Sereno
A Second Generation Virtualized Network Interface Controller

Mike Galles & Shrijeet Mukherjee

Cisco Sereno: Talk Outline

- Why is Cisco building NIC ASICs?

  convergence of Ethernet, storage, and management networks

  virtualization of PCIe devices and network interfaces

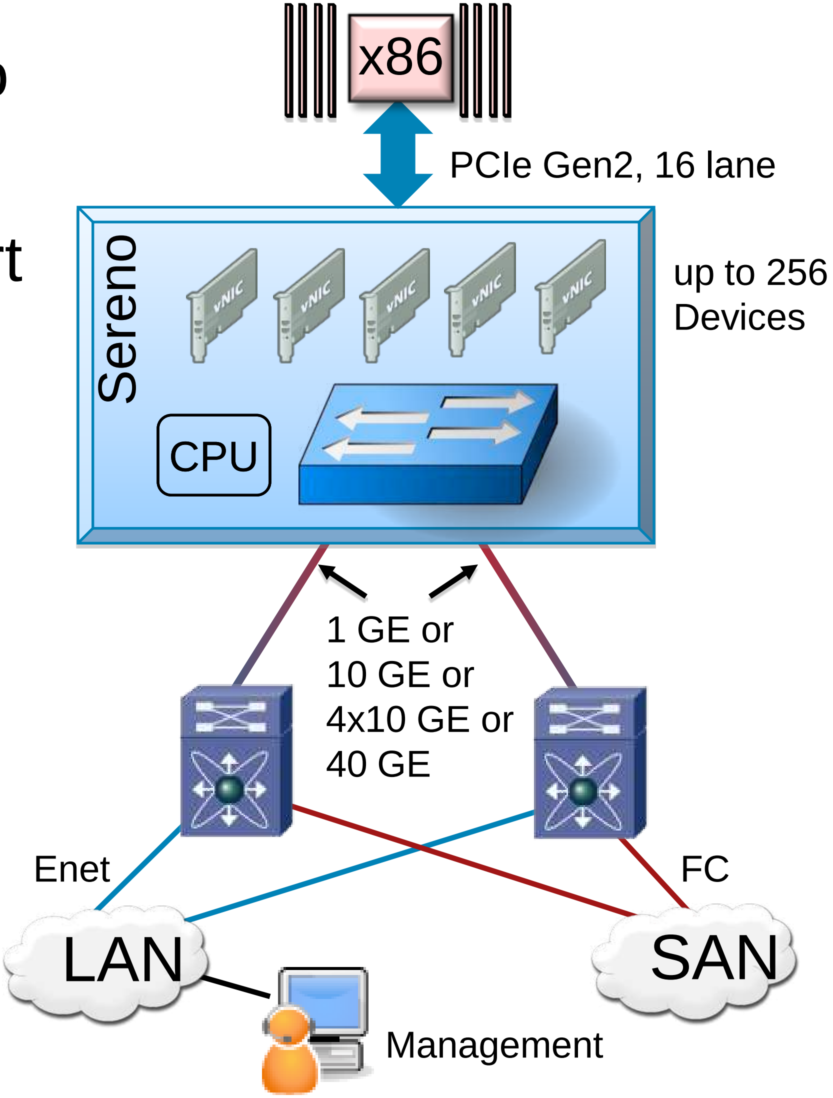
  management of virtual interfaces from the network

  network services applied at the scalable edge

- Sereno Physical Data (65 nm ASIC)
- Technical dive on select hardware features
- Drivers and Firmware
- Performance

Convergence

- Multiple Ethernet, Storage, and Management devices share two active-active physical ports
- Two physical ports each support 1GE, 10GE, 4x10GE, or 40GE operation
- 80 Gb/s of network bandwidth exceeds the 64 Gb/s of PCIe bandwidth, but enables 40 Gbs bursts to each physical port



Virtualization

- PCIe Virtualization

  256 vNICs are configurable device types (Enet, FC, …)

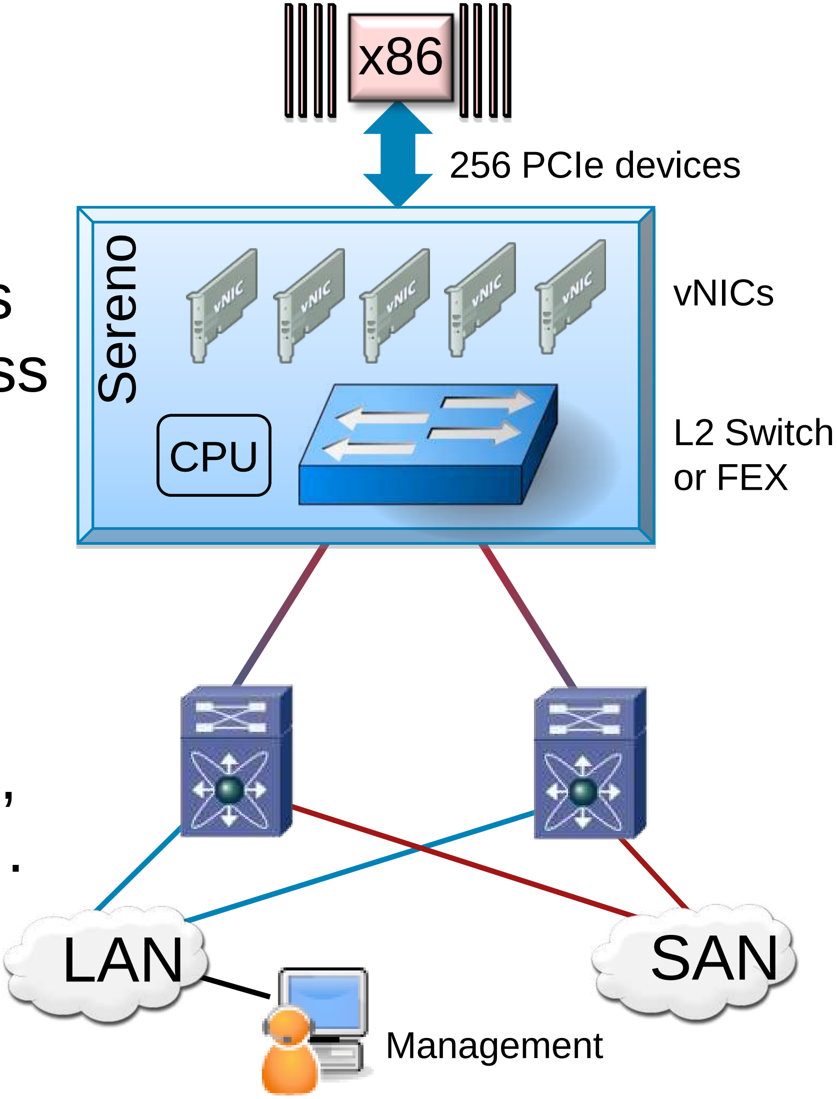
  vNICs have private PCIe BDFs with protected memory access

  vNICs may include SR-IOV functions

- Network Interface Virtualization

  each vNIC has private MAC(s), VLAN(s), multicast groups, …

  vNIC network attributes configured by local CPU



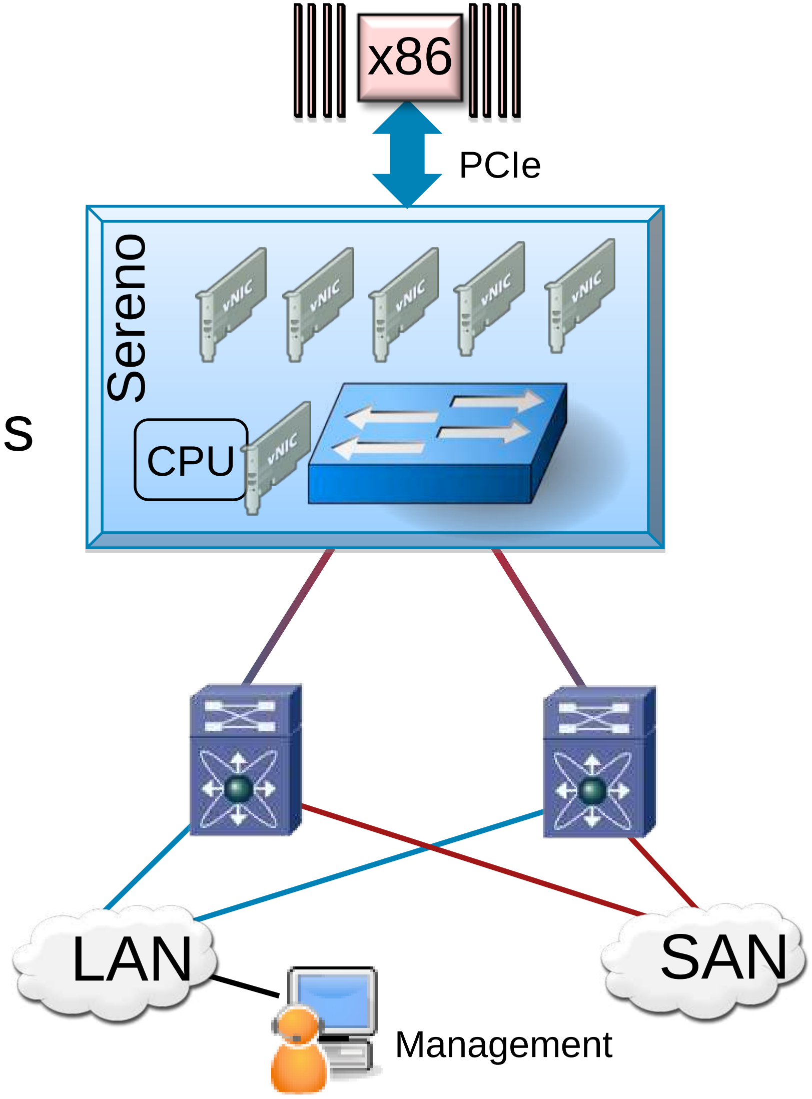
Management

- Embedded CPU (MIPS R24K) runs Linux, private vNIC gives secure network access
- Embedded CPU configures vNICs and network policies

  creates custom IO subsystems for each server

  manages network interfaces, VLANs, forwarding, filtering

  controls network failover, bandwidth allocation



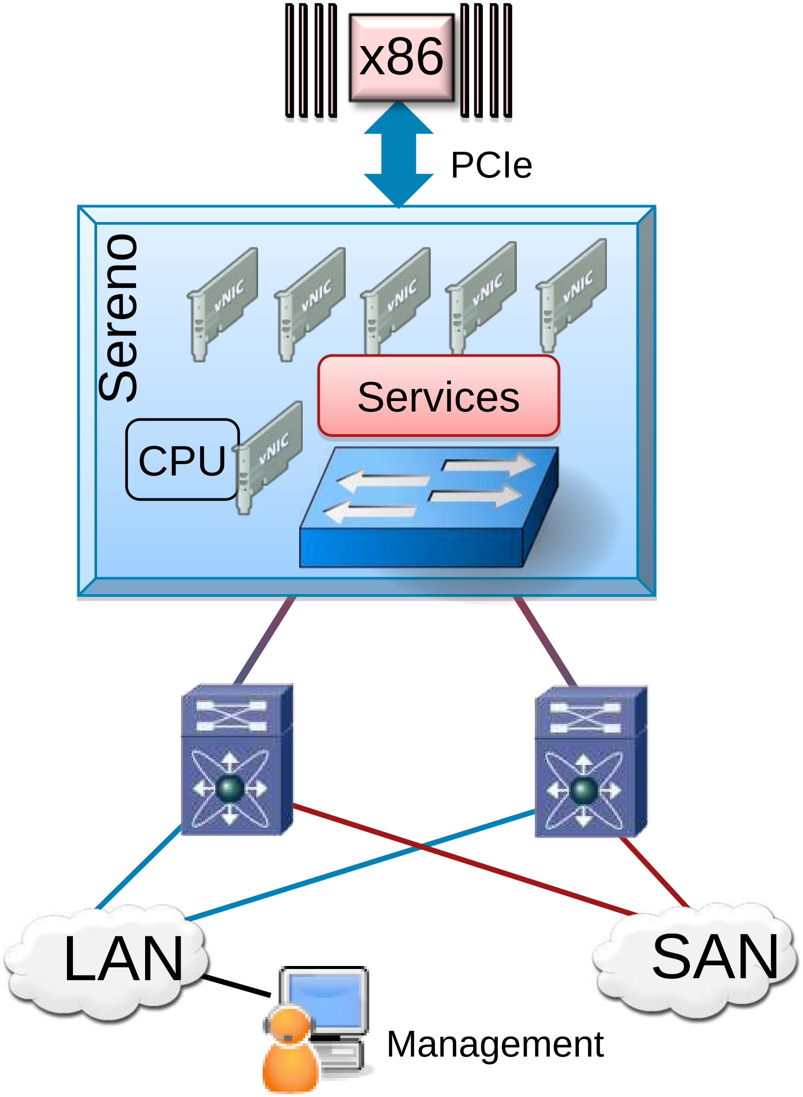
Network Services

- Services applied at the network edge

  per-flow tracking, steering, prioritization in hardware

  per-vNIC encapsulation and redirection for firewall, load balancing, and others

  Scalable point of network services application



Sereno Physical

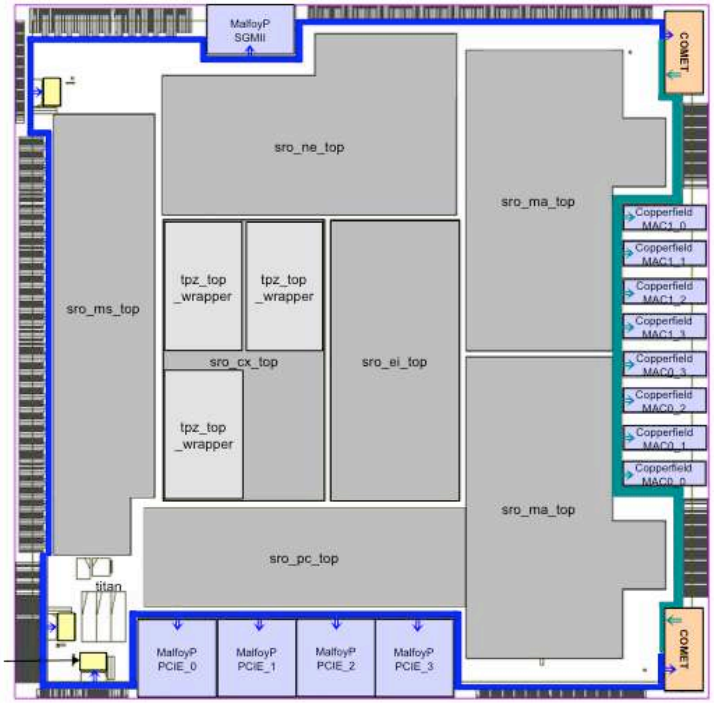
- Technology: TI 65nm

  die size: $136mm^2$

  package: organic FC BGA, 784 pins

  logic gates: 17M, SRAM: 37 Mbits

  power: 16W, when all interfaces active



- Interfaces

  PCIe x16 Gen2

  8x10Gbps XFI (40GE/10GE/1GE capable)

  1x1GE SGMII (local management/BMC port)

  32 bit DDR/Flash (local data structures)

  Misc: UART/GPIO/I2C/SPI/MDIO/JTAG

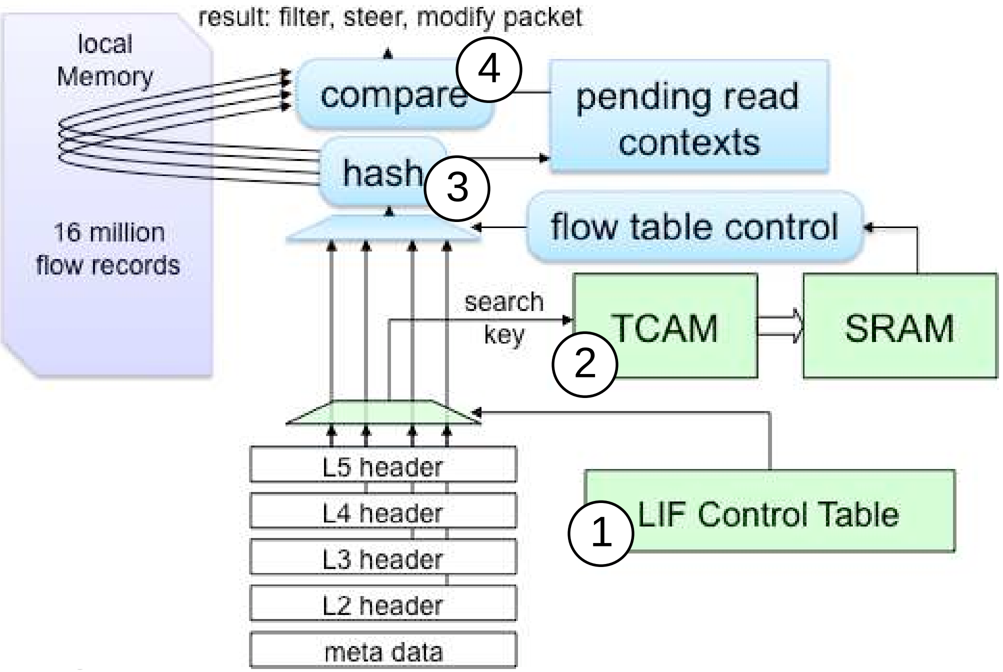
Technical Dive on Select Features

The next slides will examine a few of Sereno’s more interesting hardware mechanisms

- Packet Classifier and Flow Table mechanisms to enable network services feature
- Transmit latency reduction scheme
- Transmit scheduler to precisely control 256 vNICs

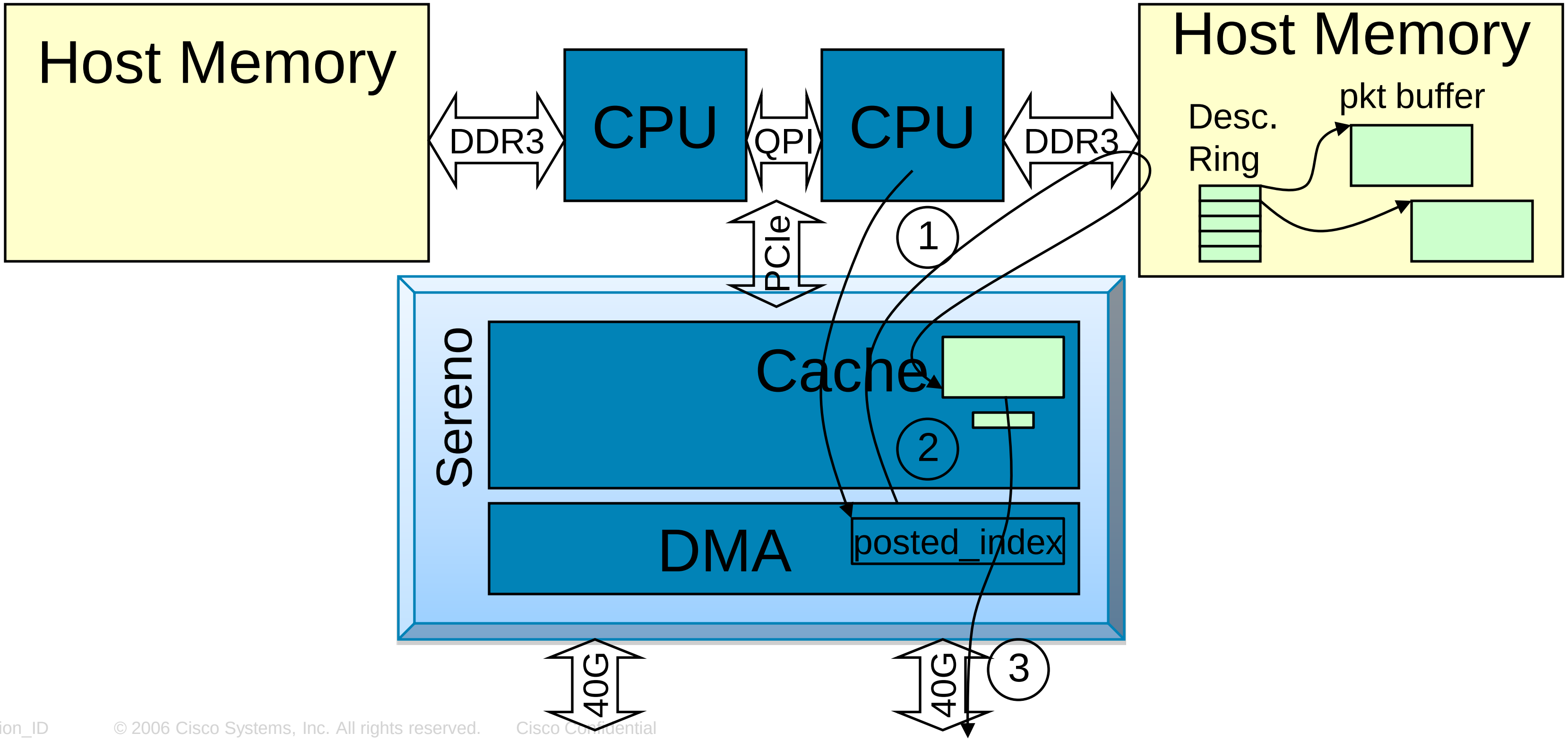
Packet Classification and Flow Tables

1. per-interface packet search criteria (TCAM key)
2. TCAM identifies flow types, result is 8-tuple key per flow table, up to 4 tables per packet
3. 40-byte key is hashed to 24-bit table index
4. final result will filter, steer, or modify packet



Packet Transmit Latency Reduction

- Goal: reduce latency using familiar descriptor ring model
  1. Host CPU builds packet & descriptor, posts index and hint
  2. Sereno fetches descriptor and prefetches data to cache
  3. Packet is transmitted from Sereno cache without host latency



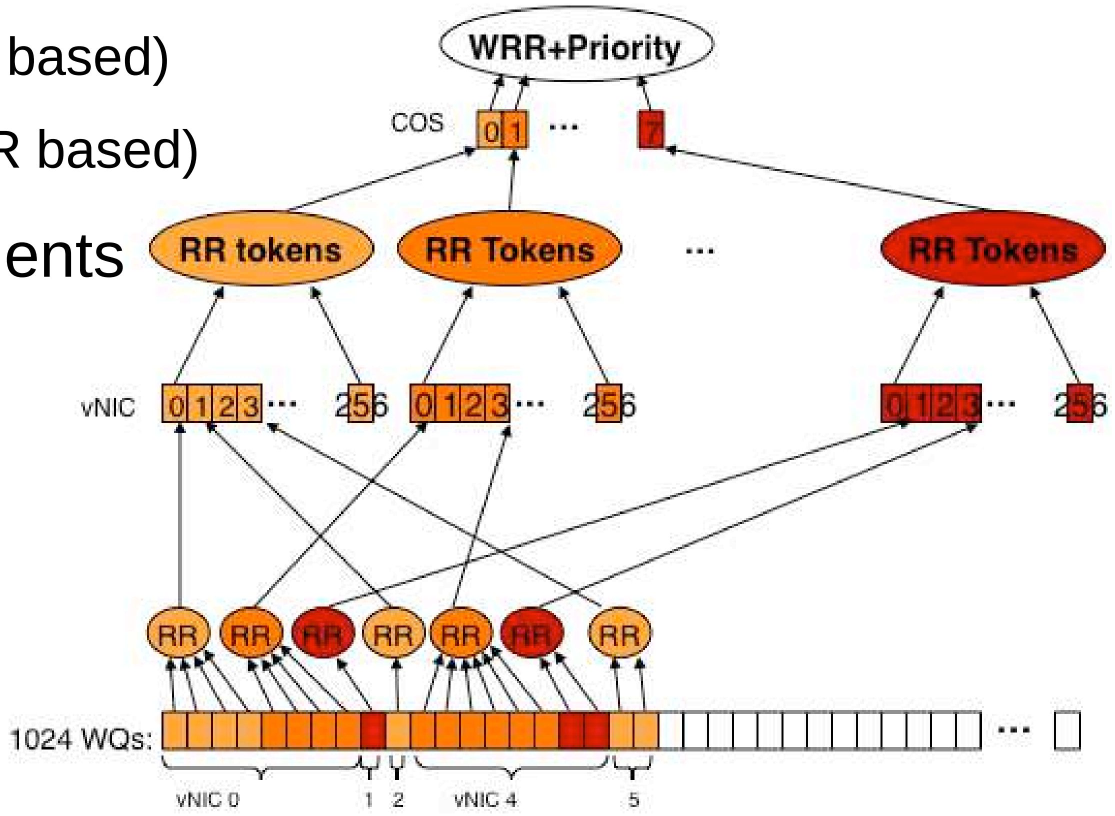
vNIC Transmit Scheduling

Ethernet, Storage, and Management traffic have different priorities and bandwidth needs

- 1024 queues scheduled across 256 vNICs
  1. Class of Service chosen (WRR and priority based)
  2. vNIC chosen (RR based)
  3. queue chosen (RR based)
- Rate based adjustments

  per-vNIC rate limits

  per-vNIC CIR



Switch to software integration

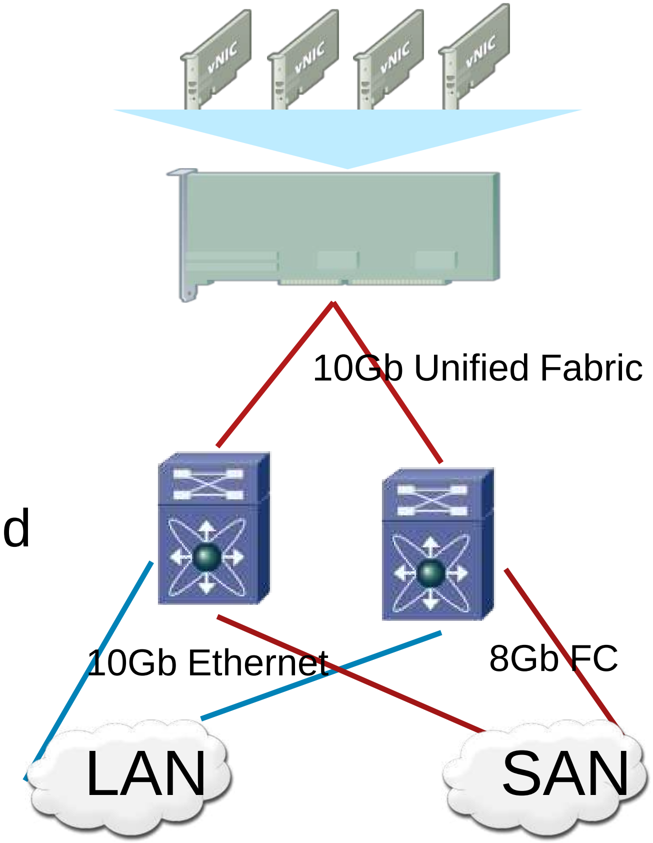
The Cisco adapter was borne out of the system view of a data center

In the next few slides we will show

- Technology direction for improved device management
- Technologies which will help system scaling
- Technologies for accelerating throughput and greater scalability
- Results seen from these experiments

Device management today

- Scale out success created a topology management nightmare
  - SR-IOV makes it worse
  - Each IO path needs network configuration & host configuration
  - Each PCIe endpoint has no knowledge of the network and vice versa



SR-IOV enum

Network config MTU, VLAN

Fabric config

Device management in Sereno

- Configure from the network

  Firmware places “shaped” devices on PCIe bus *(e.g. Ethernet/FCoE, QOS, MTU etc)*

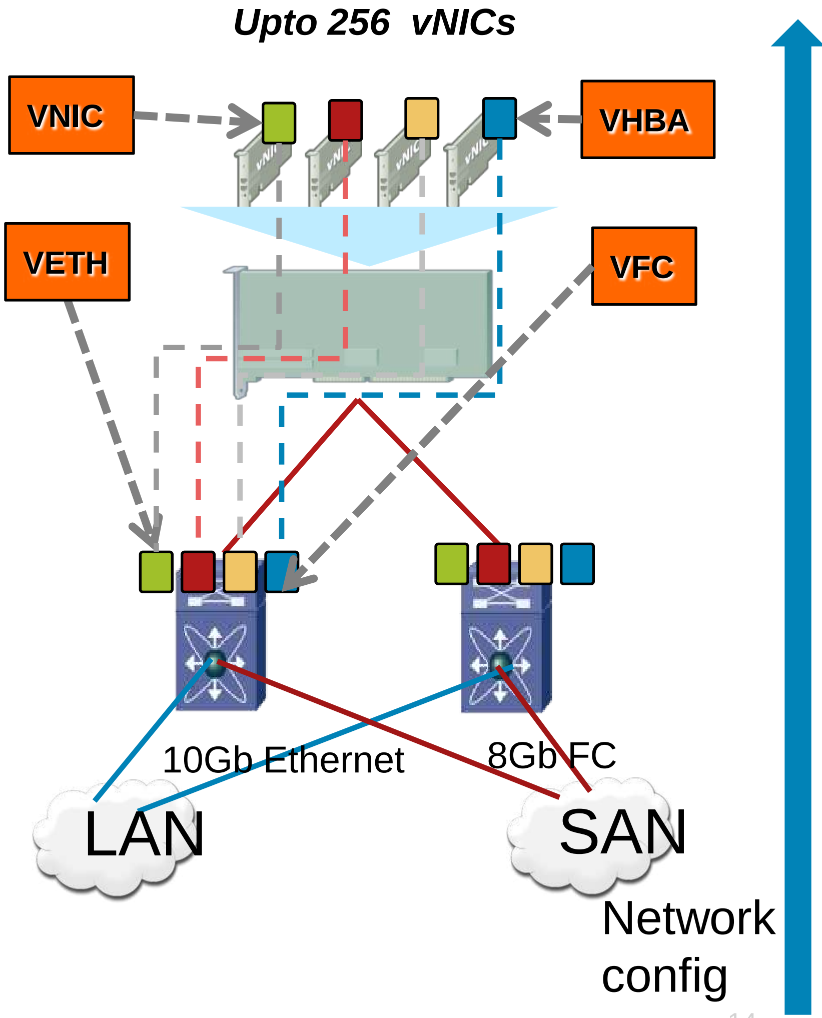
  Drivers “discover” fabric properties *(network address, MTU)*

  Host drivers maintain compatibility across generations

- Single flow of configuration

  Each NIC/HBA has a network presence and host presence

  Properties like automatic failover is built into fabric interface *(no bonding or trunking drivers)*



Problem : Interrupt storms and multiple devices

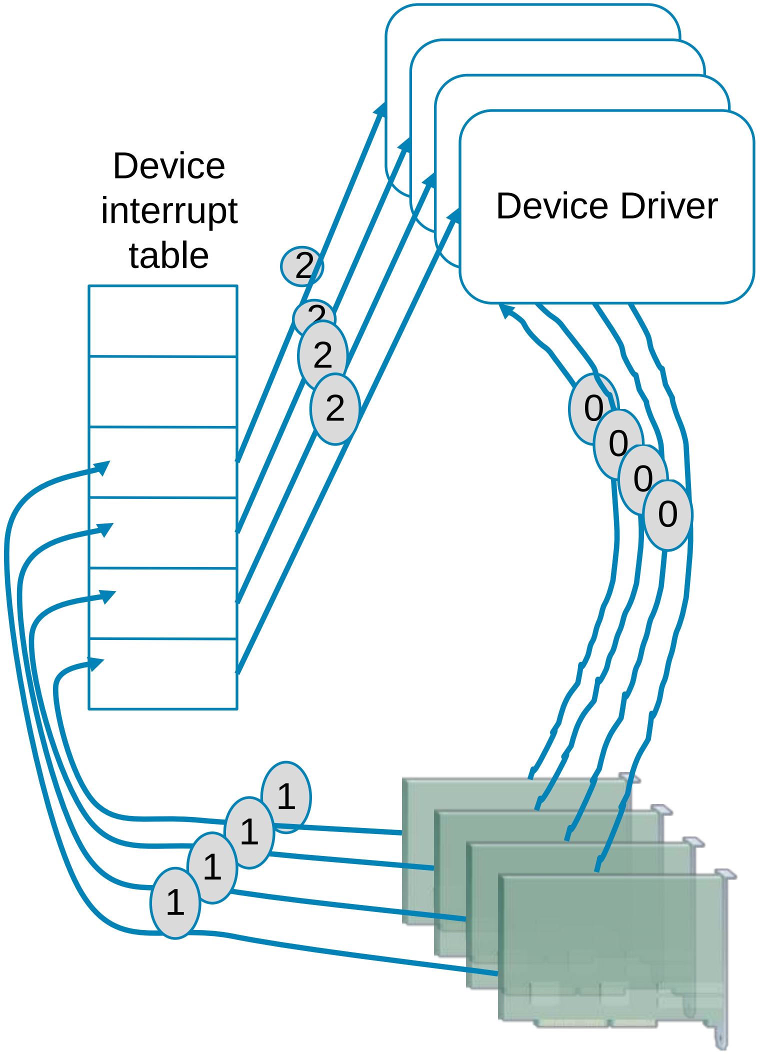
- Virtualization and Passthrough

  VM density of 50 with 8 MSI-X vectors per device creates 400 vectors on a 16-40 core system

  Available host vectors typically configured to be an order of magnitude less

  Each passed through device generates an interrupt on packet event causing ctx switch

  Broadcast packets can generate one interrupt per device *(device count > core count typically)*



Passed through Virtual Functions

Solution : Group Interrupts

- Sereno solution

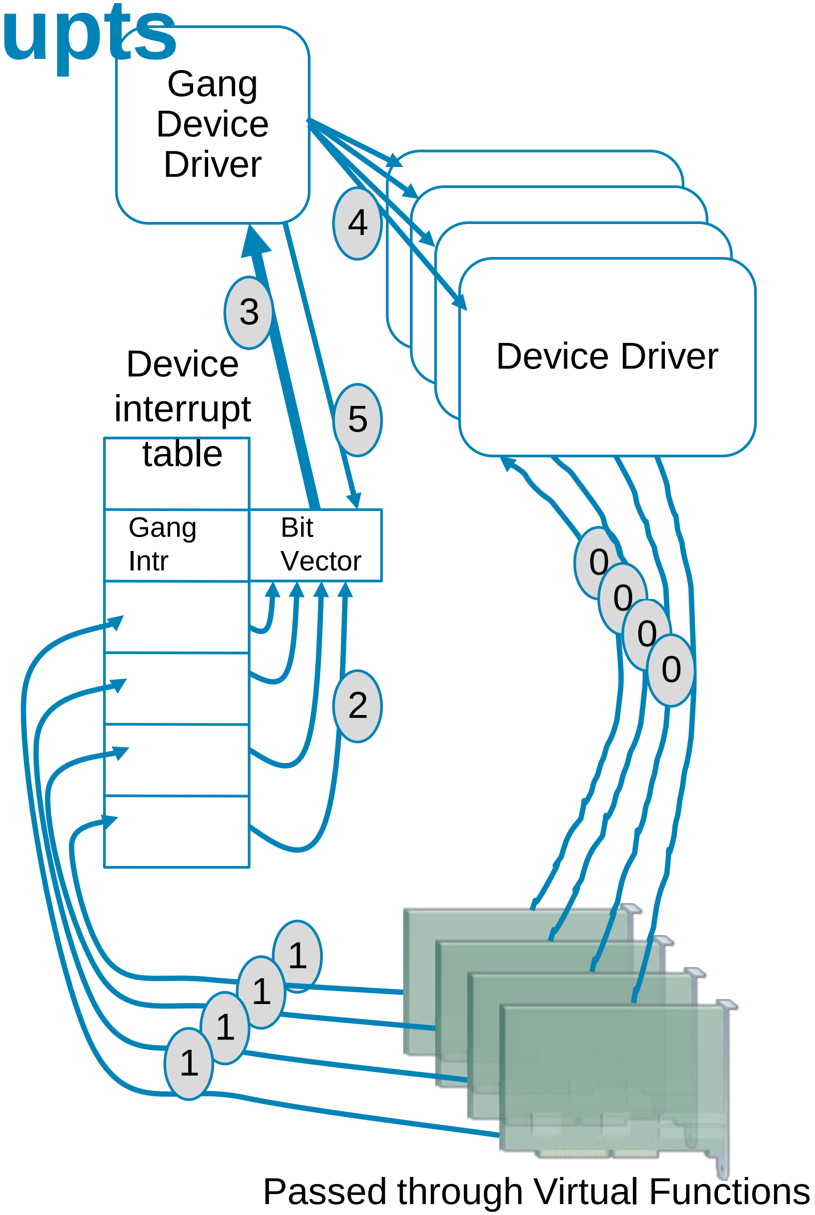
  Driver “register” for a gang instead of a single vector

  Sereno “gangs” all members together and DMA’s membership and a single vector

  N-1 interrupt ctx switches avoided

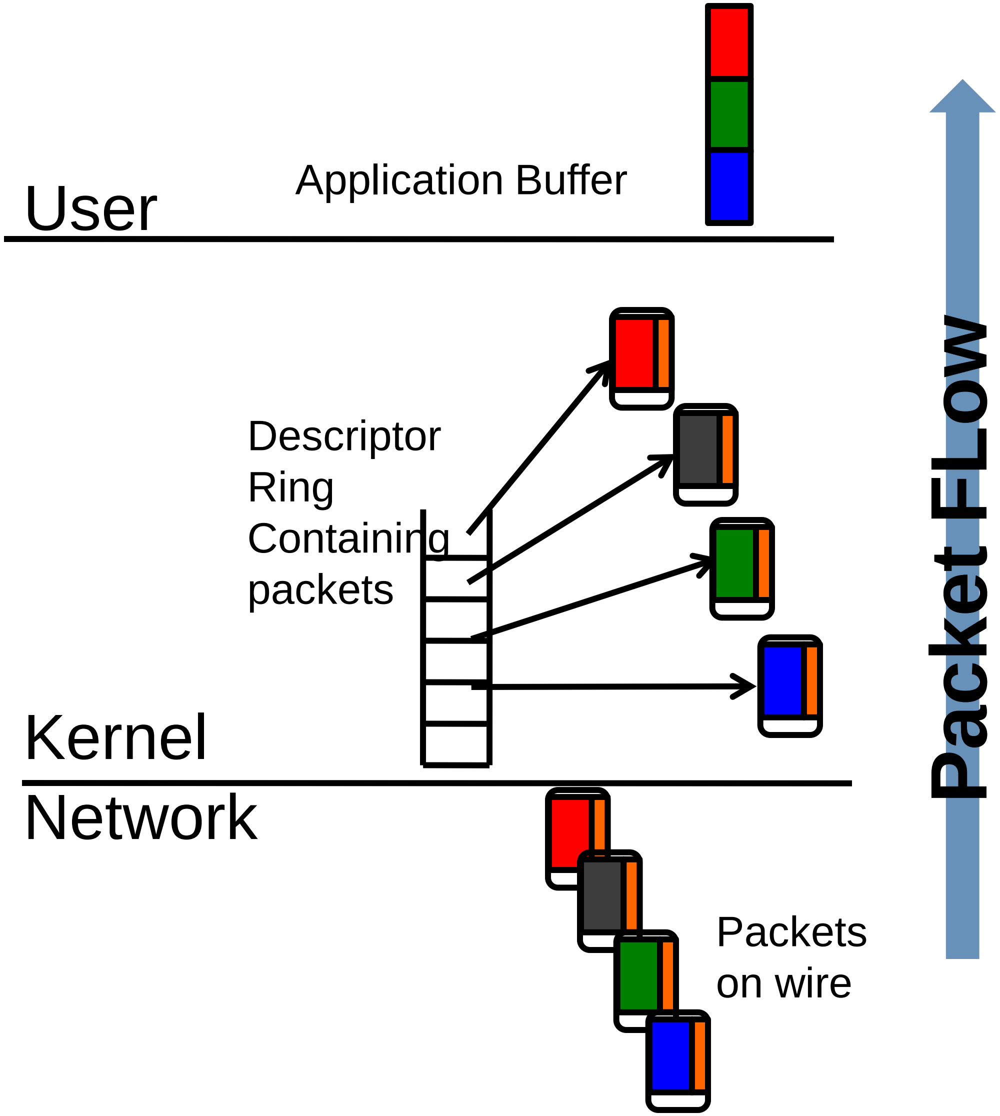
  Gang irq handler dispatches all driver handlers of gang members

- Request for OS and Hypervisor support for IRQ ganging



Traditional Network Buffer flow

- Recv is expensive
  - Requires multiple context switches and copies
  - Works well with traditional synchronous sockets
  - All flow classification and redirection is done in the kernel with locking



Solution : Zero Copy Receive

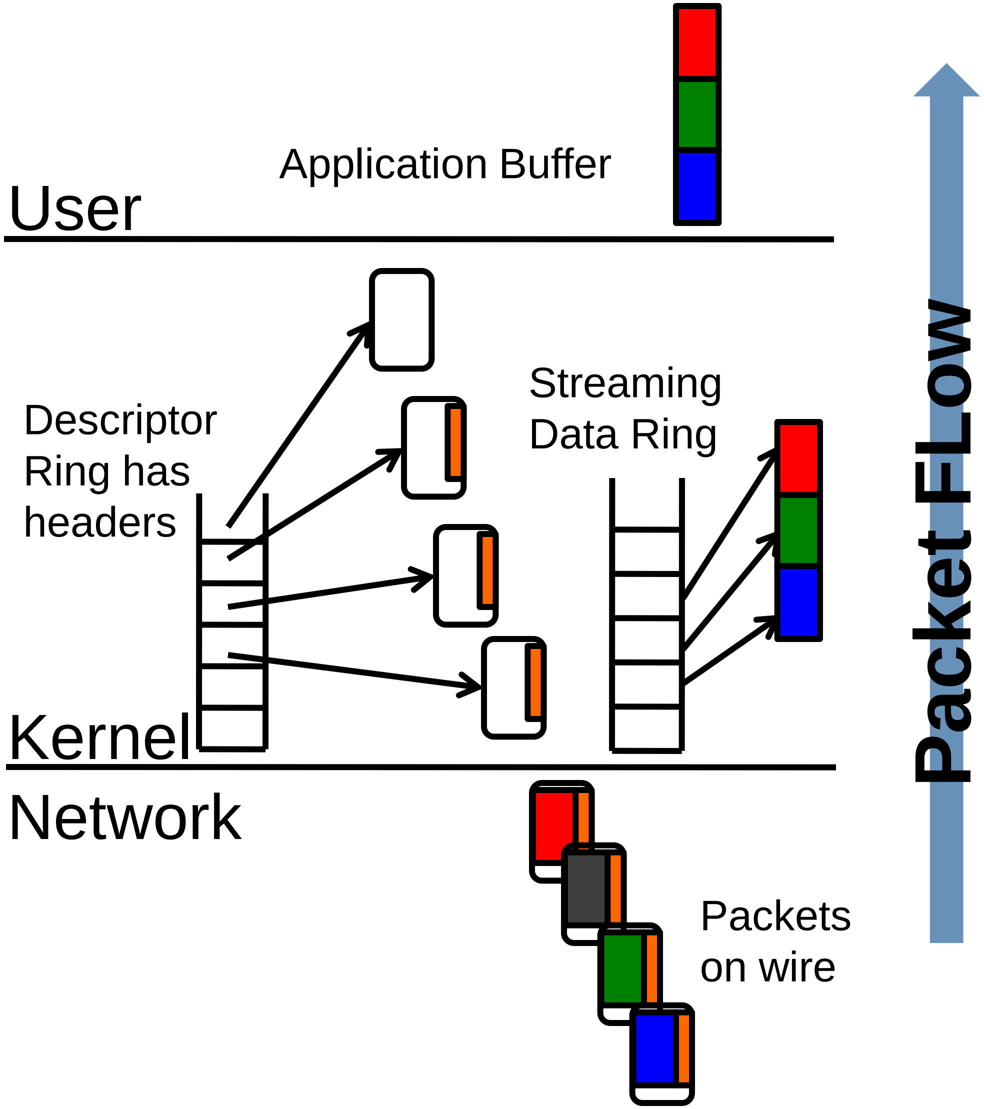
- Recv is almost zero

  Sereno does classification of packet flows to queues. The packets are header split as well

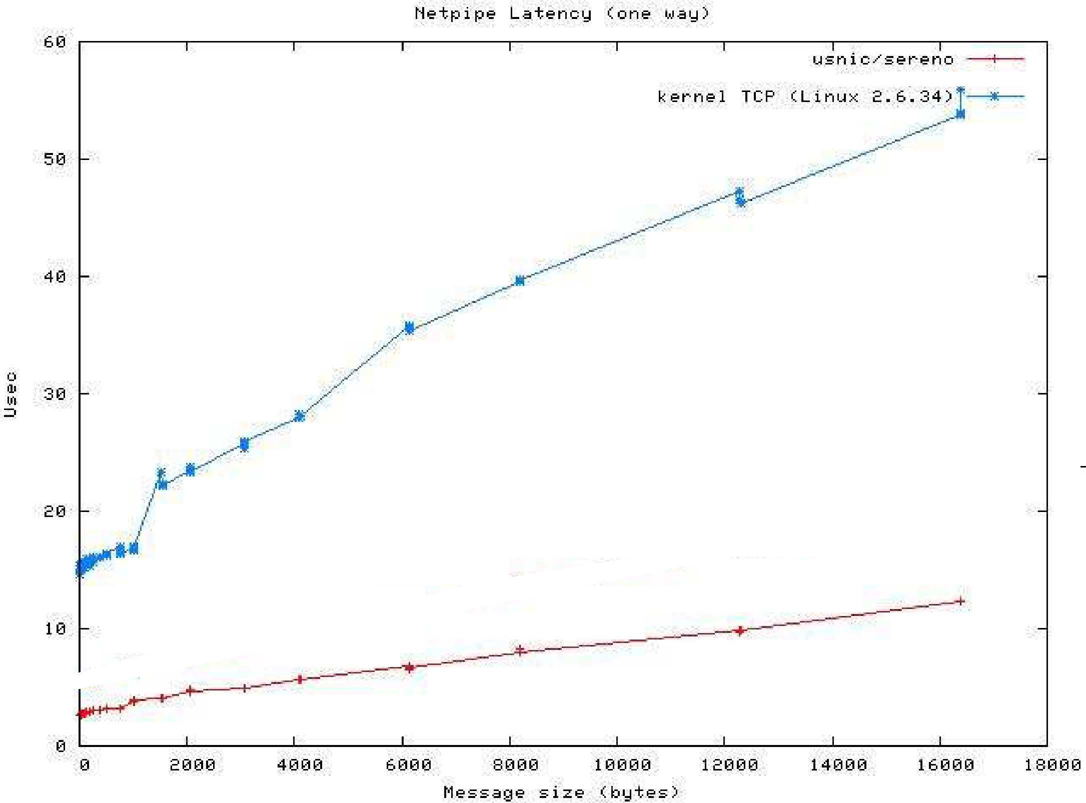
  Streaming ring is mapped directly to application space

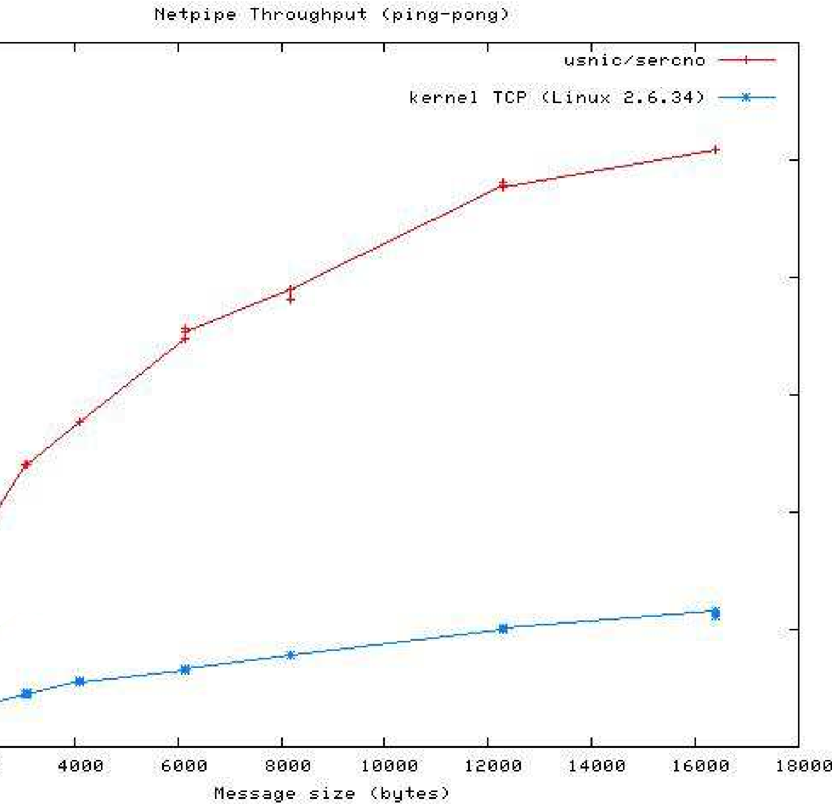
  Packets are laid out in sequence order into streaming ring

  Kernel thread processes headers, patches and signals completions



Sereno performance with netpipe in userspace


Netpipe Latency (one way)
usnic/sereno
kernel TCP (Linux 2.6.34)
Usec
Message size (bytes)


Netpipe Throughput (ping-pong)
usnic/sereno
kernel TCP (Linux 2.6.34)
MBps
Message size (bytes)