PipeDevice: A Hardware-Software Co-Design Approach to Intra-Host Container Communication

Qiang Su*‡, Chuanwen Wang†, Zhixiong Niu‡, Ran Shu‡, Peng Cheng‡,
Yongqiang Xiong‡, Dongsu Han, Chun Jason Xue*, Hong Xu†
*City University of Hong Kong; ‡Microsoft Research; KAIST; †CUHK

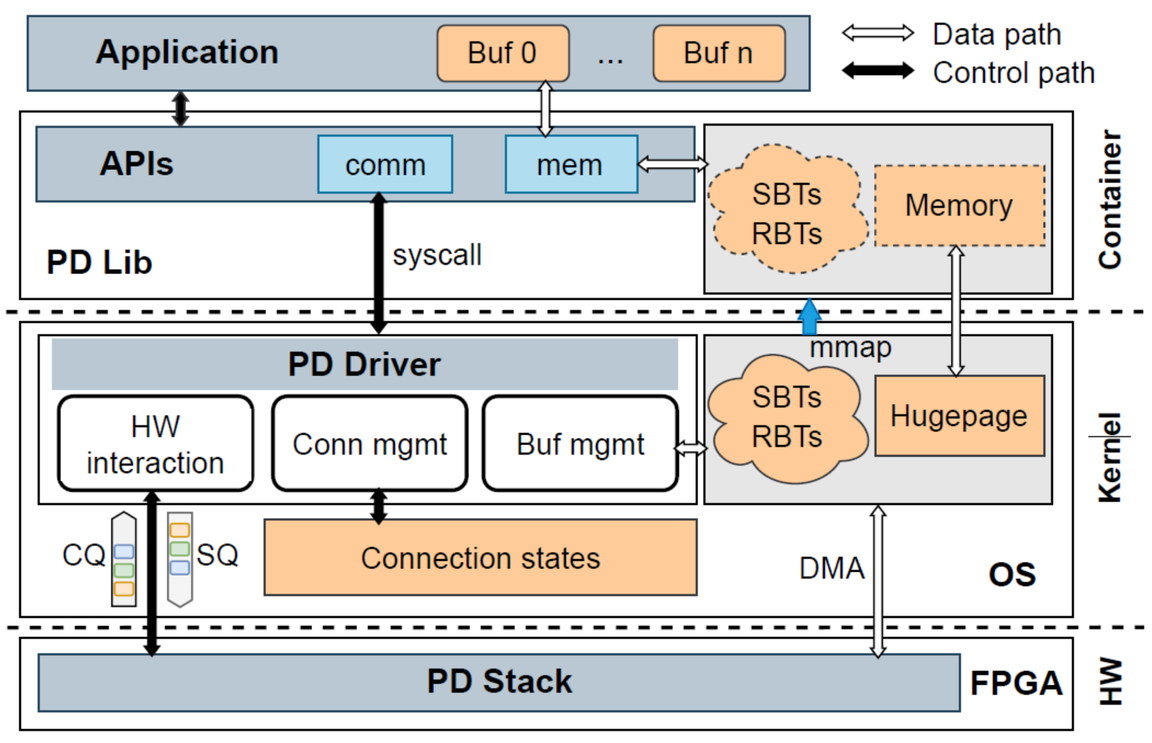
Motivation

- **Intra-host container communication is ubiquitous**
  - Various applications, e.g., the sidecar proxy in service mesh, the mapper and reducer in data-intensive frameworks
  - The principle of proximity in container placement
  - Emerging more and more resourceful servers
- **Inefficiencies of existing arts**
  - Software approach (shared-memory networking) has either high memory overhead or high CPU overhead
  - Hardware approach (RDMA) has scalability issues and insufficient semantics

PipeDevice Design

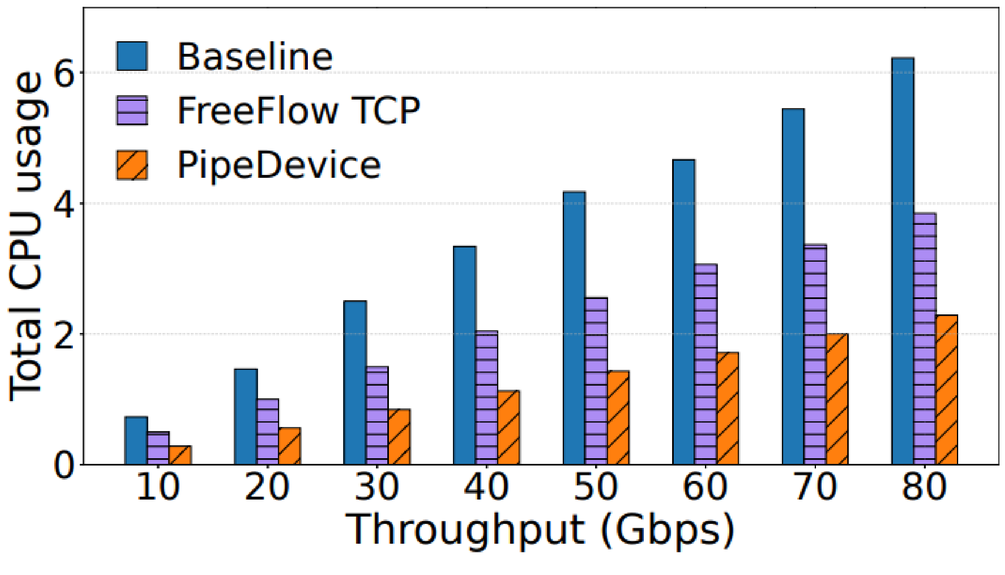
- **Hardware offloading**
  - Achieve isolation and eliminate CPU overhead
  - Support various hardware with DMA engine, e.g., FPGA, SmartNIC, and Intel IOAT
- **Scalability**
  - Keep the connection states in host DRAM and manage them entirely in software – avoid using the limited hardware resources
- **Deployability**
  - PipeDevice features a set of BSD socket-like APIs for memory and communication to allow applications to be easily ported
  - PipeDevice currently exploits FPGA that has already been deployed for accelerating various workloads

Overview

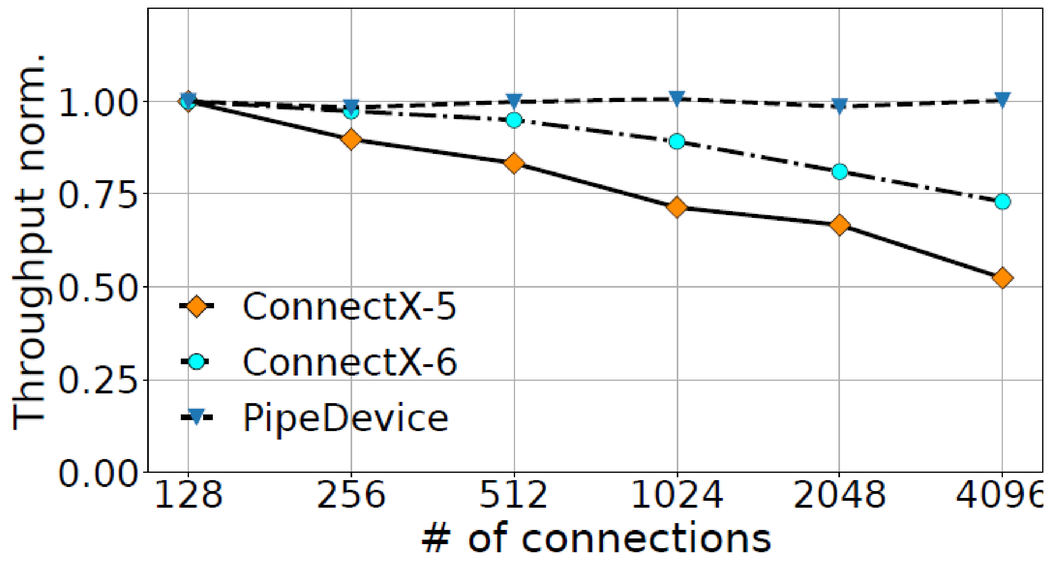


- **PD Lib**
  - Handle all communication requests
  - Expose to applications the send and receive buffers that are mapped from the hugepages
- **PD Driver**
  - Manage all the connection state information
  - Allocate per-socket send and receive buffers, and track the states SBT and RBT
  - Communicate with hardware via a lightweight per-core command queue
- **PD Stack**
  - Data transmission through hardware, *e.g.*, FPGA

Evaluation



PipeDevice significantly reduces the CPU usage while achieving the same throughput compared to Baseline (kernel TCP) and FreeFlow TCP



PipeDevice's throughput remains stable at peak while RDMA's decreases sharply due to cache contention on RNIC