Smart (Programmable) NICs

ECE/CS598HPN

Radhika Mittal
Microsoft Case Study

Azure Accelerated Networking: SmartNICs in the Public Cloud

NSDI’18

Slides borrowed from the NSDI talk
Overview

- Azure and Scale
- Recap: Virtual Filtering Platform and Host SDN
- Why Accelerated Networking? Scaling up SDN
- Hardware Choices
- Azure SmartNIC
- Accelerated Networking in Azure: Results
- Experiences and Lessons Learned
- Conclusion and Future
Microsoft Azure

App Services
- cloud services
- mobile services
- web apps
- integration
- identity
- hpc
- service bus
- media
- analytics

Data Services
- SQL database
- Data Lake
- table
- blob storage

Infrastructure Services
- virtual machines
- virtual network
- vpn
- traffic manager
- cdn
<table>
<thead>
<tr>
<th>Statistic</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fortune 500 using Microsoft Cloud</td>
<td>&gt;85%</td>
</tr>
<tr>
<td>New Azure customers a month</td>
<td>&gt;120,000</td>
</tr>
<tr>
<td>Azure Active Directory Orgs</td>
<td>&gt;9 MILLION</td>
</tr>
<tr>
<td>Azure Active Directory authentications/week</td>
<td>&gt;18 BILLION</td>
</tr>
<tr>
<td>Azure Event Hubs events/week</td>
<td>&gt;3 TRILLION</td>
</tr>
<tr>
<td>Azure storage objects</td>
<td>&gt;60 TRILLION</td>
</tr>
<tr>
<td>Azure VMs</td>
<td>&gt;50% of</td>
</tr>
<tr>
<td>Azure VM requests/day</td>
<td>&gt;900 TRILLION</td>
</tr>
<tr>
<td>Azure DB requests/day</td>
<td>&gt;110 BILLION</td>
</tr>
</tbody>
</table>
Overview

- Azure and Scale
- Recap: Virtual Filtering Platform and Host SDN
- Why Accelerated Networking? Scaling up SDN
- Hardware Choices
- Azure SmartNIC
- Accelerated Networking in Azure: Results
- Experiences and Lessons Learned
- Conclusion and Future
Virtual Filtering Platform (VFP) Azure’s SDN Dataplane

- Virtual switch for Hyper-V / Azure
Key Primitive: Match Action Tables

- VFP exposes a typed Match-Action-Table API to the agents/controllers
- One table ("Layer") per policy
- Inspired by OpenFlow and other MAT designs, but designed for multi-controller, stateful, scalable host SDN applications
Unified Flow Tables – A Fastpath Through VFP

Rule Lookups (Expensive)

First Packet

SLB Decap
SLB NAT
VNET
ACL
Metering

VFP

Second+ Packet

Transposition

Hash Lookups (Cheap)

Flow | Action
--- | ---
1.2.3.4 = 3.4.5.6, 6/30/2 = 30 | Decap, DNAT, Rewrite, Meter
Overview

• Azure and Scale
• Recap: Virtual Filtering Platform and Host SDN
• Why Accelerated Networking? Scaling up SDN
• Hardware Choices
• Azure SmartNIC
• Accelerated Networking in Azure: Results
• Experiences and Lessons Learned
• Conclusion and Future
Scaling Up SDN: NIC Speeds in Azure

- 2009: 1Gbps
- 2012: 10Gbps
- 2015: 40Gbps
- 2017: 50Gbps
- Soon: 100Gbps?

We got a 50x improvement in network throughput, but not a 50x improvement in CPU power!
Host SDN worked well at 1GbE, ok at 10GbE... what about 40GbE+?
Traditional Approach to Scale: ASICs
Example ASIC Solution: Single Root IO Virtualization (SR-IOV) gives native performance for virtualized workloads
Hardware or Bust

• SR-IOV is a classic example of an “all or nothing” offload – its latency, jitter, CPU, performance benefits come from skipping the host entirely
• If even one widely-used action isn’t supported in hardware, have to fall back to software path and most of the benefit is lost even if hardware can do 99% of the work
• Other examples: RDMA, DPDK, ... a common pattern
• This means we need to consider carefully how we will add new functionality to our hardware as needed over time
Overview

• Azure and Scale
• Recap: Virtual Filtering Platform and Host SDN
• Why Accelerated Networking? Scaling up SDN
• Hardware Choices
• Azure SmartNIC
• Accelerated Networking in Azure: Results
• Experiences and Lessons Learned
• Conclusion and Future
Silicon alternatives

Option 5: Don’t offload at all, instead make SDN more efficient with e.g. poll-mode DPDK
CPU vs. FPGA

CPU: temporal compute

FPGA: spatial compute
What is an FPGA, Really?

- Field Programmable Gate Array
- Chip has large quantities of programmable gates – highly parallel
- Program specialized circuits that communicate directly
- Two kinds of parallelism:
  - Thread-level parallelism (stamp out multiple pipelines)
  - Pipeline parallelism (create one long pipeline storing many packets at different stages)
Our Solution: Azure SmartNIC (FPGA)

- HW is needed for scale, perf, and COGS at 40G+
- 12-18 month ASIC cycle + time to roll new HW is too slow
- To compete and react to new needs, we need agility – SDN
- Programmed using Generic Flow Tables
  - Language for programming SDN to hardware
  - Uses connections and structured actions as primitives
FPGAs: Internal Q&A

1. Aren’t FPGAs much bigger than ASICs?
2. Aren’t FPGAs very expensive?
3. Aren’t FPGAs hard to program?
4. Isn’t my code locked in to a single FPGA vendor?
5. Can FPGAs be deployed at hyperscale? Are they DC-ready?
Overview

- Azure and Scale
- Recap: Virtual Filtering Platform and Host SDN
- Why Accelerated Networking? Scaling up SDN
- Hardware Choices
- Azure SmartNIC
- Accelerated Networking in Azure: Results
- Experiences and Lessons Learned
- Conclusion and Future
### Azure Accelerated Networking

- **Highest bandwidth VMs of any cloud so far...**
  - Standard compute VMs get up to 32Gbps
  - Stock Linux VM with CUBIC gets 30+Gbps on a single connection

- **Consistent low latency network performance**
  - Provides SR-IOV to the VM
  - 5x+ latency improvement – sub 15us within tenants
  - Increased packets per second – Up to 25M PPS (12M forwarding) for DPDK VMs
  - Reduced jitter means more consistency in workloads

- **Enables workloads requiring native performance to run in cloud VMs**
  - >2x improvement for many DB and OLTP applications
AccelNet
Comparative Results

VM-VM Throughput, Gbps

VM-VM Latency, μs

VM-VM Tail Latencies, μs
AccelNet
Comparative Results

VM-VM Throughput, Gbps

VM-VM Latency, μs

VM-VM Tail Latencies, μs
Overview

• Azure and Scale
• Recap: Virtual Filtering Platform and Host SDN
• Why Accelerated Networking? Scaling up SDN
• Hardware Choices
• Azure SmartNIC
• Accelerated Networking in Azure: Results
• Experiences and Lessons Learned
• Conclusion and Future
Serviceability is Key

- All parts of this system can be updated, any of which require us to take out the hardware path – or VM can be live migrated
  - FPGA image, driver, GFT layer, Vswitch/VFP, NIC PF driver
Changes, Changes, Changes

A few examples of many...

- TCP and protocol state machines
- Complex packet forwarding and duplication actions
- New SDN actions
- Accelerating the offload path
- Line rate diagnostics and monitoring
Changes, Changes, Changes

A few examples of many...

• TCP and protocol state machines
• Complex packet forwarding and duplication actions
• New SDN actions
• Accelerating the offload path
• Line rate diagnostics and monitoring
Lessons Learned

- Design for serviceability upfront
- Use a unified development team
- Use software development techniques for FPGAs
- Better perf means better reliability
- HW/SW co-design is best when iterative
- Failure rates remained low – FPGAs in the DC were reasonably reliable
- Upper layers should be agnostic of offloads
- Mitigating Spectre performance impact
Pros:

• Discusses pros and cons of different offload mechanisms.
• Achieve flexibility with hardware speed.
• Focus on maintainance and upgradability.
• Reduced CPU utilization.
• Exception packets handled in software – FPGA need not remember all policies.
• Mature well-tested solution.
Your Opinions

Cons:
Cons:

- No performance comparison (shown) with other smartNIC solutions.
- FPGA-equipped smartNIC vs programmable switches?
- Area and cost of FPGA over ASIC (compared to other alternatives).
- FPGA development is harder than software.
- Recompiling an FPGA is more “expensive” than recompiling software.
- No evidence that the design can scale to 100Gbps.
- Security vulnerabilities?
- Viable only at Azure scale.
Your Opinions

Ideas:

• How to divide functionality between software, programmable NICs, and programmable switches?
• Use FPGAs for offloading more features.
• More evaluation under high workload.
• Can one design ”custom” FPGAs with smaller area and cost?
• Compare FPGA-based approach with other alternatives (e.g. SoftNIC) – in terms of both performance and flexibility.
• A compiler that translates code written in higher-level language to FPGA.
Your Opinions

Software or Hardware?