White-Boxing RDMA with Packet-Granular Software Control

NSDI'25

Frank Chenxingyu Zhao, Jaehong Min, Ming Liu, Arvind Krishnamurthy

University of Washington; University of Wisconsin-Madison

Presenter: Huang Kai

2025.9.20

Author

- **研究方向**
  - 可编程网络、高性能网络、数据中心分布式系统、机器学习系统
- **近期论文**
  - **可编程网络**: Gimbal: Enabling Multi-tenant Storage Disaggregation on SmartNIC JBOFs (SIGCOMM'21)
  - **高性能网络**: CC-NIC: a Cache-Coherent Interface to the NIC(ASPLOS'24)　Host Congestion Control (SIGCOMM'23)
  - **数据中心分布式系统**: Xenic: SmartNIC-Accelerated Distributed Transactions (SOSP'21)
  - **机器学习系统**: Efficient Direct-Connect Topologies for Collective Communications (NSDI'25)　Punica: Multi-tenant LoRA serving (MLSys'24)

Arvind Krishnamurthy

Short-Dooley Professor,
University of Washington

Content

- Background
- Motivation
- Design
- Evaluation
- Thinking

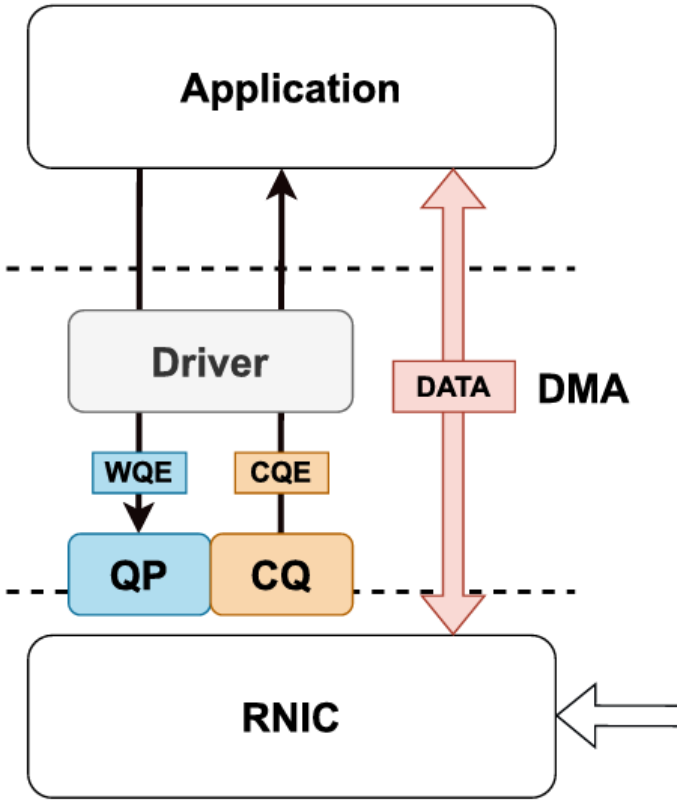
Background

- 数据中心网络对高吞吐低延迟有要求
  - RDMA NIC

特性	普通 NIC (传统网卡)	RNIC (RDMA 网卡)
核心技术	标准以太网/IP协议处理	支持 RDMA (远程直接内存访问)
CPU 开销	较高，CPU 深度参与数据传输与协议处理	极低，RDMA 绕过 CPU 进行内存直接访问
延迟	相对较高	非常低
吞吐量	受 CPU 限制	更高，能更充分利用网络带宽
主要优势	成本较低，通用性强	极低的延迟，极高的吞吐量，CPU 卸载
适用场景	通用网络连接	高性能计算、数据中心、低延迟应用等

Background

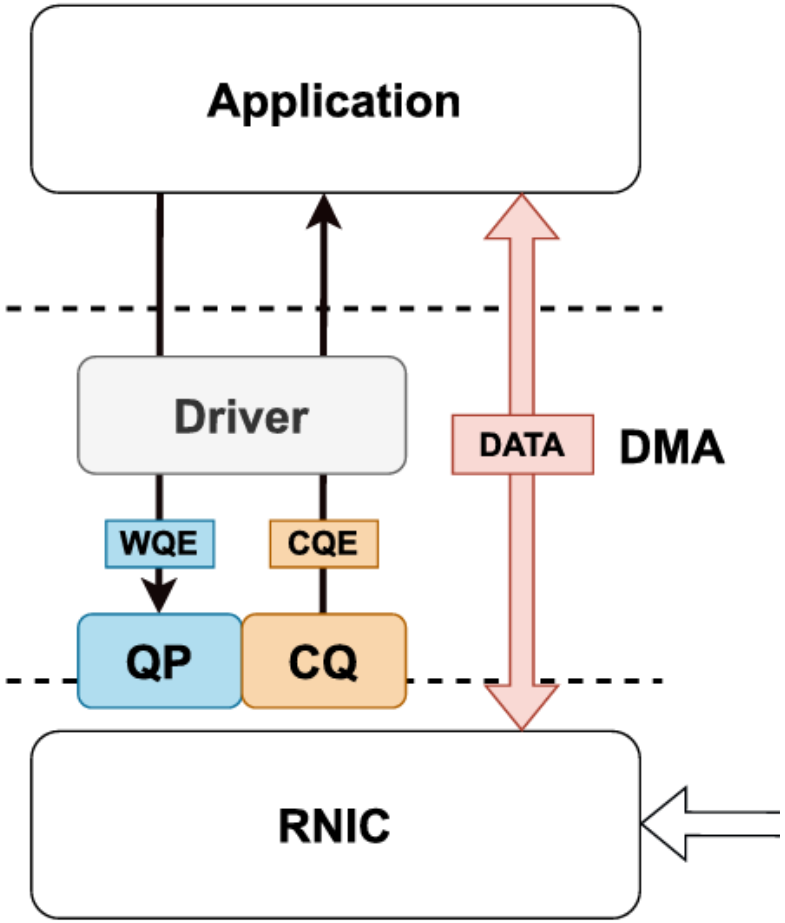
- 数据中心网络对低CPU开销、高吞吐低延迟有要求
  - RDMA 传输机制



- Work Queue Entry, WQE QP操作单元
- Queue Pair, QP 包括发送队列和接收队列
- Complete Queue, CQ 通知App任务完成
- Complete Queue Entry, CQE WQE完成后生成相应的CQE，并传递给CQ

Motivation

- RNIC(RDMA NIC) 算法定制 与 细粒度控制难度大



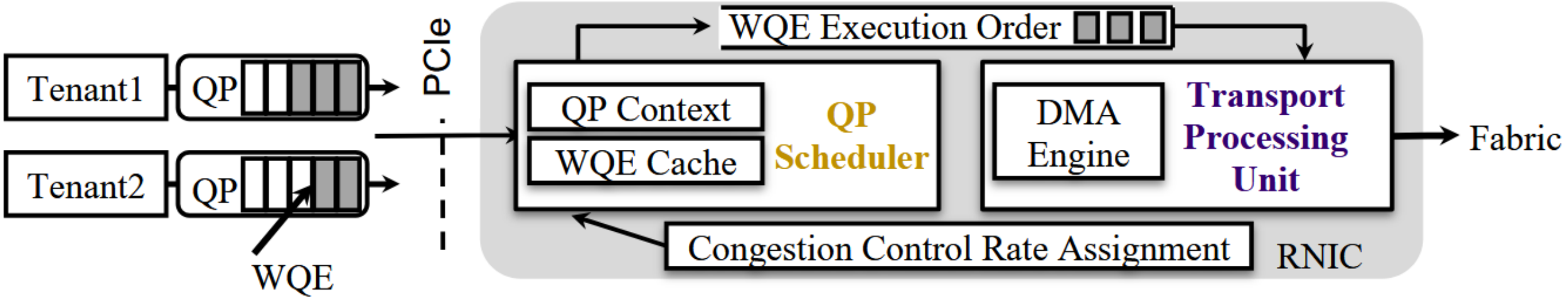
FPGA
High Dev. Cost

ASIC
Wait for Years

SW Emulate
Perf. Loss

Blackboxing
Coarse & Delay

Motivation



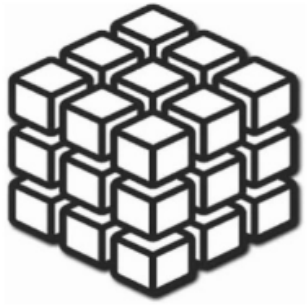
FPGA	**ASIC**	**SW Emulate**	**Whiteboxing**
High Dev. Cost	Wait for Years	Perf. Loss	SW Ctrl HW Data

Challenges

- SCR(Software Control RNIC) 存在的挑战

Flexibility

Express Diverse Customizations



Granularity

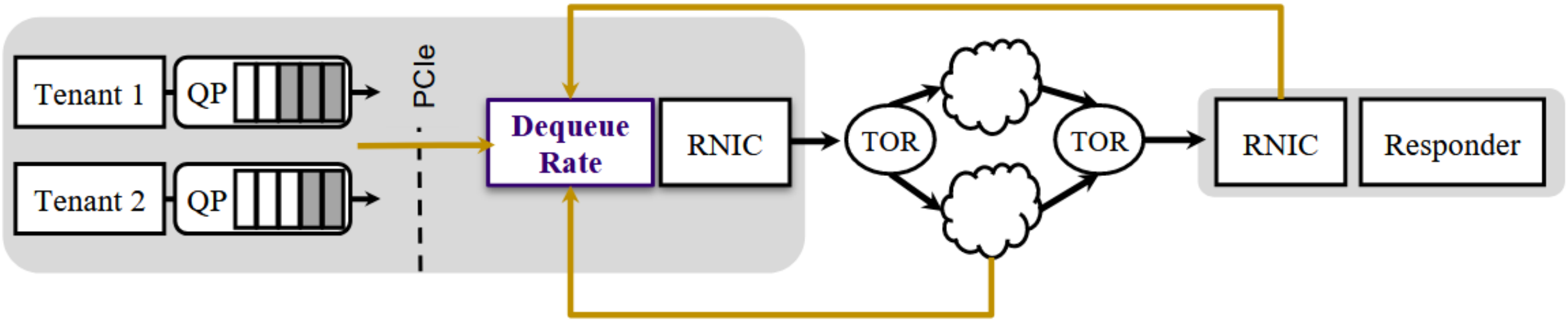
Packet-Granular Events from High Line Rates

Scalability

Handle Many Concurrent Flows

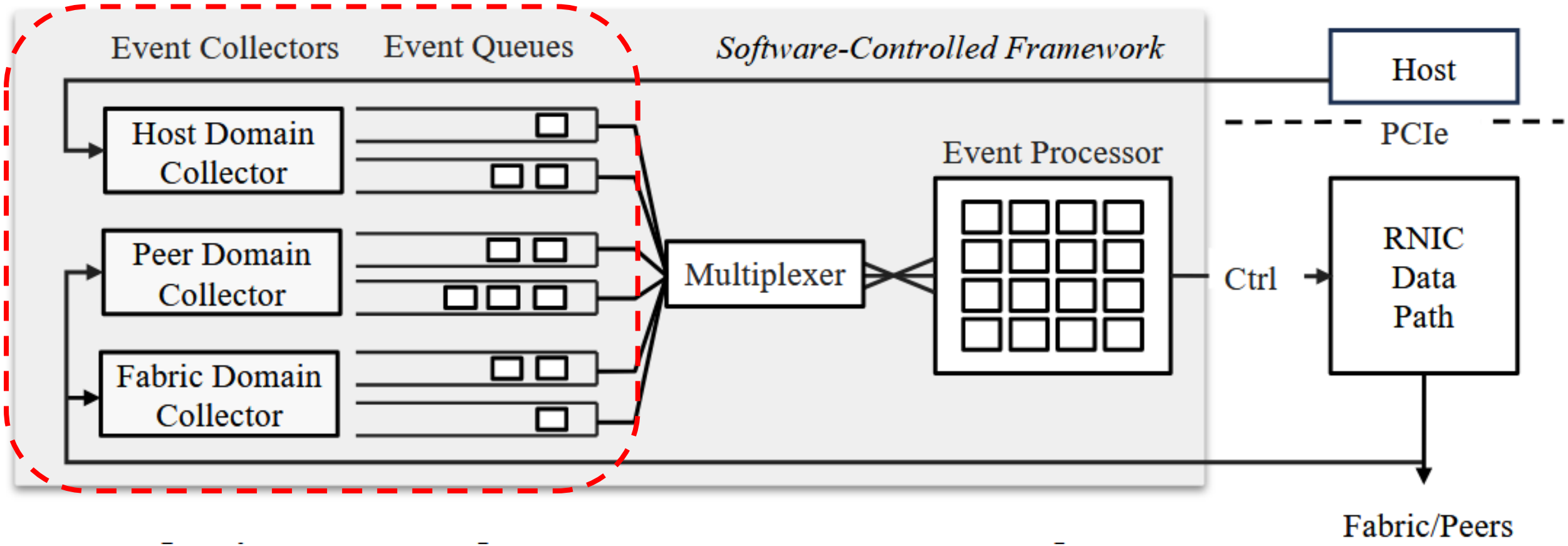
What to Control & How to Control

What to Control -- Flexibility



- 针对不同控制信号实现复杂场景下的细粒度控制
  - 主机控制域：QP(流)出队速率
  - 链路控制域：RNIC链路吞吐率、网络拥塞信号
  - 端到端控制域：接收端反馈信号

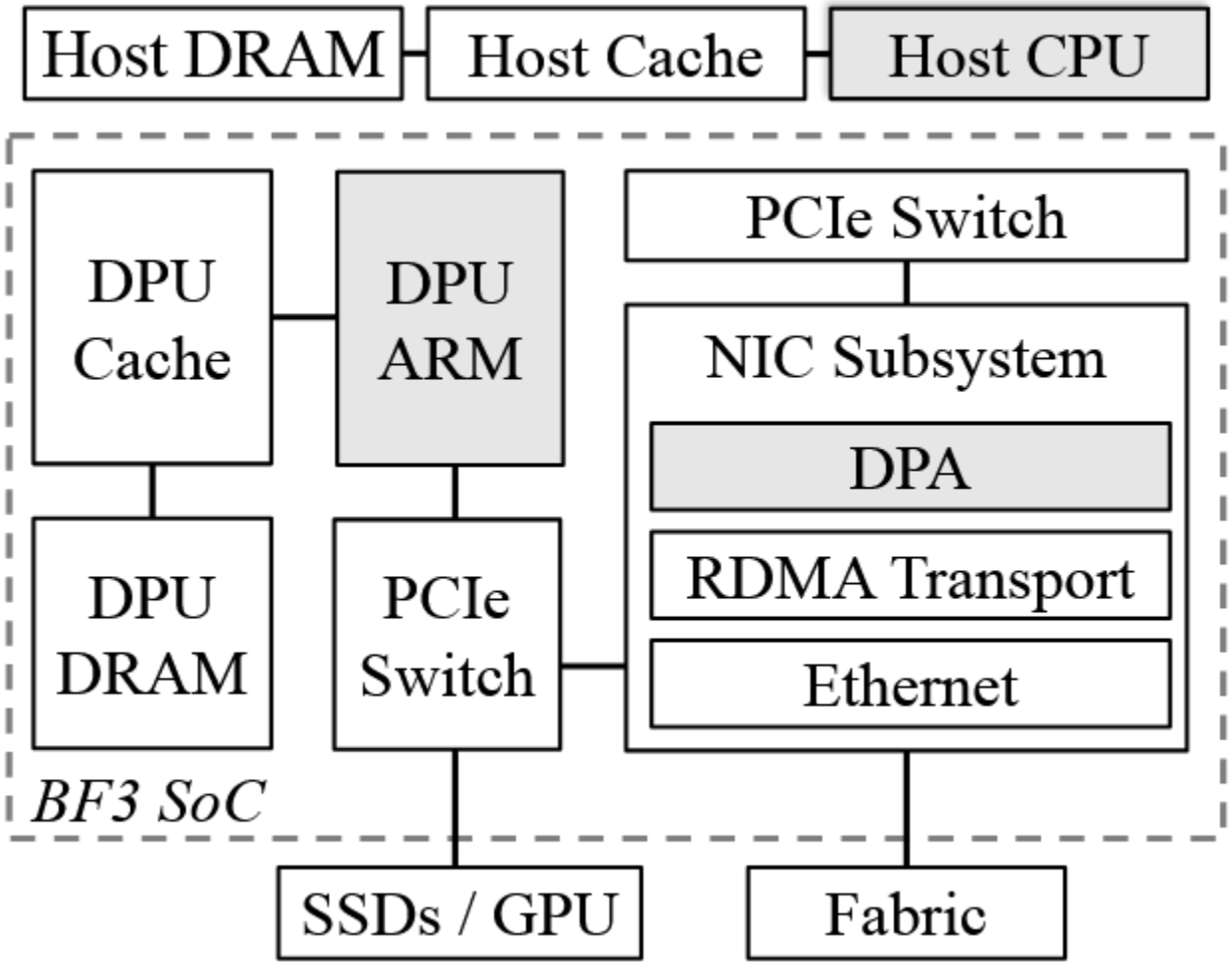
Design -- Overview



QP事件驱动的速率计算模型

How to Control

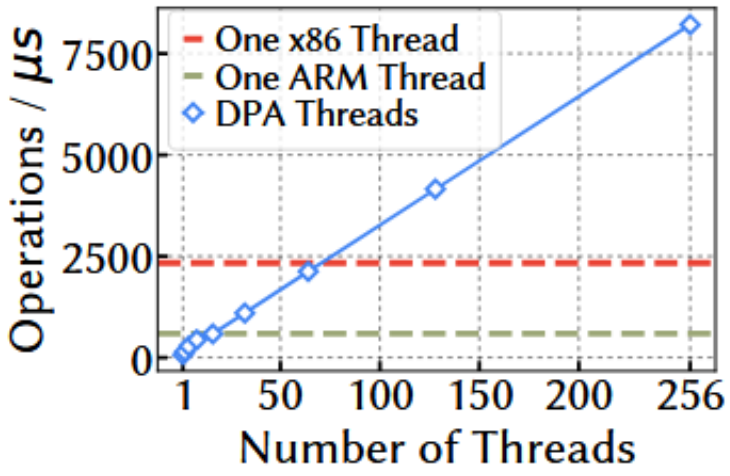
- Nvidia BlueField-3架构处理器
  - Host CPU
  - DPU-embedded ARM CPU
  - DPA(Datapath Accelerators)
    - 卸载控制平面 (CPU) 的数据转发任务，实现硬件级加速

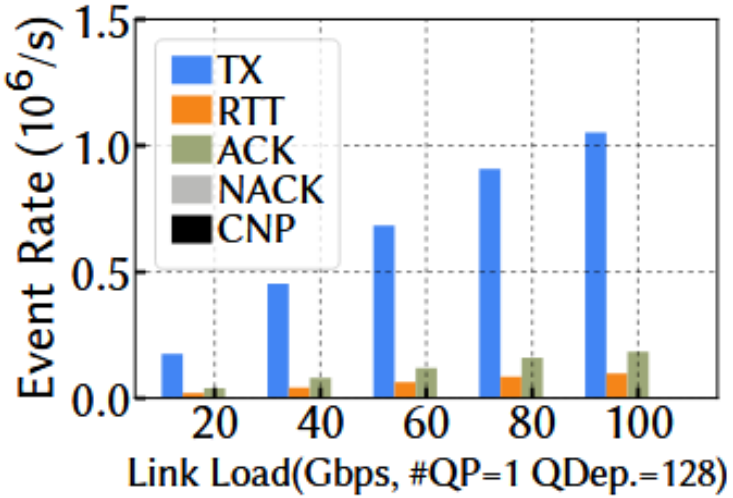


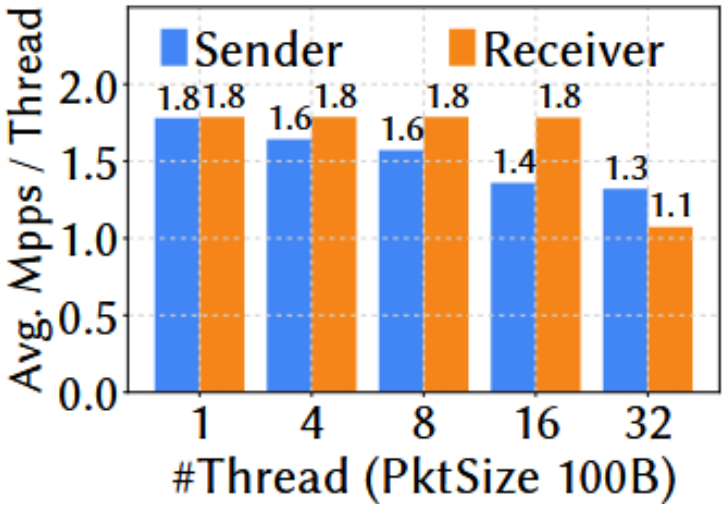
BlueField-3架构

How to Contro -- DPA

- DPA性能测试
  - 多线程性能
  - 端到端信号的采集
  - 吞吐效率

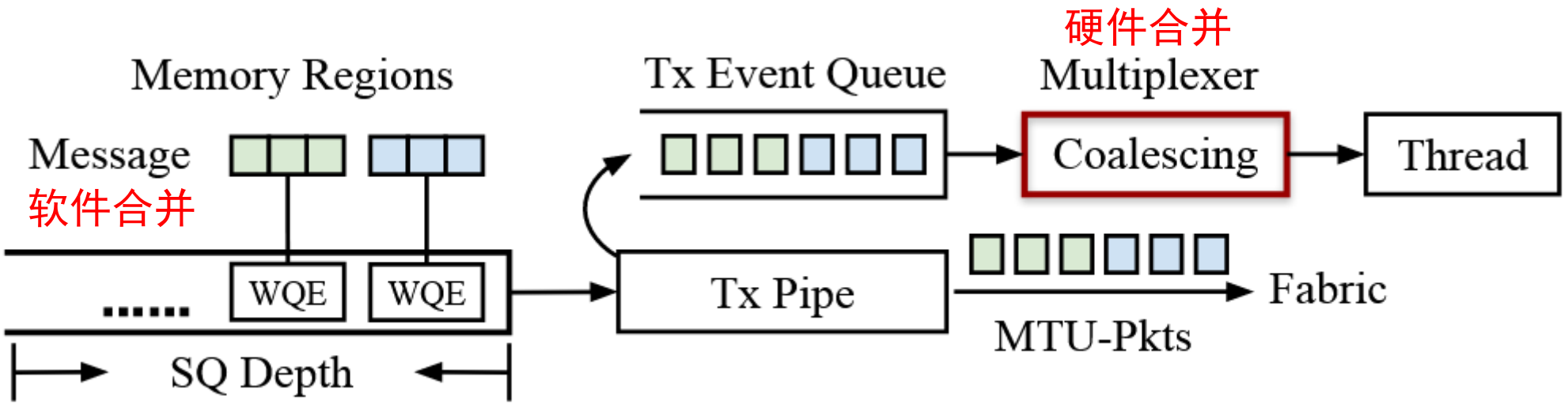






How to Control -- Granularity

- 硬件合并（ASIC -> RNIC TX/RX pipelines）
  - 只能合并链路信号
- 软件合并（DPA -> QP SQ/RQ）
  - DPA进程间通信能力有限——CPU/DPU ARM 代理进程间通信

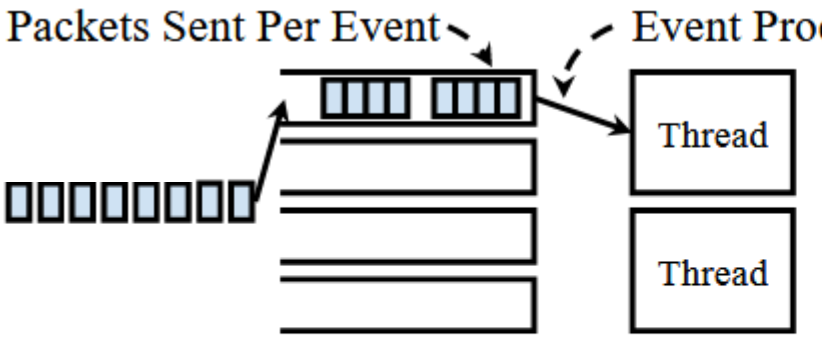


How to Control Many Flows -- Scalability

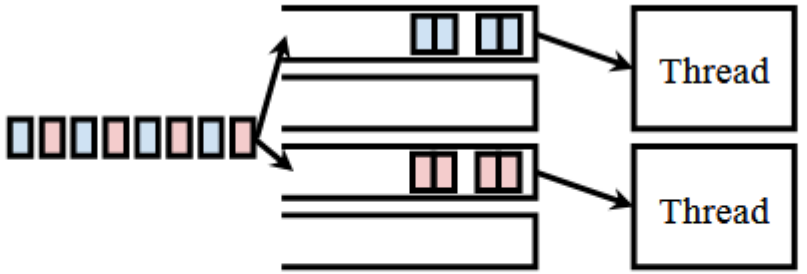
- 处理线程的数量不需要随着QP(流)的数量进行扩大

$$LineRate = \sum_{i=1}^{N} BytesSentPerEvent_i \times EventRate_i$$

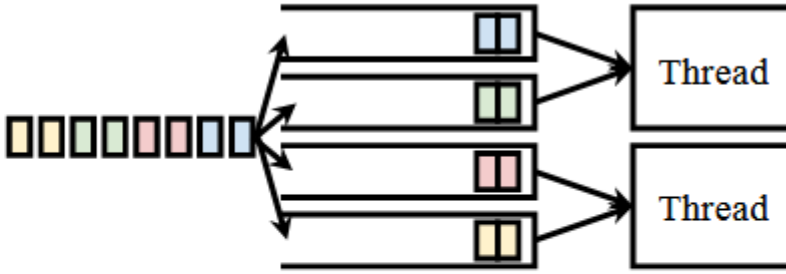
$$= Bspe \times \sum_{i=1}^{N} EventRate_i$$



1 QP : *2* Threads
Granularity = 4



2 QPs : *2* Threads
Granularity = 2



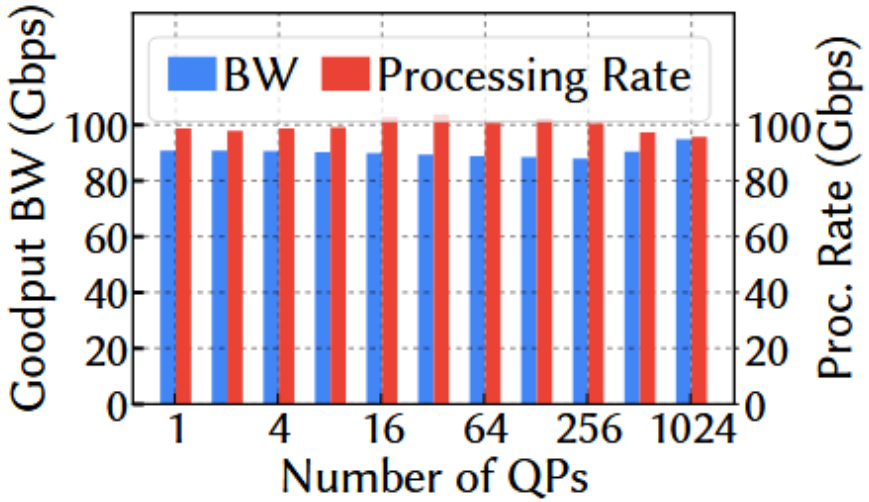
4 QPs : *2* Threads
Granularity = 2

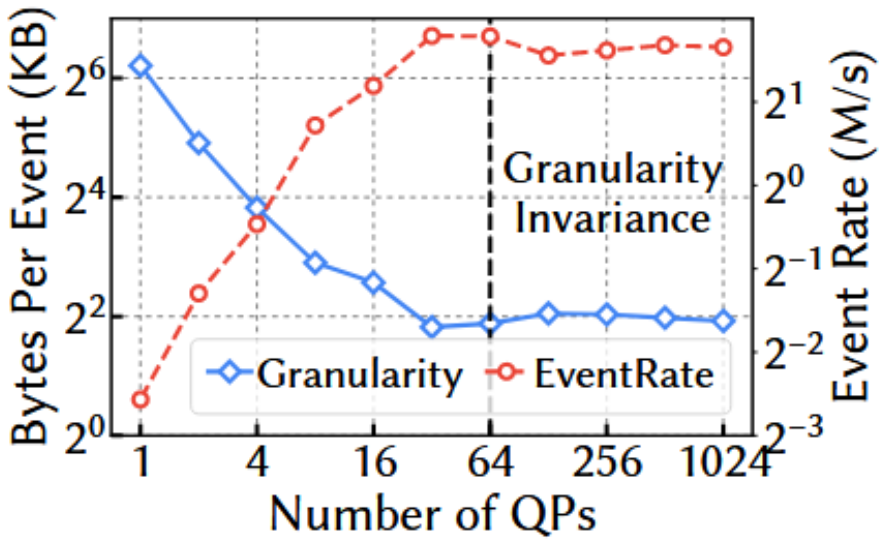
Evaluation -- Setting

- 实验设置
  - 每个节点的QP (流)的数量最多为1024个
  - 数据包大小为64KB，MTU为1500B
  - 吞吐速率为100Gbps
  - DPA线程数为64
- 评价指标
  - Event Rate：事件（信号）处理速率（million/s）
  - Coalescing Granularity：数据包处理粒度
  - Processing Rate： Event Rate × Coalescing Granularity

Evaluation -- Efficiency and Granularity

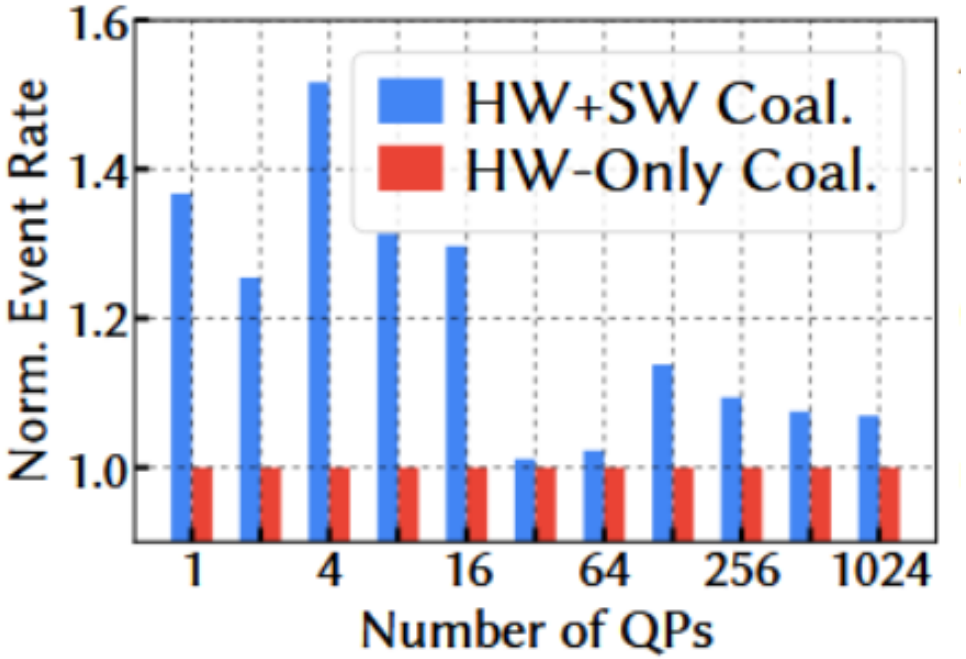
- 处理速率与粒度
  - 处理速率与数据包带宽(BW)一致
  - QP数大于64后，粒度保持不变
    - 粒度稳定时为4，符合包级粒度控制





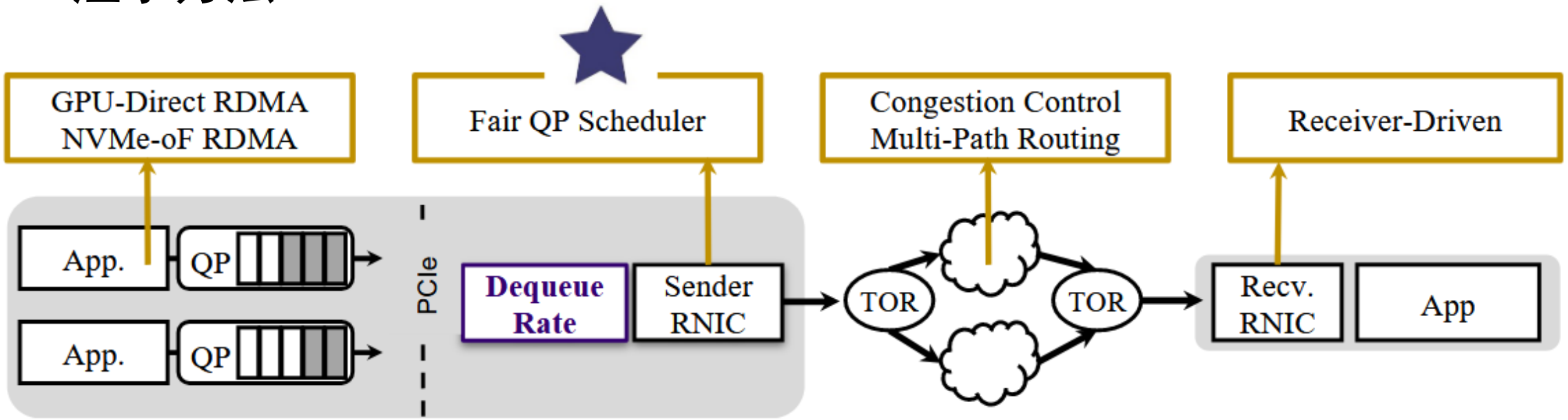
Evaluation -- Software Coalesce

- 软硬件合并，事件（信号）处理速率
  - 软件(SW)合并后的事件处理速率相对于硬件(HW)合并有所提升
    - QP数较小时提升倍率更为明显



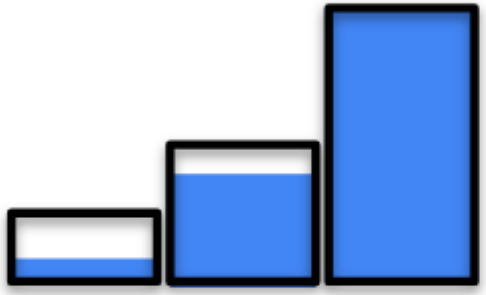
Evaluation -- New Control Laws

- 以公平队列调度为例
  - 长流优先（默认方法）
  - 静态分配
  - 注水方法



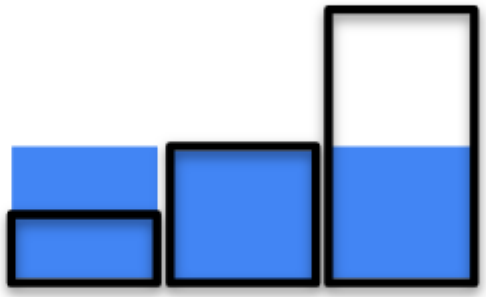
Evaluation -- Fair QP Scheduler

Available Bandwidth



Small QP | Mid. QP | Large QP

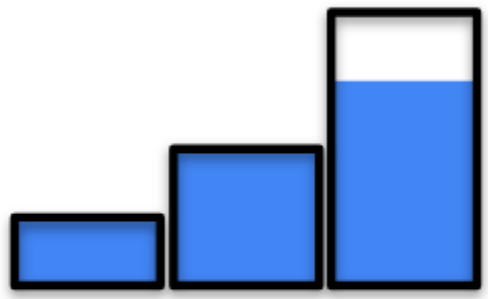
Default RNIC
Unfair for Small QP



Small QP | Mid. QP | Large QP

Static Allocation
Small QP wastes BW

Remaining BW
Zero

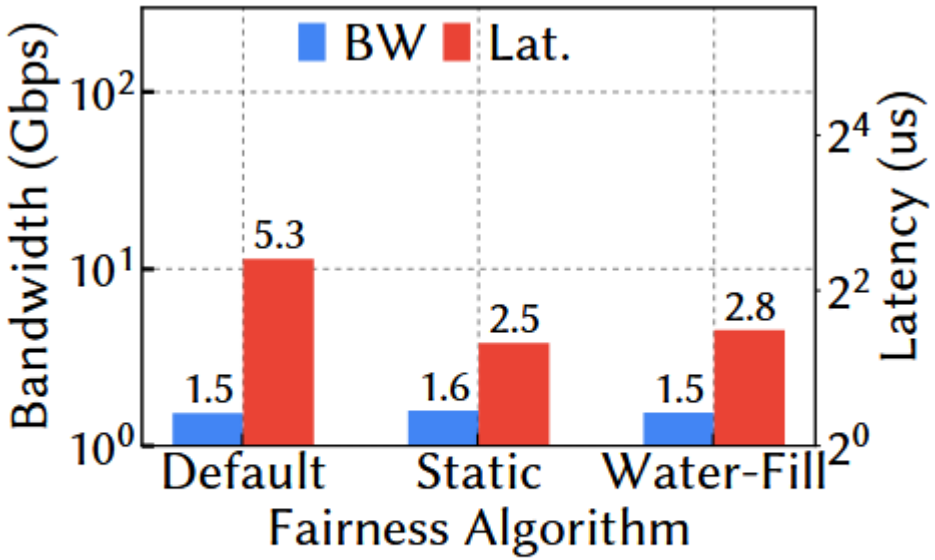


Small QP | Mid. QP | Large QP

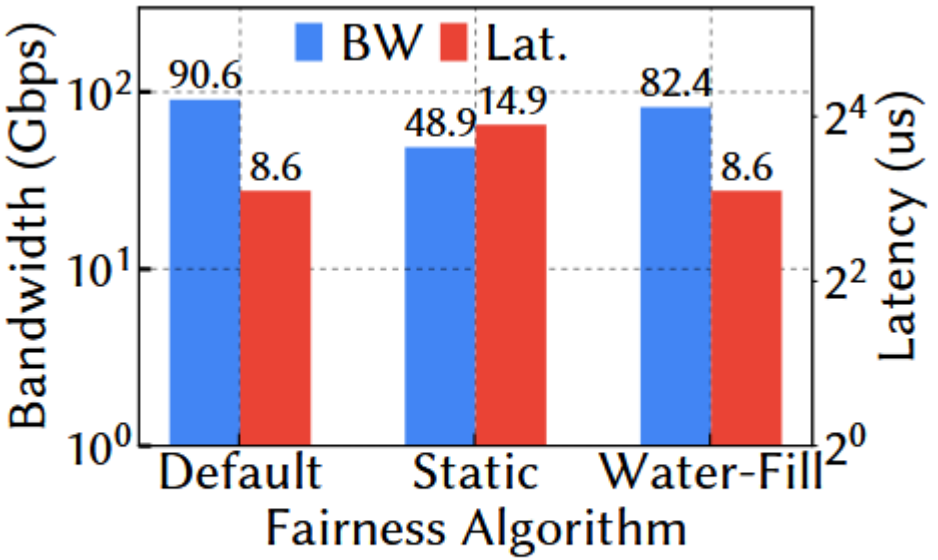
Water Filling
Fair and High Util.

Evaluation -- Fair QP Scheduler

- 公平队列调度——注水方法
  - 延迟敏感QP——延迟优于默认方法
  - 带宽敏感QP——带宽吞吐优于静态分配方法



(a) Latency-Sensitive QP



(b) Bandwidth-Hungry QP

Thinking

- 能否提高
  - 收集链路信号为局部信号，可以在RNIC内存中留存一块存储全局链路信号，并通过链路传输或者数据包携带全局信息定期更新，实现类似背压树的全局控制图
  - CPU在运行过程中只用于辅助进程间通信，可以利用CPU处理低频率单线程的全局信号，从而避免大规模网络下，全局信息读写对处理粒度产生影响

Thinking

- 与此类似
  - 绝大多数流控机制的数据面与控制面强关联，需要接收对端信号后才会实现触发式流控。可以采用论文中提到的隔离控制面的方式，实现更灵活的流控方法
- 问题泛化
  - 处理粒度取决于物理硬件的吞吐速率和线程处理速率，硬件依赖性强，泛化难度高

Q & A

Presenter: Huang Kai