Model (Scheduler): Behavioral
  - Circuit Partitioning
- Debugging of Large clusters
  - Combine individual debug environments
  - Monitor health

Future Work

- Infrastructure Upgrade
  - 10G
  - Partial Reconfiguration
- Automatic Partitioning/Scheduling
  - HLS Model (Scheduler): Behavioral
  - Circuit Partitioning
- Debugging of Large clusters
  - Combine individual debug environments
  - Monitor health
- Large Scale Applications
  - Networking Applications (NFV)
  - Distributed Applications (Web-search)
  - Heterogeneous IOT Applications

Thank You

Questions?

Email: naif.tarafdar@mail.utoronto.ca