HIGH PERFORMANCE NETWORK MULTIPLEXING WITH IX++

Albert Chen and Greg Hill

Background

- IX is an experimental operating system designed for high networking performance being developed at Stanford
- Originally, IX required exclusive access to a physical network interface card (NIC) to run a single application
- Running two IX applications required having two dedicated NICs

Prior work

- Prior work
  - IX
  - DPDK

Goals

Basic

- Implement SR-IOV support under IX
- Run and capture benchmarks

Stretch

- Implement software network multiplexing under IX (alternatives to SR-IOV)
- Additional benchmarks with different applications
- Explore interference patterns between applications
- Explore hardware TX scheduling (explored Linux traffic control)

Our Contributions

- Instantiated SR-IOV virtual functions and modified IX initialization code to work with network virtual functions
- Implemented TCP upload benchmark application for IX and Linux
- Benchmarked SR-IOV performance for IX and Linux

SR-IOV Integration

- VF creation and configuration
  - Set kernel parameter “pci=assign-busses”
  - Instantiate NIC virtual functions (VF) through sysfs
  - Configure VF MAC address using "ip link set" command
  - Use default Linux in-box driver for physical function (PF) operations
    - Mailbox communication between VF and PF
    - NIC management
- IX initialization
  - Modify NIC initialization to
    - Skip PF functionalities (e.g. EEPROM validation, promiscuous mode)
    - Query initialization parameters from PF (e.g. MAC Address, number of queues)
    - Merged in latest DPDK changes for VF (e.g. Rx/Tx queue setup)

TCP upload benchmark application

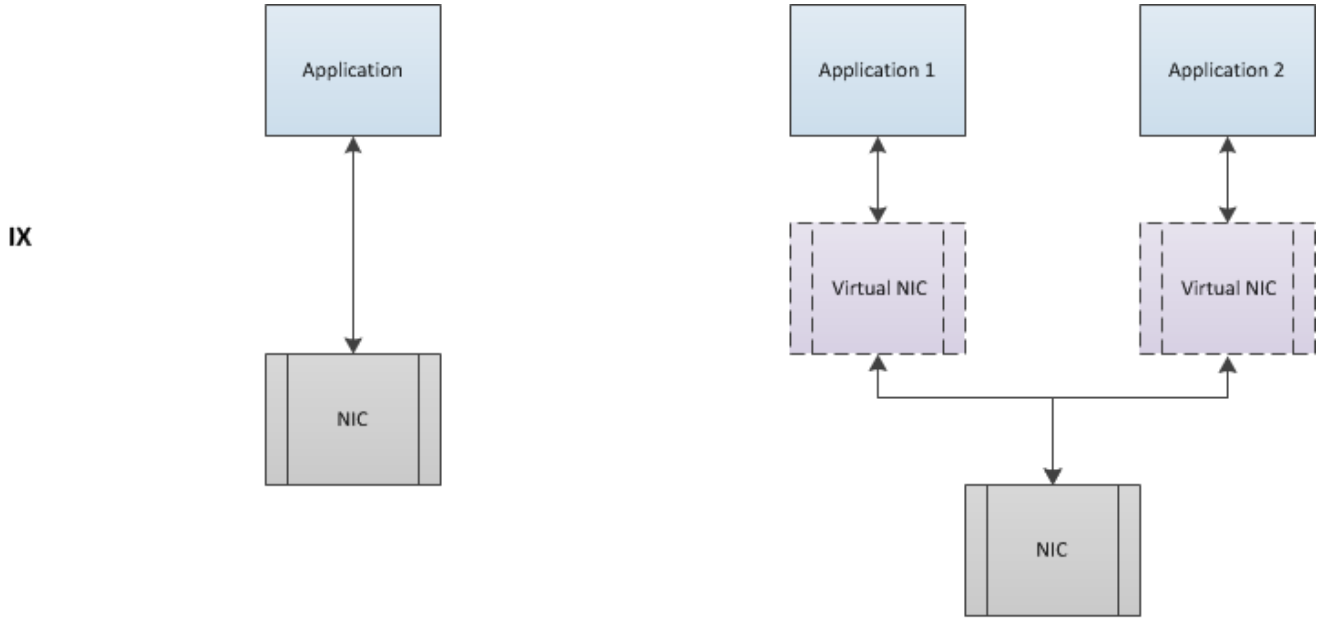
- Server
  - Linux and IX application
  - Sends a stream of bytes to each client connection as fast as TCP allows
- Client
  - Linux application
  - Reports Rx throughput
  - Multiple concurrent download streams (multi-threaded)

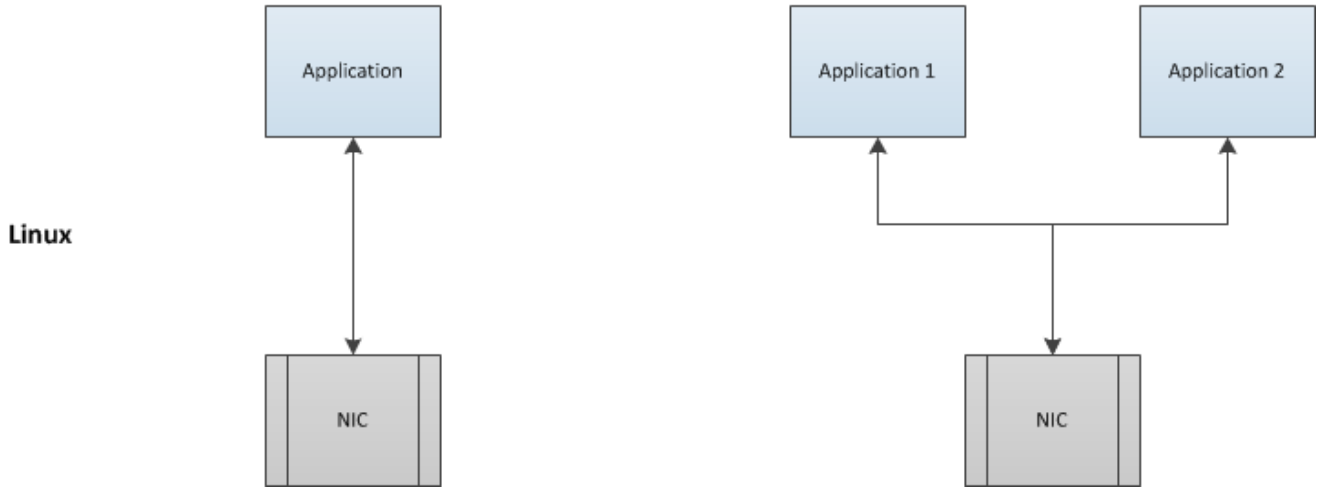
Evaluation

- Memcached with varying number of TCP streams
- TCP streams with varying Memcached throughput
- Linux traffic control

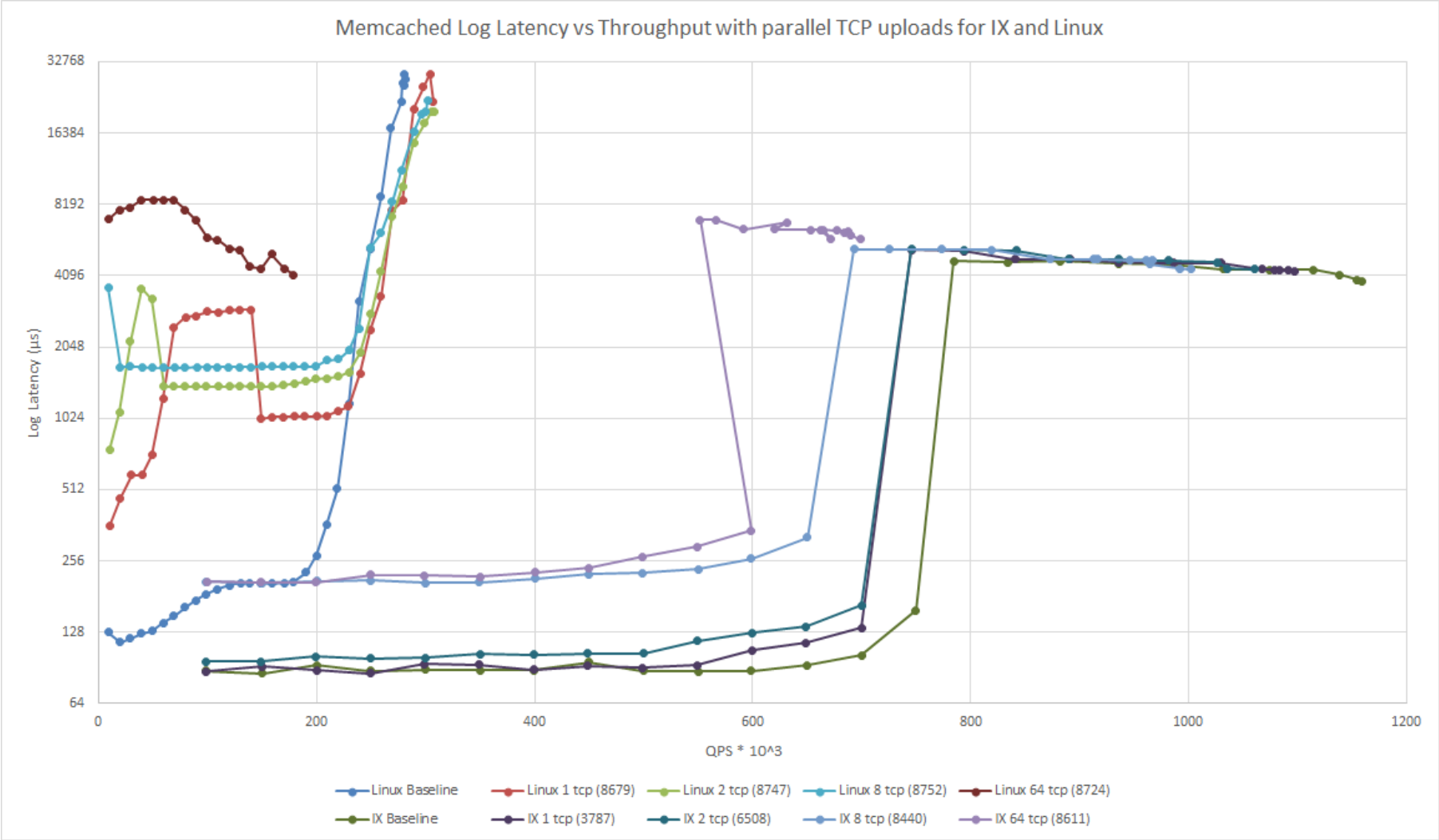
Evaluation setup

We compared running multiple applications on IX to running those applications on Linux. Our applications are TCP upload and Memcached.

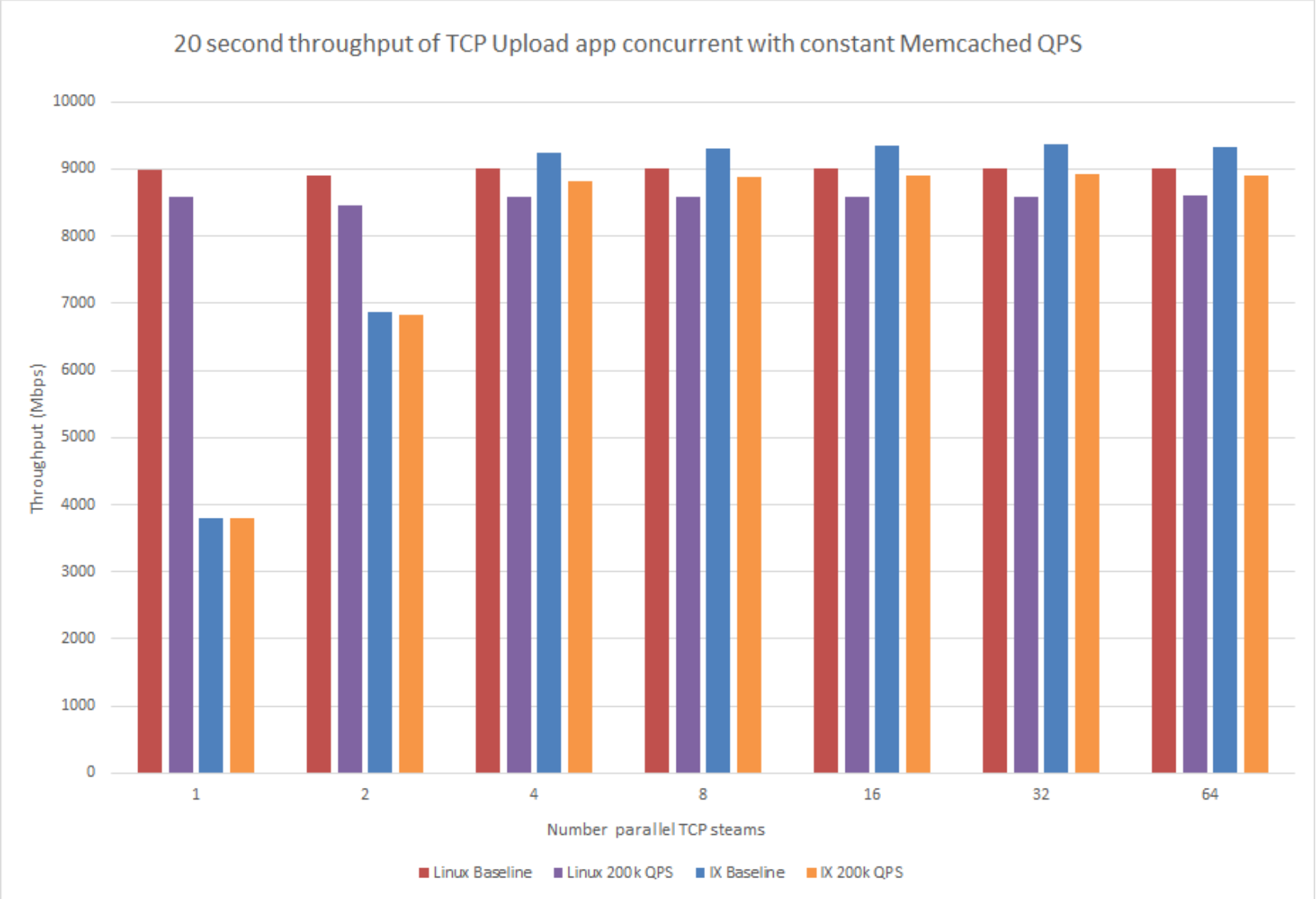




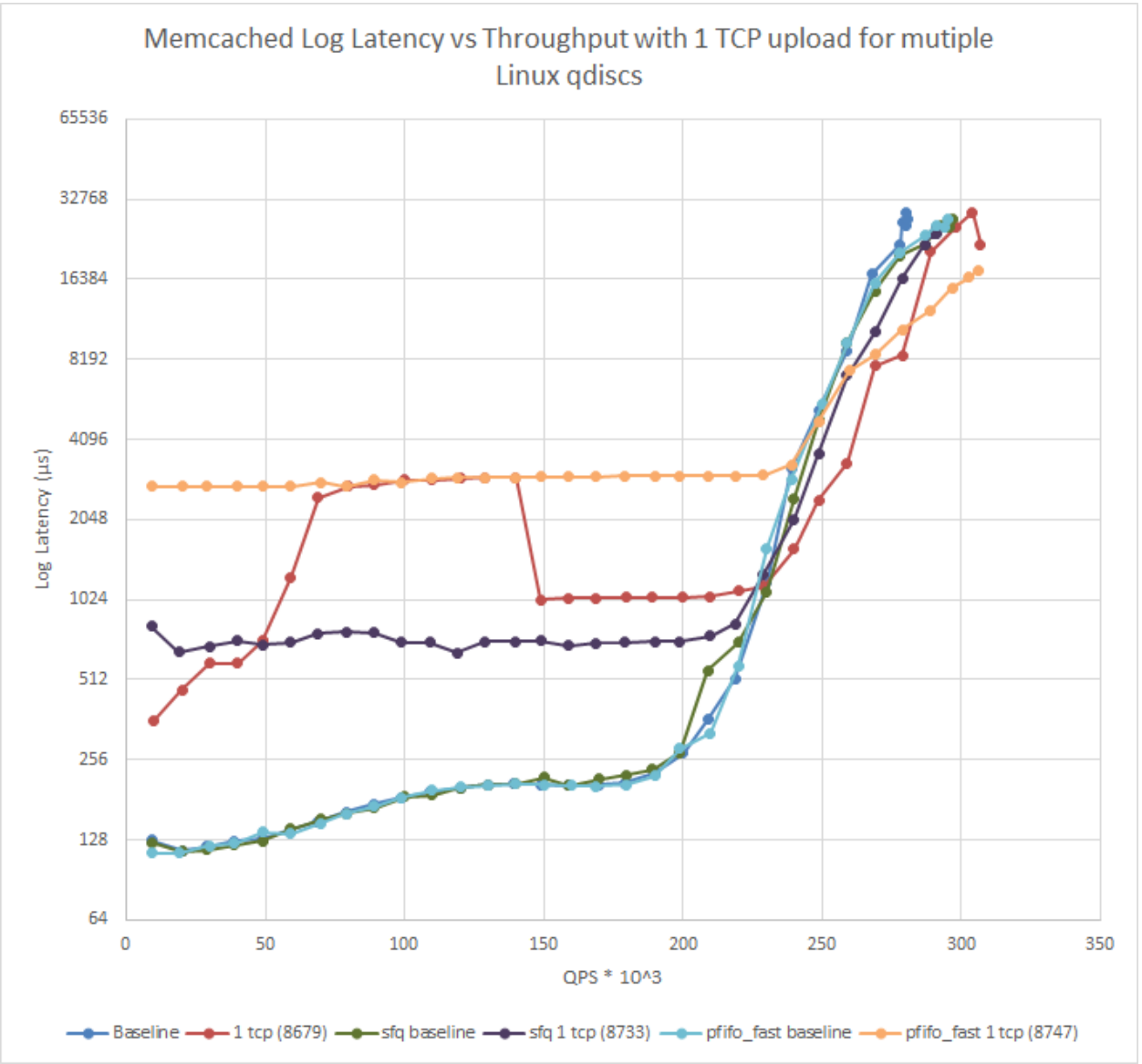
Memcached with varying number of TCP streams

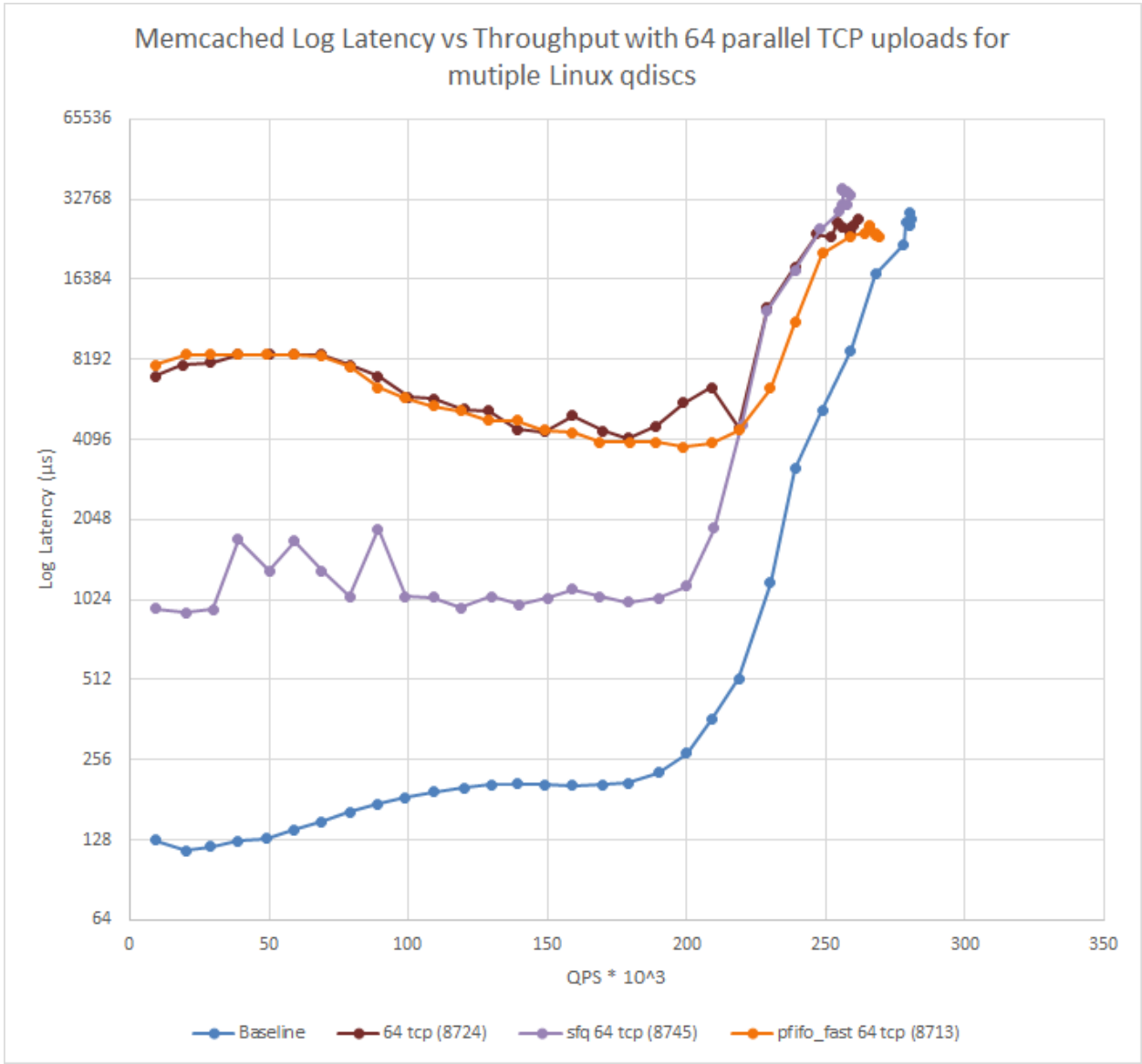


TCP streams with varying Memcached throughput



Linux qdiscs





Take aways

- Low latency applications are highly sensitive to network interference from other applications in standard Linux, regardless of Linux Traffic Control Queueing Disciplines
- IX++ maintains the performance advantages of IX while allowing multiple applications to use the NIC with minimal interference
  - It achieves this by multiplexing the NIC at the lowest possible level

Limitations

- Only able to use one CPU core per IX application
  - Tried many approaches to get around this, none successfully
    - Attempted to control hardware from VF
    - Modified Linux Intel Ethernet driver (IXGBE)
  - Probably still possible

DEMO