The World Needs a New NIC (and it needs to run Homa)

John Ousterhout
Overview

- New transport protocol: Homa
- How to replace TCP in the datacenter?
- Datacenters will require new NICs
  - Time to remove transport protocols from the OS
  - Implement Homa on the NIC
Mission: Low Latency RPC

- **Ultimate goal**: 2-3 µs round-trips in datacenters
  - Assumptions:
    - 3-layer switching structure (10 switch traversals in round-trip)
    - 1 µs speed-of-light delay round-trip
    - Minimal degradation for short messages (2-3x) at high network loads

- **Deployed today**:
  - 35 µs best-case (unloaded)
  - > 1 ms tail latency under load

- **Impediments to low latency**:
  - Operating system kernel
  - TCP
  - Virtualization/management
  - PCIe interconnect
  - Dispatching/load-balancing

June 7, 2019
Impediment #1: OS Kernel

- **Linux adds about 10 µs overhead:**
  - Homa RPC via Linux kernel: 14.6 µs
  - Homa RPC w. kernel bypass (RAMCloud): 4.0 µs
  (2 servers attached to same TOR)

- **Kernel bypass is essential, but:**
  - Today must implement protocols in applications
  - Other issues, such as dispatching
Impediment #2: TCP Protocol

- **4-5 µs extra round-trip overhead:**
  
<table>
<thead>
<tr>
<th></th>
<th>TCP</th>
<th>Homa</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudLab xl170</td>
<td>28 µs</td>
<td>23 µs</td>
</tr>
<tr>
<td>CloudLab m510</td>
<td>32 µs</td>
<td>28 µs</td>
</tr>
<tr>
<td>CloudLab c220gl</td>
<td>41 µs</td>
<td>37 µs</td>
</tr>
</tbody>
</table>

- **100-1000x slowdown under load:**
  - Congestion control requires buffer occupancy

- **Problematic features:**
  - Connection-oriented (large amounts of state)
  - Stream-oriented
Impediment #3: Virtualization/Mgmt

- **Features such as:**
  - Virtual host addresses
  - Performance isolation (rate limiting)
  - Live migration

- **Current implementations in software:**
  - VMware Open vSwitch
  - Google Andromeda

- **Significant latency penalty: > 10 µs round-trip?**
Impediment #4: PCIe Bus

- PCIe bus has high latency: 100s of ns
- Multiple transits for each host-NIC exchange
- Synchronous: locks up CPU
- Overall cost of host-NIC communication:
  - ~ 0.5 μs per packet sent or received
  - ~ 2 μs per round trip

PCle bus accounts for half of total RPC time under kernel bypass!
Impediment #5: Load Balancing

**Choice #1: dispatch thread**

- Extra latency for worker handoff
- Dispatch thread is throughput bottleneck

**Choice #2: sharding**

- NIC dispatches based on shard info provided by sender
- Prone to hot spots
Things Will Get Worse

● The future:
  § Faster networks (40 Gbps → 100 Gbps → ??)
  § Extremely high packet rates for small messages
  § CPUs not getting faster
  § More and more cores generating/handling traffic

● Software-based packet handlers increasingly problematic:
  § Centralized handlers can’t handle traffic
  § Distributed software approaches challenging
    ● Load balancing
    ● Synchronization hot spots
Homa Transport Protocol

- **Solves congestion problem**
  - Uses in-switch priority queues to prioritize short messages
  - Receiver-driven flow control

- **Other attractive features:**
  - Simpler than TCP
  - Message-oriented (not streaming)
  - Connectionless (state only for active RPCs)

- **Performance >> TCP**
  - Small RPCs: 5 µs (small networks, kernel bypass)
  - Tail latency slowdown ~ 3x at 80% bandwidth utilization

How to make Homa the standard for datacenter communication?
Bring the Network to the Cores?

- Distribute raw network on-chip
- Simple NIC for each core
- Special user-space instructions to send/receive packets
- Run Homa in apps on each core

Problems:
- Requires processor modifications
- Homa needs centralized state for network link
- No solution for virtualization/management
Better Solution: New NIC

NIC Features:

- Transport protocol implemented by NIC
- Kernel bypass
- Message-based interface on host side (no packets!)
- Dispatching/load-balancing
  - E.g., pick idle thread
- Virtualization/mgmt (zero overhead)
- Encryption (and authentication?)
- Replace PCIe for NIC-app communication (eventually)
Why Homa Instead of TCP?

- 100-1000x better tail latency for small messages
  - Homa has better congestion control
- Homa is simpler than TCP
- NIC can’t do dispatching with TCP
  - Dispatchable units (messages) must be visible
  - TCP: stream based (no message boundaries)
  - Homa: message based
- TCP requires expensive state
  - TCP: connection oriented (separate state for each source-destination pair)
  - Homa: no connections; state only for active RPCs
- Kernel bypass awkward/expensive with TCP
  - TCP: kernel (and hypervisor?) interaction for each new connection
  - Homa: kernel interaction once per application: single socket for all communication
How Do We Get There?

- Today’s “smart NICs” inadequate
- Too large a project for a university?
  - But Nick McKeown has ideas…
- Special-purpose Homa NIC unlikely to come from industry
  - Companies unlikely to bet on Homa until it is more established
- Best hope: a new programmable NIC
  - Looks more like a switch than a traditional NIC?
  - High performance datapath
  - Customizable/programmable (> P4)
- Risks:
  - Proprietary protocol implementations
  - Special-purpose TCP implementation makes new protocols impossible
Conclusion

- Trend towards special-purpose hardware
- Opportunity in the datacenter: new NIC
  - As important as FPUs, GPUs, and TPUs
- Should support complete transport protocol implementation(s)
- Homa is the right protocol