Janus: Enabling Expressive and Efficient ACLs in High-speed RDMA Clouds

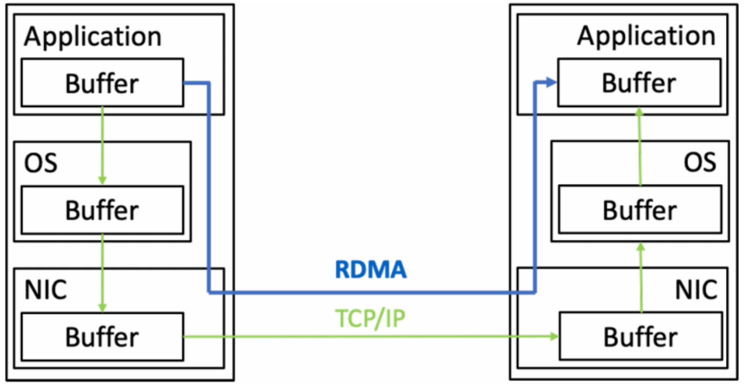
Ziteng Chen, Menghao Zhang, Jiahao Cao, Xuzheng Chen,
Qiyang Peng, Shicheng Wang, Guanyu Li, Mingwei Xu

Outline

- Background and Motivation
- Janus Design
  - System overview
  - Tailored ACL expression
  - DPU-based ACL enforcement
- Evaluation

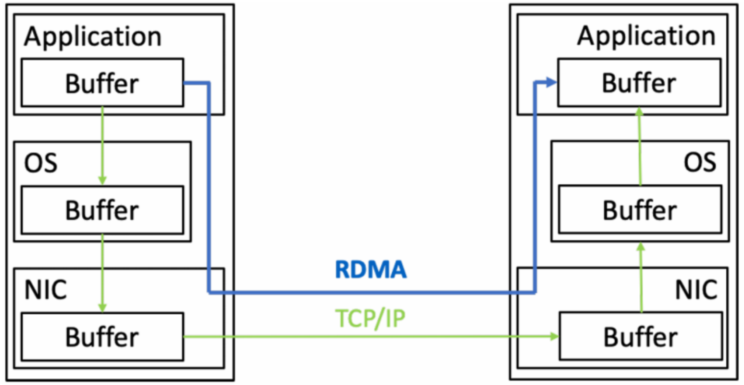
RDMA Networks

- RDMA: A Kernel-bypass Technology
  - Direct access to remote host memory without CPU
    - High bandwidth: SOTA RNICs can support 400Gbps
    - Low latency: <5us



RDMA Networks

- RDMA: A Kernel-bypass Technology
  - Direct access to remote host memory without CPU
    - High bandwidth: SOTA RNICs can support 400Gbps
    - Low latency: <5us



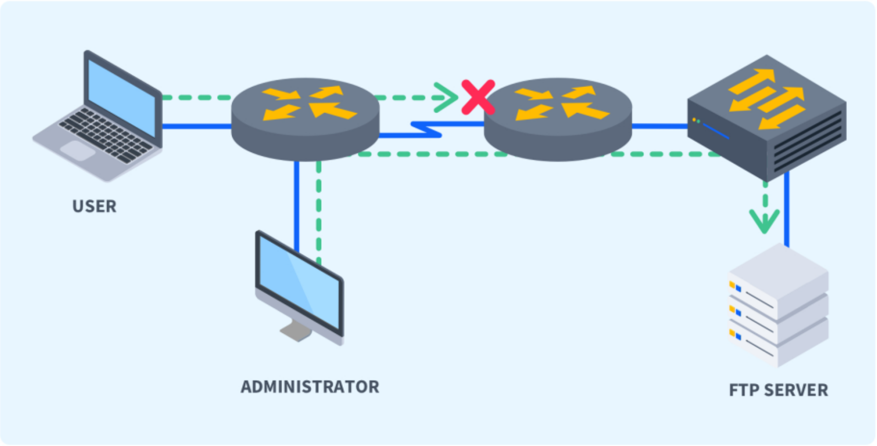
- Application Scenarios
  - LLM training and inference
  - Distributed cloud storage
  - Search query

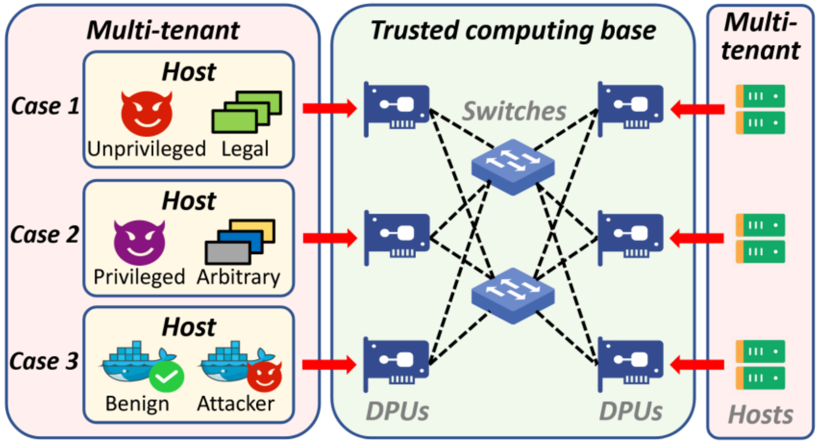
Microsoft Meta amazon Google NVIDIA Alibaba Cloud

- RDMA Clouds are Getting Prevalence
  - Tenants share high-bandwidth and low-latency RDMA networks

Access Control Lists for RDMA Clouds

- RDMA Clouds Need ACLs
  - Per-packet inspection to regulate traffic
    - Intra-tenant: protect sensitive services and applications
    - Inter-tenant: block cross-tenant communication
  - Govern unauthorized accesses
    - Allow operators to make policies towards target nodes

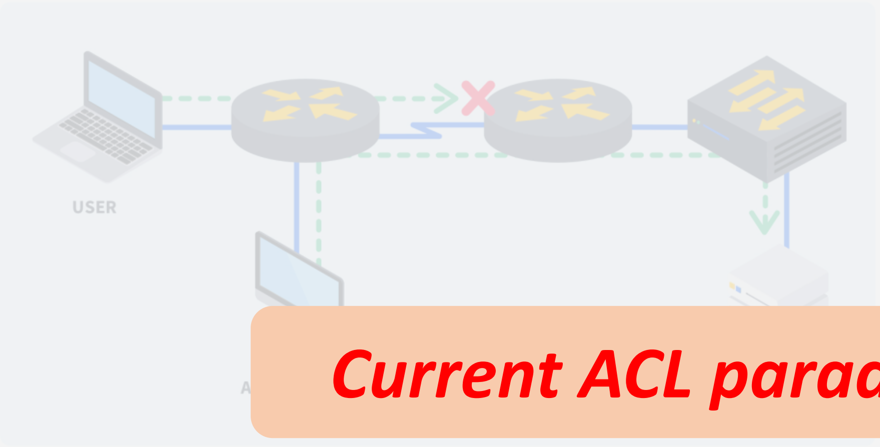


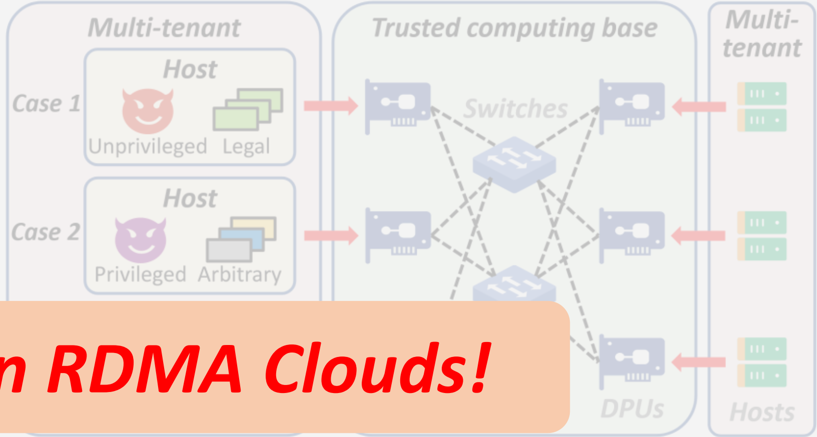


[1] https://www.cbtnuggets.com/blog/technology/networking/what-is-access-control-list

Access Control Lists for RDMA Clouds

- RDMA Clouds Need ACLs
  - Per-packet inspection to regulate traffic
    - Intra-tenant: protect sensitive services and applications
    - Inter-tenant: block cross-tenant communication
  - Govern unauthorized accesses
    - Allow operators to make policies towards target nodes



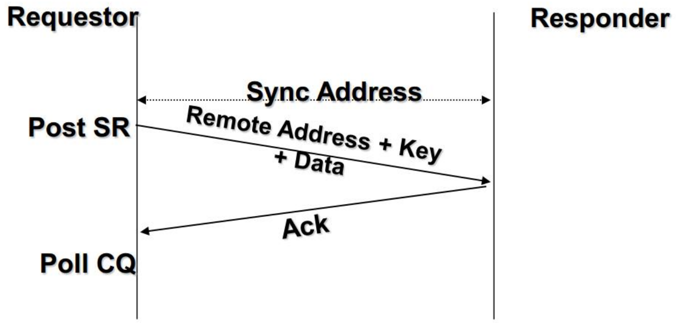


Current ACL paradigm fails in RDMA Clouds!

1. https://www.cbtnuggets.com/blog/technology/networking/what-is-access-control-list

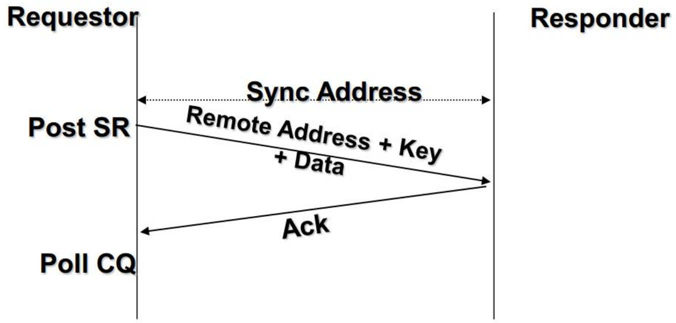
Current ACLs Fail in RDMA Clouds (1)

- Five-tuple Expressions Cannot Represent QP Semantics
  - Intricate operations and disaggregated traffic
    - Control path: QP state maintenance
    - Data path: application data transmission

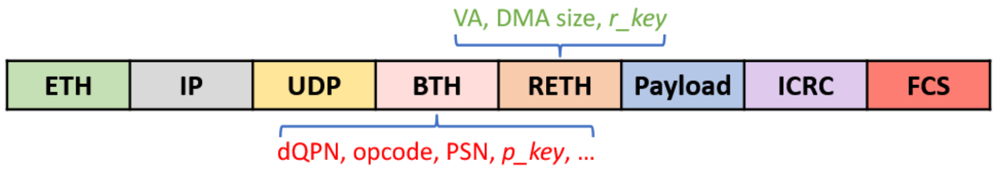


Current ACLs Fail in RDMA Clouds (1)

- Five-tuple Expressions Cannot Represent QP Semantics
  - – Intricate operations and disaggregated traffic
    - Control path: QP state maintenance
    - Data path: application data transmission

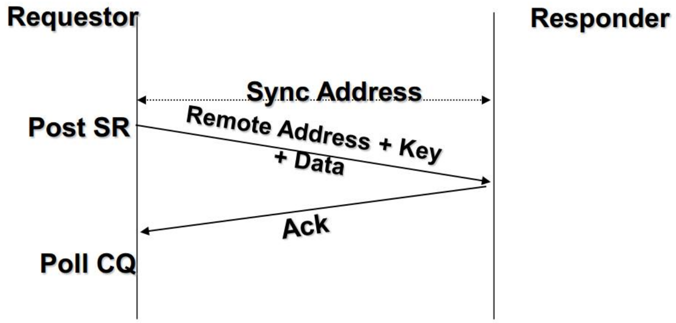


  - – Complex and fine-grained metadata in RDMA packets
    - QP-semantics fields: QP number, operation code, memory address

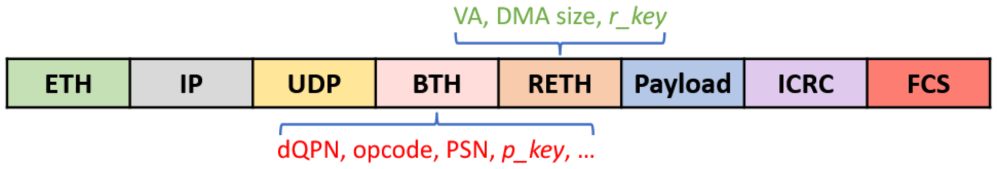


Current ACLs Fail in RDMA Clouds (1)

- Five-tuple Expressions Cannot Represent QP Semantics
  - Intricate operations and disaggregated traffic
    - Control path: QP state maintenance
    - Data path: application data transmission


Requestor
Responder
Sync Address
Post SR
Remote Address + Key + Data
Ack
Poll CQ

  - Complex and fine-grained metadata in RDMA packets
    - QP-semantics fields: QP number, operation code, memory address
  - TCP/IP expressions fail in RDMA scenario
    - Cannot describe QP semantics and operations
    - Cannot resolve RDMA-specific fields


VA, DMA size, *r_key*
ETH | IP | UDP | BTH | RETH | Payload | ICRC | FCS
dQPN, opcode, PSN, *p_key*, …

Current ACLs Fail in RDMA Clouds (2)

- Lack Comprehensive and Efficient Coverage on RDMA Traffic
  - Software-based enforcer
    - Iptables and open vSwitch: at kernel
      - Cannot capture kernel-bypass traffic
    - Snap [1] and FreeFlow [2]: at shim-layer
      - Cannot deliver line-rate throughput and ultra-low latency

1. Snap: a microkernel approach to host networking, SOSP 19
2. FreeFlow: Software-based Virtual RDMA Networking for Containerized Clouds, NSDI 19
3. Bedrock: Programmable Network Support for Secure RDMA Systems, Security 21

Current ACLs Fail in RDMA Clouds (2)

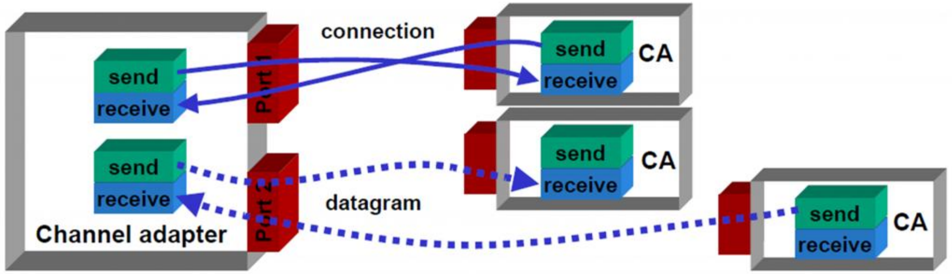
- Lack Comprehensive and Efficient Coverage on RDMA Traffic
  - Software-based enforcer
    - Iptables and open vSwitch: at kernel
      - Cannot capture kernel-bypass traffic
    - Snap [1] and FreeFlow [2]: at shim-layer
      - Cannot deliver line-rate throughput and ultra-low latency
  - Hardware-based enforcer
    - Bedrock [3]: at programmable switches
      - Introduce extra latency for intra-host traffic
      - Lack regulation for control path traffic

1. Snap: a microkernel approach to host networking, SOSP 19
2. FreeFlow: Software-based Virtual RDMA Networking for Containerized Clouds, NSDI 19
3. Bedrock: Programmable Network Support for Secure RDMA Systems, Security 21

RDMA Clouds Need New ACL Paradigm

Coverage

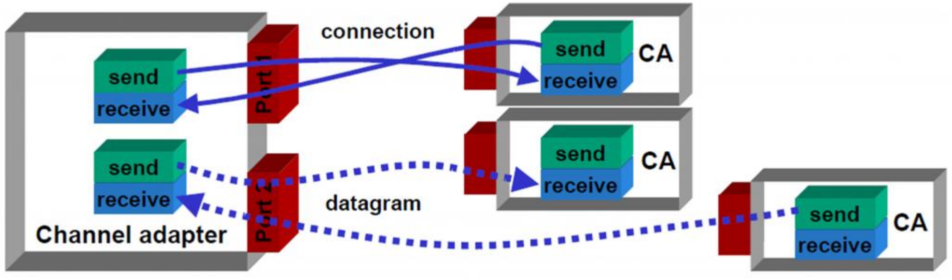
1. Describe QP semantics
2. Cover disaggregated traffic



RDMA Clouds Need New ACL Paradigm

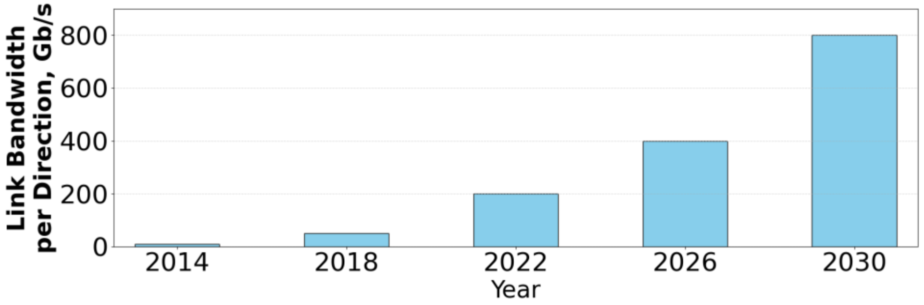
Coverage

1. Describe QP semantics
2. Cover disaggregated traffic



Performance

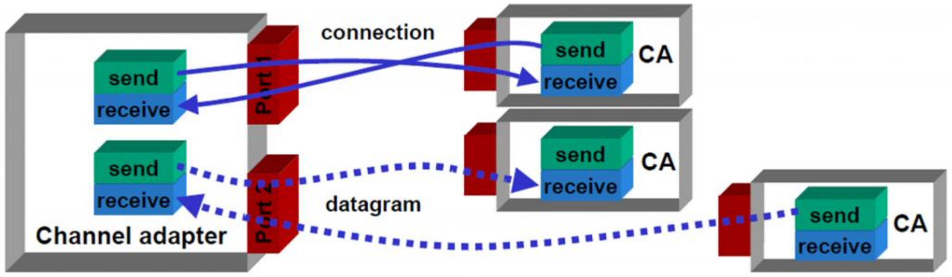
1. Line-rate throughput
2. Ultra-low latency
3. Low overhead



RDMA Clouds Need New ACL Paradigm

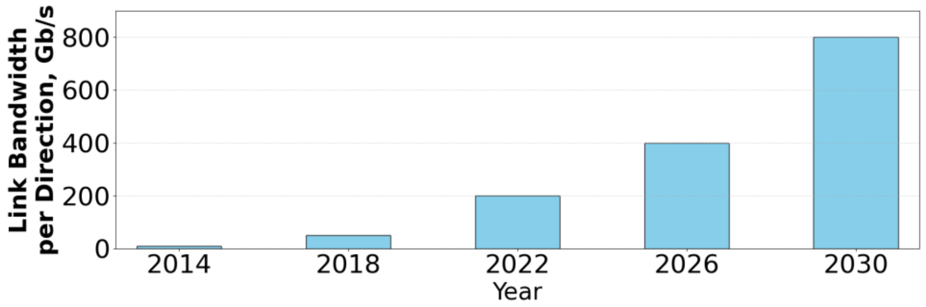
Coverage

1. Describe QP semantics
2. Cover disaggregated traffic



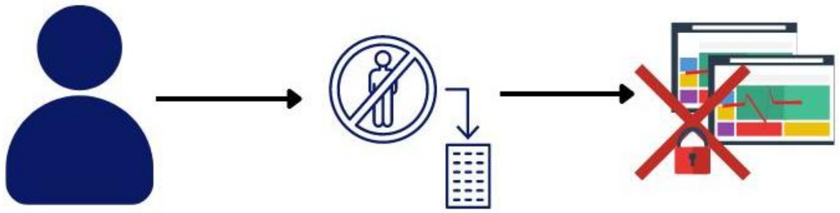
Performance

1. Line-rate throughput
2. Ultra-low latency
3. Low overhead



Usability

1. Transparent to tenants
2. Friendly interfaces

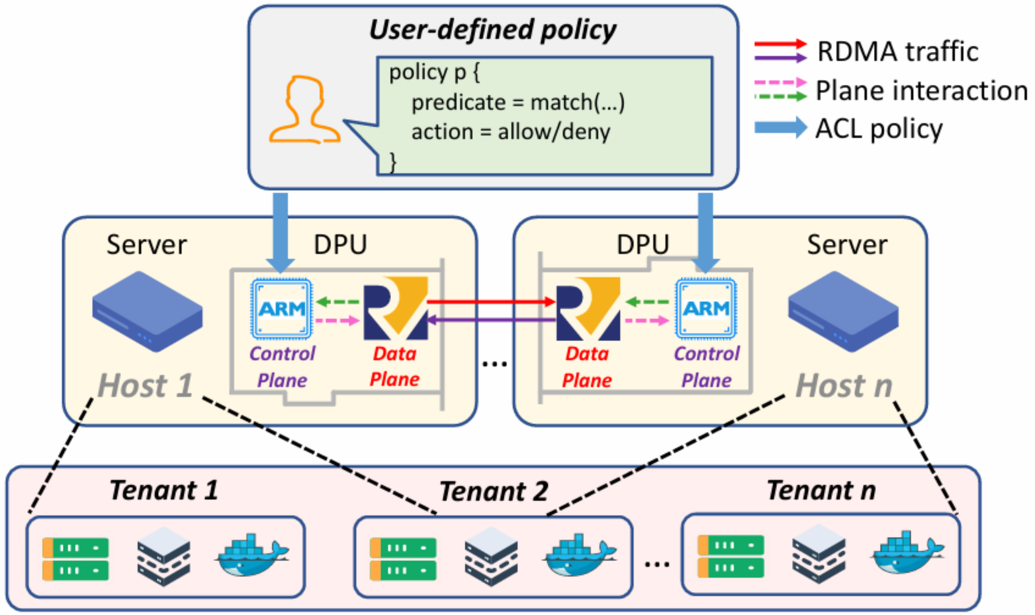


Outline

- Background and Motivation
- Janus Design
  - System overview
  - Tailored ACL expression
  - DPU-based ACL enforcement
- Evaluation

System Overview

- Janus Overview
  - Operators make ACL policies at controller
    - Use Janus expressions with QP semantics
  - End-host DPUs enforce ACL function
    - Control plane for policy maintenance; data plane for per-packet inspection



Outline

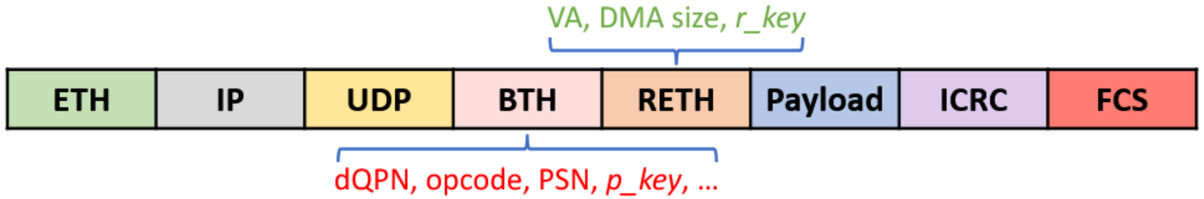
- Background and Motivation
- Janus Design
  - System overview
  - Tailored ACL expression
  - DPU-based ACL enforcement
- Evaluation

Expression Challenges

- Challenge 1: Which Packet Fields are Necessary?
  - Basic transmission unit of RDMA
    - How sender QP operates on destination QP’s memory region

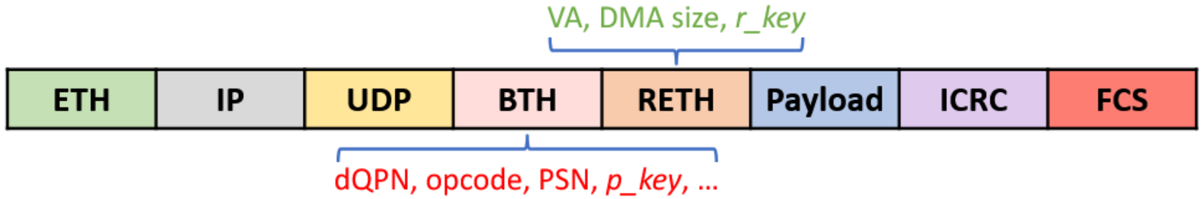
Expression Challenges

- Challenge 1: Which Packet Fields are Necessary?
  - Basic transmission unit of RDMA
    - How sender QP operates on destination QP's memory region
  - Identity the minimum but necessary packet fields
    - Describe QP semantics of RDMA
    - Exclude irrelevant fields



Expression Challenges

- Challenge 1: Which Packet Fields are Necessary?
  - – Basic transmission unit of RDMA
    - How sender QP operates on destination QP’s memory region
  - – Identity the minimum but necessary packet fields
    - Describe QP semantics of RDMA
    - Exclude irrelevant fields



- Challenge 2: Relax Policy-making Difficulty
  - – Finer-grained dimensions than five-tuples
    - More packet fields and traffic paths
  - – How to make intricate policies more easily

Janus Expressions

- Objective: Identify Minimum but Necessary Packet Fields
  - Necessary fields
    - Network entities to imply a node’s network-layer identity

Janus Expressions

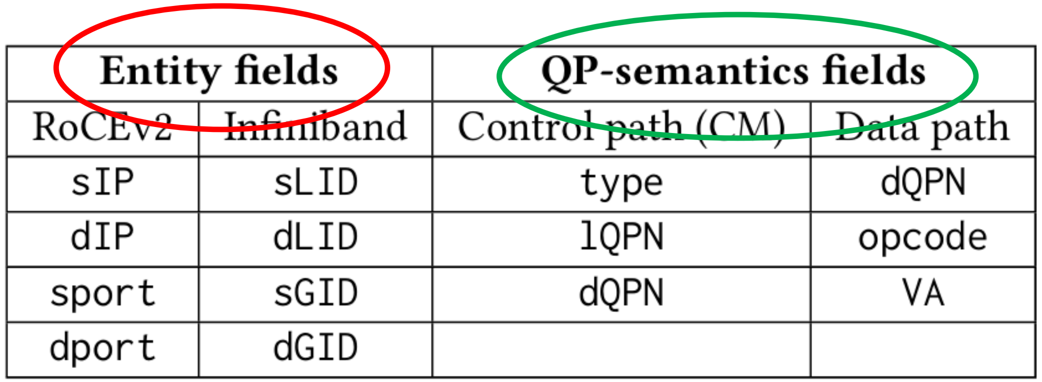
- Objective: Identify Minimum but Necessary Packet Fields
  - Necessary fields
    - Network entities to imply a node’s network-layer identity
    - QP operations
      - Control path: QP creation and destroy
      - Data path: reads/writes/atomics on remote memory

Janus Expressions

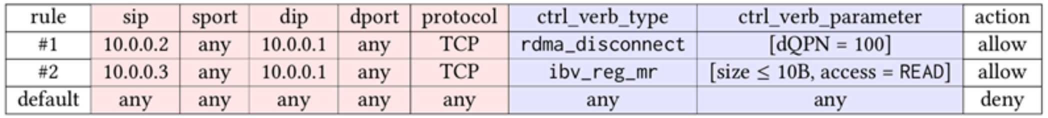
- Objective: Identify Minimum but Necessary Packet Fields
  - Necessary fields
    - Network entities to imply a node’s network-layer identity
    - QP operations
      - Control path: QP creation and destroy
      - Data path: reads/writes/atomics on remote memory
  - Irrelevant fields
    - Application-layer: packet sequence number, rkey, DMA length…

Janus Expressions

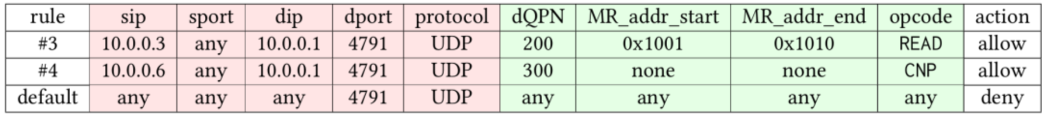
- Objective: Identify Minimum but Necessary Packet Fields
  - Necessary fields
    - Network entities to imply a node’s network-layer identity
    - QP operations
      - Control path: QP creation and destroy
      - Data path: reads/writes/atomics on remote memory
  - Irrelevant fields
    - Application-layer: packet sequence number, rkey, DMA length…



Entity fields		QP-semantics fields	
RoCEv2	Infiniband	Control path (CM)	Data path
sIP	sLID	type	dQPN
dIP	dLID	lQPN	opcode
sport	sGID	dQPN	VA
dport	dGID		



rule	sip	sport	dip	dport	protocol	ctrl_verb_type	ctrl_verb_parameter	action
#1	10.0.0.2	any	10.0.0.1	any	TCP	rdma_disconnect	[dQPN = 100]	allow
#2	10.0.0.3	any	10.0.0.1	any	TCP	ibv_reg_mr	[size ≤ 10B, access = READ]	allow
default	any	any	any	any	any	any	any	deny



rule	sip	sport	dip	dport	protocol	dQPN	MR_addr_start	MR_addr_end	opcode	action
#3	10.0.0.3	any	10.0.0.1	4791	UDP	200	0x1001	0x1010	READ	allow
#4	10.0.0.6	any	10.0.0.1	4791	UDP	300	none	none	CNP	allow
default	any	any	any	4791	UDP	any	any	any	any	deny

Janus Expressions

- Objective: Identify Minimum but Necessary Packet Fields
  - Necessary fields
    - Network entities to imply a node’s network-layer identity
    - QP operations
      - Control path: QP creation and destroy
      - Data path: reads/writes/atomics on remote memory
  - Irrelevant fields
    - Application-layer: packet sequence number, rkey, DMA length…
  - High-level policy languages
    - Easier interface with necessary fields and actions

Janus Expressions

- Objective: Identify Minimum but Necessary Packet Fields
  - Necessary fields
    - Network entities to imply a node's network-layer identity
    - QP operations
      - Control path: QP creation and destroy
      - Data path: reads/writes/atomics on remote memory
  - Irrelevant fields
    - Application-layer: packet sequence number, rkey, DMA length…
  - High-level policy languages
    - Easier interface with necessary fields and actions

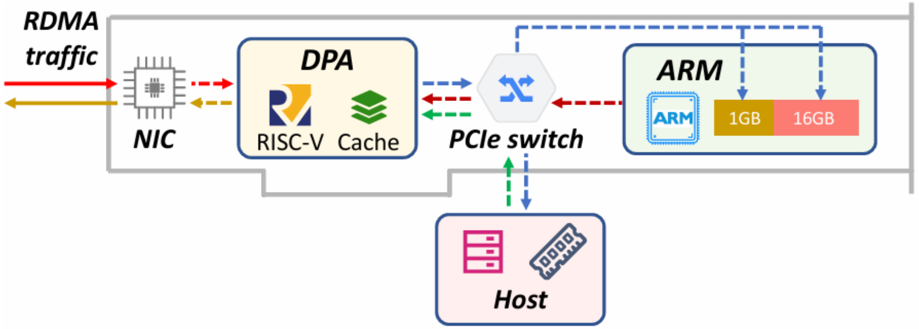
Please check out more details in our paper!

Outline

- Background and Motivation
- Janus Design
  - System overview
  - Tailored ACL expression
  - DPU-based ACL enforcement
- Evaluation

DPU Hardware

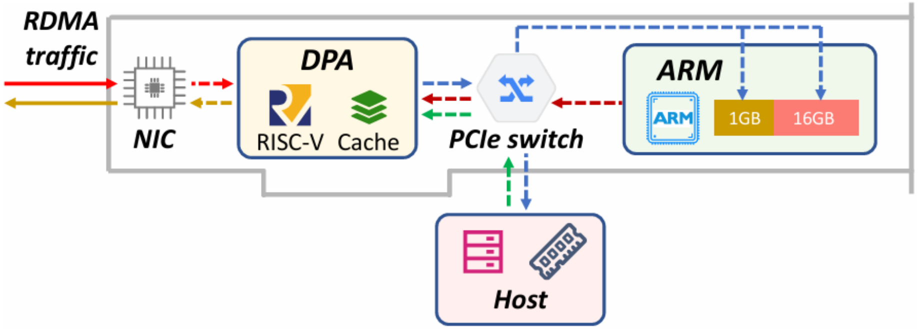
- DPUs as ACL Enforcer
  - NVIDIA BlueField-3 DPU specification
    - Programmable smartNIC for network function offloading



[1] Demystifying Datapath Accelerator Enhanced Off-path SmartNIC, ICNP 24

DPU Hardware

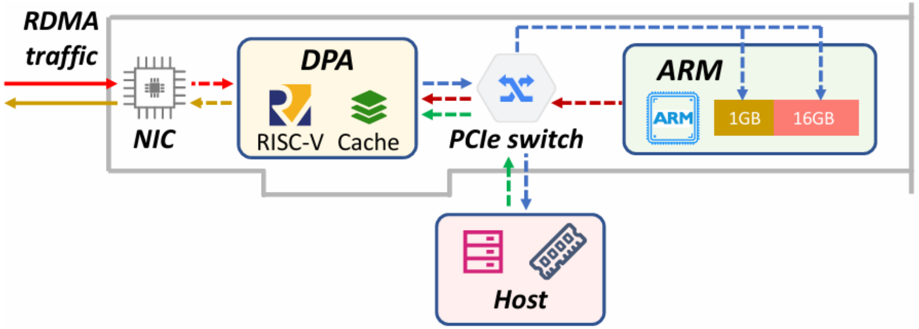
- DPUs as ACL Enforcer
  - NVIDIA BlueField-3 DPU specification
    - Programmable smartNIC for network function offloading
    - Compute resource
      - ARM core for general computation
      - Multi-thread DPA core with parallelism capability



[1] Demystifying Datapath Accelerator Enhanced Off-path SmartNIC, ICNP 24

DPU Hardware

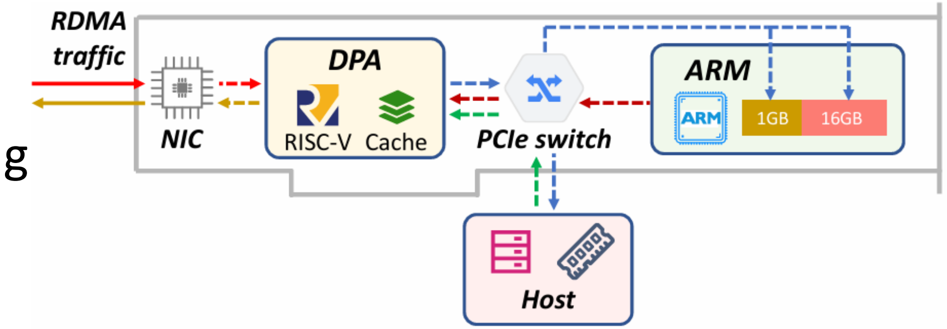
- DPUs as ACL Enforcer
  - NVIDIA BlueField-3 DPU specification
    - Programmable smartNIC for network function offloading
    - Compute resource
      - ARM core for general computation
      - Multi-thread DPA core with parallelism capability
    - Memory resource
      - Local L1/2/3 cache; sharable DRAM at DPA and ARM

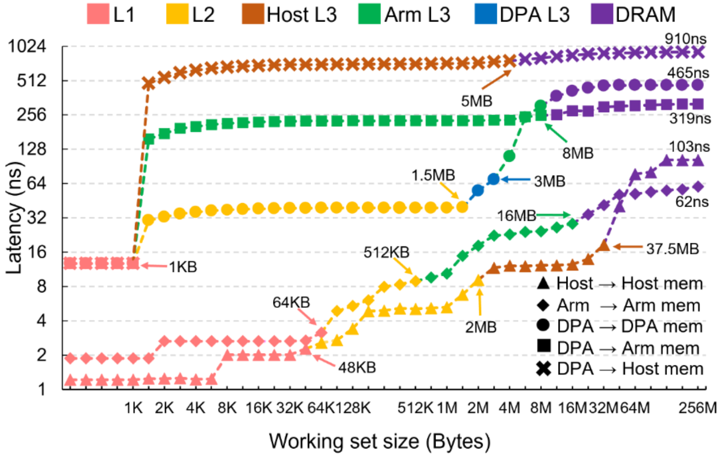


1. Demystifying Datapath Accelerator Enhanced Off-path SmartNIC, ICNP 24

DPU Hardware

- DPUs as ACL Enforcer
  - NVIDIA BlueField-3 DPU specification
    - Programmable smartNIC for network function offloading
    - Compute resource
      - ARM core for general computation
      - Multi-thread DPA core with parallelism capability
    - Memory resource
      - Local L1/2/3 cache; sharable DRAM at DPA and ARM
  - Why choose DPU?
    - Locate in the critical path with full coverage of RDMA traffic
    - High parallelism with line-rate potentials

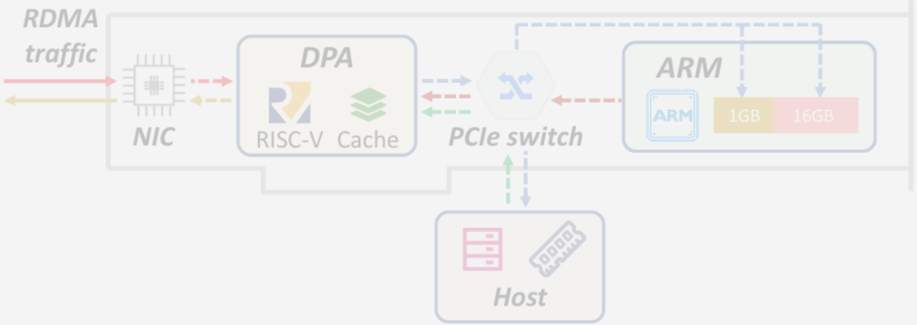


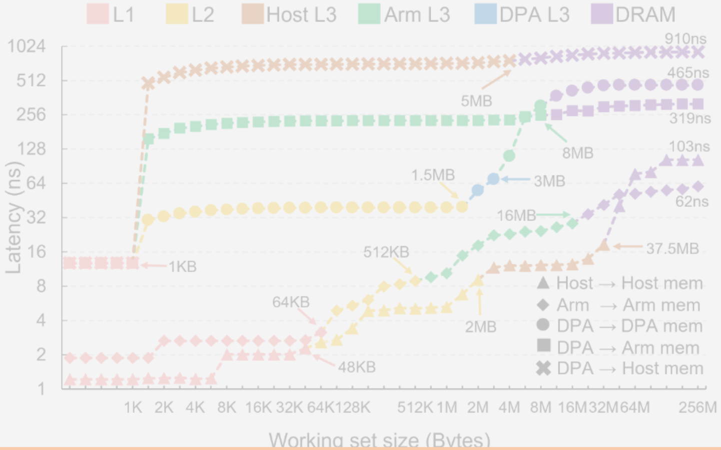


[1] Demystifying Datapath Accelerator Enhanced Off-path SmartNIC, ICNP 24

DPU Hardware

- DPUs as ACL Enforcer
  - NVIDIA BlueField-3 DPU specification
    - Programmable smartNIC for network function offloading
    - Compute resource
      - ARM core for general computation
      - Multi-thread DPA core with parallelism capability
    - Memory resource
      - Local L1/2/3 cache; sharable DRAM at DPA and ARM
  - Why choose DPU?
    - Locate in the critical path with full coverage of RDMA traffic
    - High parallelism with line-rate potentials

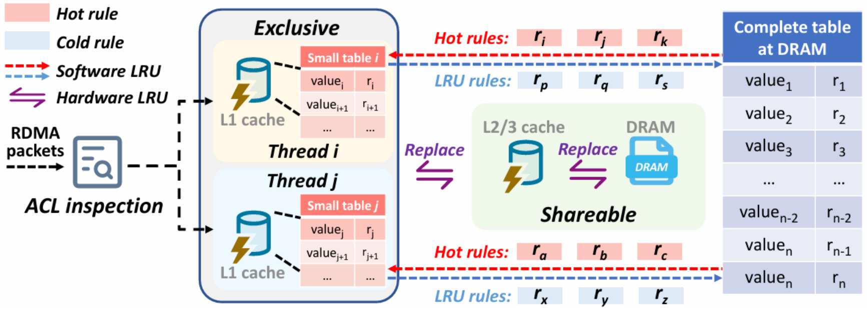




How to fully utilize DPU capability to deliver performant ACLs?

Janus Enforcement

- Efficient Enforcement with Tailored Designs
  - Cache-friendly hash table
    - Isolate hot-cold rules with hierarchical tables
    - Improve L1 cache hit rate for higher throughput



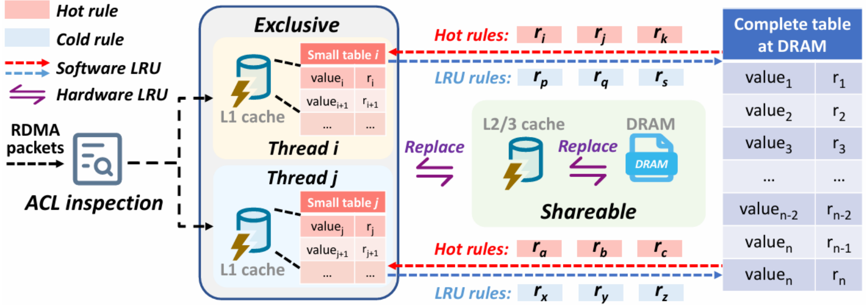
Small table *i*	
$value_i$	r_i
$value_{i+1}$	r_{i+1}
…	…

Small table *j*	
$value_j$	r_j
$value_{j+1}$	r_{j+1}
…	…

Complete table at DRAM	
$value_1$	r_1
$value_2$	r_2
$value_3$	r_3
…	…
$value_{n-2}$	r_{n-2}
$value_n$	r_{n-1}
$value_n$	r_n

Janus Enforcement

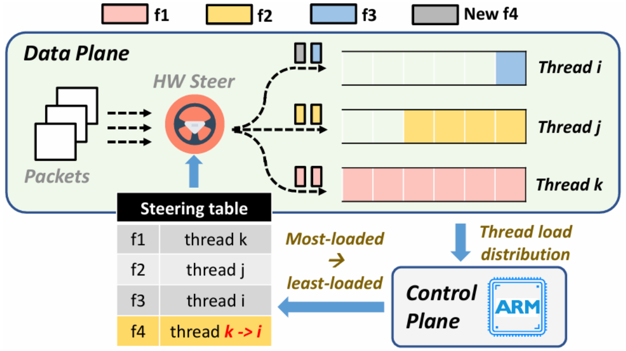
- Efficient Enforcement with Tailored Designs
  - Cache-friendly hash table
    - Isolate hot-cold rules with hierarchical tables
    - Improve L1 cache hit rate for higher throughput
  - Load-aware packet steering
    - Dynamic configuration to steering table
    - Balance thread loads with heuristic allocation



Small table i	
$value_i$	r_i
$value_{i+1}$	r_{i+1}
...	...

Small table j	
$value_j$	r_j
$value_{j+1}$	r_{j+1}
...	...

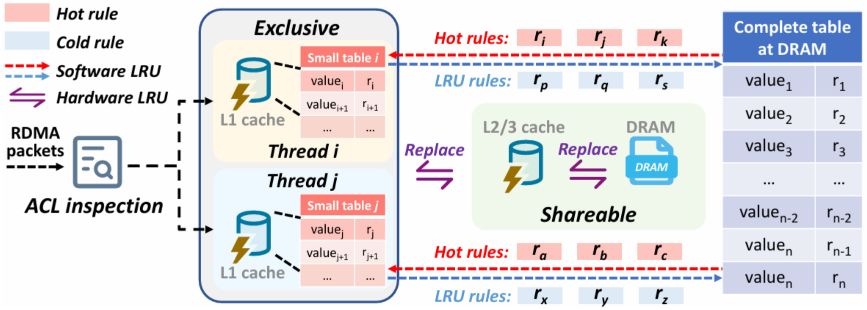
Complete table at DRAM	
$value_1$	r_1
$value_2$	r_2
$value_3$	r_3
...	...
$value_{n-2}$	r_{n-2}
$value_n$	r_{n-1}
$value_n$	r_n



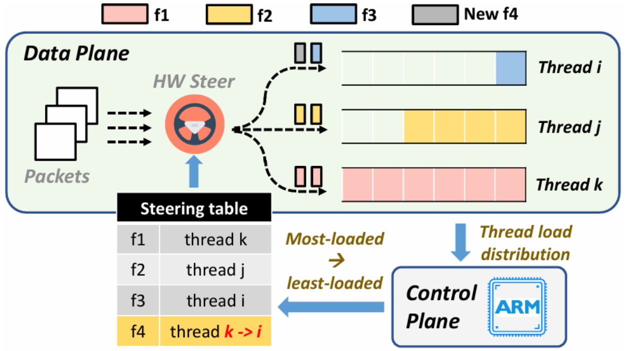
Steering table	
f1	thread k
f2	thread j
f3	thread i
f4	thread k -> i

Janus Enforcement

- Efficient Enforcement with Tailored Designs
  - – Cache-friendly hash table
    - Isolate hot-cold rules with hierarchical tables
    - Improve L1 cache hit rate for higher throughput
  - – Load-aware packet steering
    - Dynamic configuration to steering table
    - Balance thread loads with heuristic allocation



Complete table at DRAM	
$value_1$	r_1
$value_2$	r_2
$value_3$	r_3
...	...
$value_{n-2}$	r_{n-2}
$value_n$	r_{n-1}
$value_n$	r_n



Steering table	
f1	thread k
f2	thread j
f3	thread i
f4	thread *k -> i*

Please check out more details in our paper!

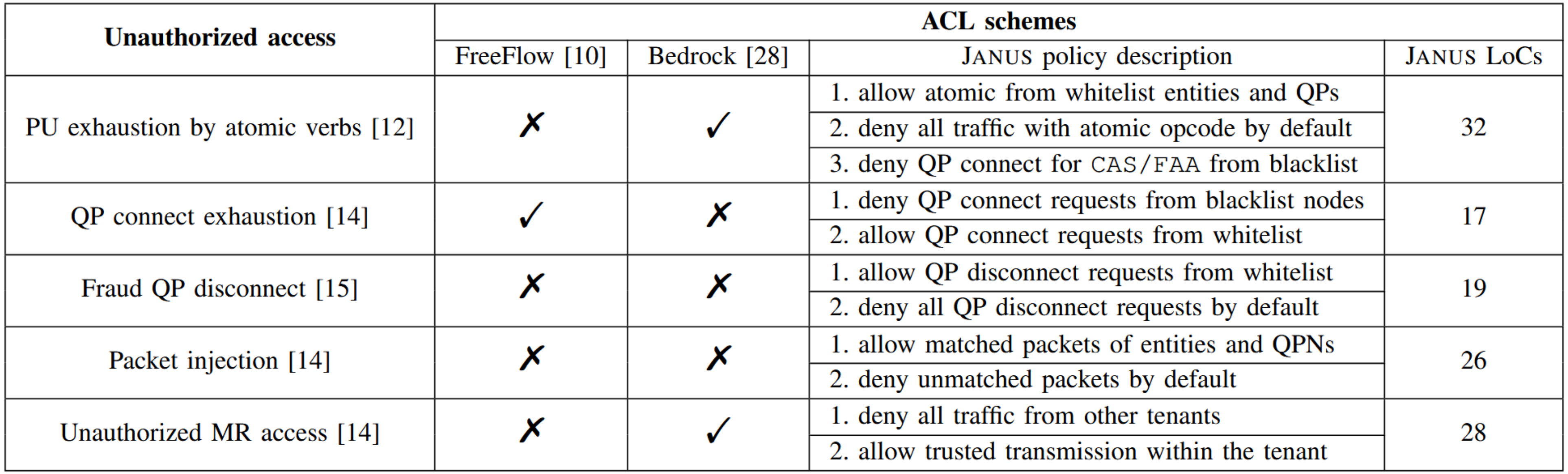
Outline

- Background and Motivation
- Janus Design
  - System overview
  - Tailored ACL expression
  - DPU-based ACL enforcement
- Evaluation

Evaluation Results

- Expression Evaluation

TABLE IV: Comparison of existing ACLs against unauthorized accesses.

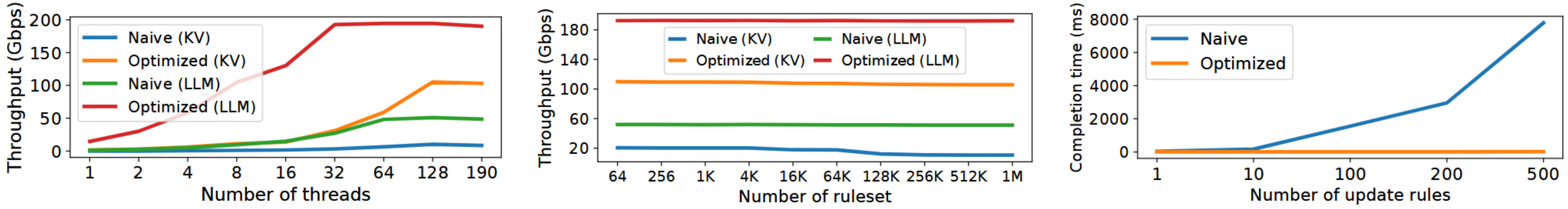


Unauthorized access	ACL schemes			
	FreeFlow [10]	Bedrock [28]	JANUS policy description	JANUS LoCs
PU exhaustion by atomic verbs [12]	✗	✓	1. allow atomic from whitelist entities and QPs 2. deny all traffic with atomic opcode by default 3. deny QP connect for `CAS/FAA` from blacklist	32
QP connect exhaustion [14]	✓	✗	1. deny QP connect requests from blacklist nodes 2. allow QP connect requests from whitelist	17
Fraud QP disconnect [15]	✗	✗	1. allow QP disconnect requests from whitelist 2. deny all QP disconnect requests by default	19
Packet injection [14]	✗	✗	1. allow matched packets of entities and QPNs 2. deny unmatched packets by default	26
Unauthorized MR access [14]	✗	✓	1. deny all traffic from other tenants 2. allow trusted transmission within the tenant	28

Operators can block unauthorized accesses with easy usage!

Evaluation Results

- Enforcement Evaluation
  - Overall performance

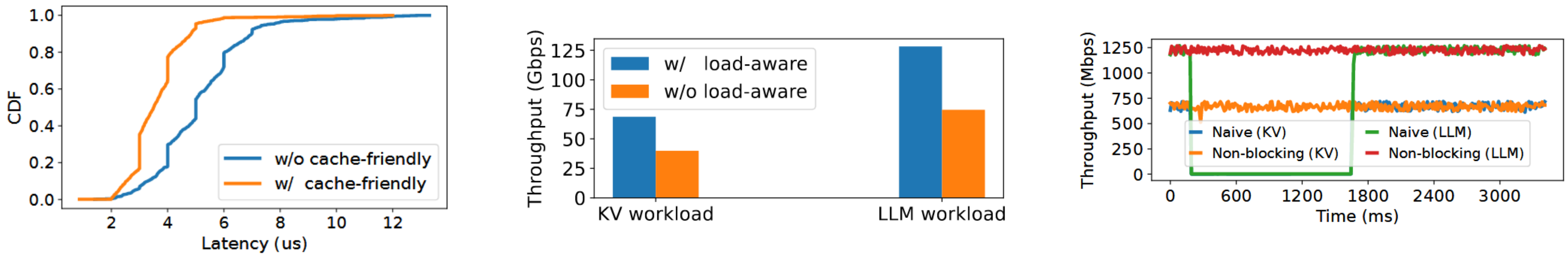


(a) Parallelism setting

(b) Ruleset size

(c) Policy update

  - Ablation studies



Janus can deliver ACLs with 200Gbps throughput and <5us latency!

Takeaways

- Existing ACL Paradigm Fails in RDMA Clouds
  - Five-tuple expressions cannot represent QP semantics
  - Lack comprehensive and efficient coverage on RDMA traffic

Takeaways

- Existing ACL Paradigm Fails in RDMA Clouds
  - Five-tuple expressions cannot represent QP semantics
  - Lack comprehensive and efficient coverage on RDMA traffic
- Janus: Expressive and Efficient ACLs
  - Tailored expressions to capture QP semantics
  - DPU-based enforcement with tailored optimizations

Takeaways

- Existing ACL Paradigm Fails in RDMA Clouds
  - Five-tuple expressions cannot represent QP semantics
  - Lack comprehensive and efficient coverage on RDMA traffic
- Janus: Expressive and Efficient ACLs
  - Tailored expressions to capture QP semantics
  - DPU-based enforcement with tailored optimizations
- Implementation and Evaluation
  - Achieve better expressivity and line-rate inspection in real testbed

Thanks for Your Attention!

Q & A