Network Interface Cards

ECE/CS598HPN

Radhika Mittal
Network Interface Card (NIC)

- Physical layer processing
- Some link layer processing
- Direct Memory Access (DMA) for copying data.
- Mechanism to trigger interrupts.
Modern NICs do much more than this.
NIC Features: Protocol Offload

• TCP Segmentation offload
  • Split a large outgoing packet into MTU-sized packets and assign appropriate headers.

• Checksum Offload
  • TCP / UDP / IPv4 checksum computation.

• Large Receive Offload
  • Combine multiple MTU-sized packets for the same connection into a single large packet.
NIC Features: Packet Steering

• Receive Side Scaling
  • Load balance incoming packets across different queues.
  • Hash of packet header fields mapped to queue index.
  • Can pick which queue corresponds to which index.

• Flow Director
  • Maintain explicit mapping between packet header fields and queue.
  • Other actions including dropping and incrementing counters.
NIC Features: Virtualization

Virtual Machine Manager (VMM)

Physical Host Hardware
NIC Features: Virtualization
NIC Features: Virtualization
NIC Features: Virtualization

- SR-IOV (Single Root I/O Virtualization)

  - Provides the hardware abstraction of a ‘virtual function’ (VF).
  
  - Multiple ‘virtual functions’ mapped to a single physical function.
  
  - VMM maps a virtual function space to a specific VM.
NIC Features: Virtualization

• SR-IOV (Single Root I/O Virtualization)
  • Share a single physical port across multiple VMs.

• VMDq (Virtual Machine Device Queues)
  • Sort packets across VM specific queues based on MAC address and VLAN tags.
  • Round-robin across VM queues.

• VT-d (Virtualization technology for directed I/O)
  • DMA support for VMs, manage interrupts for VMs, protection and isolation across VMs for I/O operations.
NIC Features: Tunneling

- Examples:
  - VXLAN:
  - NVGRE:

- Offload encapsulation/decapsulation.
- Ability to parse tunneled information.
Limitations

• Lack of flexibility and fine-grained control.
  • E.g. TSO offload can be useless without VXLAN support.
  • Even minor fixes can take years.

• Resource constraints.
  • Limited memory (packet buffers, flow table size, etc).
  • E.g. Flow Director allows only 8K flow entries.
SoftNIC: A Software NIC to Augment Hardware

Sangjin Han, Keon Jang, Aurojit Panda, Shoumik Palkar, Dongsu Han, Sylvia Ratnasamy
SoftNIC Design Goals

• Programmability and extensibility

• Application performance isolation

• Backwards Compatibility
Architecture
Packet Processing Example
Resource Scheduling

• Allocate both processor and bandwidth resources.

C1, C3: high priority, 1 Gbps
C2, C4: low priority, no limit
Per VM: 5 Gbps limit
Implementation

• Over DPDK.

• Dedicate a small number of cores to SoftNIC.
  • Multi-core scaling achieved by associating each SoftNIC core with different set of queues.
  • Requires peers to ensure packets from same flow go to the same queue.

• Supports different packet I/O interfaces at vports for user-space / kernel-bypass applications and kernel.
  • Implement a kernel driver, requiring no modification to kernel.

• Polling to check for packets from vport and pport.

• Batching to amortize software processing overheads.
Evaluation

- T1: Hosts (raw packet)
- T2: Hosts (kernel UDP)
- T3: VM-to-VM (inter-machine)
- T4: VM-to-VM (intra-machine)
Case Studies

• If NICS do not understand tunneling format, cannot support TSO for “inner” TCP frames.

• SoftNIC can be used to augment the TSO/LRO feature in these cases.
Case Studies

• NIC supports a limited number of “rate-limiters” – few hundreds.
  • There may be thousands of flows.

• SoftNIC can be used to implement a scalable rate limiter.
Case Studies

- Flow Director directs packets with specific header fields to specific queues.
  - Can only support 8K entries.

- SoftNIC can support almost unlimited flow entries using system memory.
Case Studies

• Scaling legacy applications: send packets to different cores based on hash of packet header fields.

• RSS (NIC feature) is too limiting.

• SoftNIC can be used to provide such scaling.
Your thoughts?

• What did you like about the paper?
• What are its limitations?
• Other ways of achieving flexible NIC offload?
Next few classes

• Host SDN and network virtualization in multi-tenant datacenters.

• Two case-studies:
  • Google (SNAP)
  • Microsoft (AccelNet)

• Other forms of programmable NICs
  • FPGA-based NICs (AccelNet)
  • NICs with general-purpose compute (FlexTOE)
Update in schedule

• Nov 16th: no class; optional reading on course website.

• Student presentations on Nov 18th and Nov 30th
  • Present a relevant paper of your choice
    • A paper that is related to the topics we covered, but not part of your reading list (can select a paper from the “optional” list).
    • 6 minute presentation with 2mins for Q/A.
      • What problem is the paper trying to solve?
      • How does it solve it at a high-level / what’s the key idea?
      • Key result.
  • Watch out for an email with a sign-up sheet.
    • Select a slot and a paper of your choice on a first-come-first serve basis.