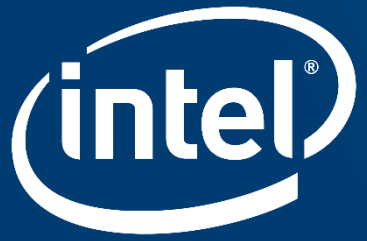


NETWORK I/O VIRTUALIZATION AND ITS FUTURE

Anjali Singhai Jain

Alexander Duyck

Parthasarathy Sarangam

Nrupal Jani

April 2017, NetDev 2.1

Progress, far from consisting in change, depends on retentiveness. When change is absolute there remains no being to improve and no direction is set for possible improvement: and when experience is not retained, as among savages, infancy is perpetual. Those who cannot remember the past are condemned to repeat it.

- George Santayana

Agenda

- **Evolution of Network I/O Virtualization in Linux**
- **SR-IOV Pros and Cons**
- **Future of Network I/O Virtualization**
- **Comparison**
- **Conclusion**

Evolution of Network I/O Virtualization

- SW Emulated (e1000)
  - Pros: A driver already existed based on actual device. “It just works!”
  - Cons: Very Basic designed for Single Core system. Optimized for HW.
- Para Virtualized (Virtio)
  - Pros: Optimized for SW Emulation, support for 64K frames, East-West traffic.
  - Cons: Small packet performance suffers as we still have to take traps, memcpy, etc.
- Direct Assignment
  - Pros: Existing driver can be used, direct access to device w/o traps.
  - Cons: Security issues, scaling issues
- SR-IOV Contd.

SR-IOV Pros and Cons

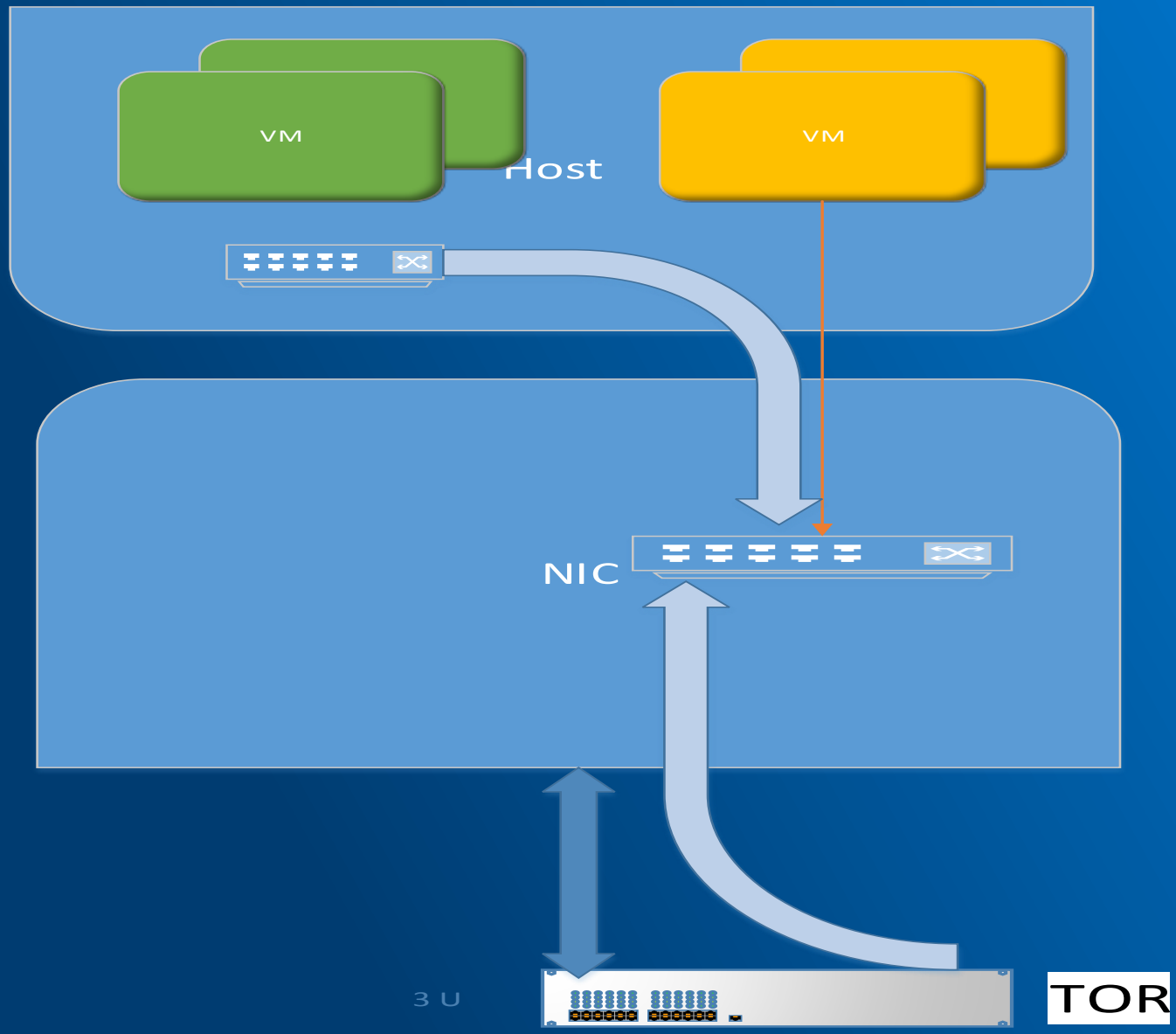
Pros:

- More Scalable than Direct Assign
- Security through IOMMU and function isolation
- Control Plane separation through PF/VF notion
- High packet rate, Low CPU, Low latency thanks to Direct Pass through

Cons:

- Rigid: Composability issues
- Control plane is pass through, puts pressure on Hardware resources
- Parts of the PCIe config space are direct map from Hardware
- Limited scalability (16 bit)
- SR-IOV NIC forces switching features into the HW
- All the Switching Features in the Hardware or nothing

Switching in the NIC



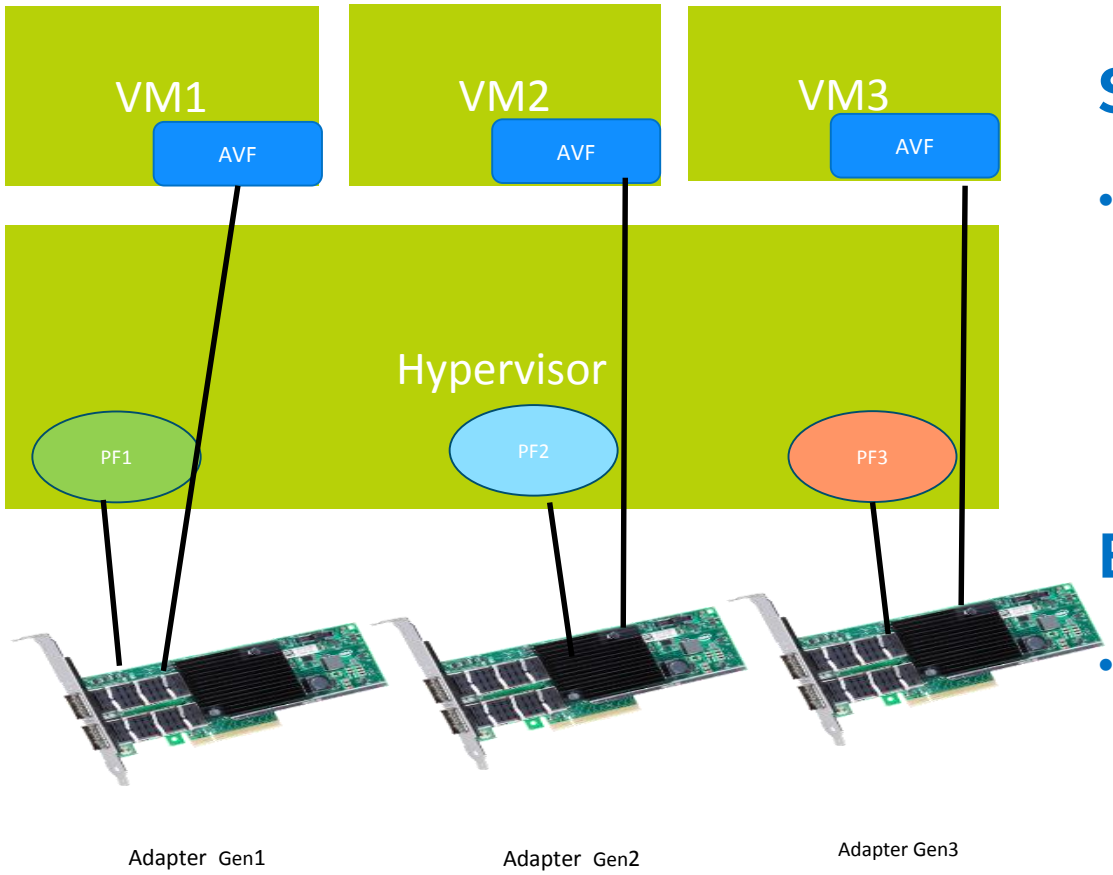
Switching Features in the Hardware

From Basic MAC filtering and mac based switching…	…to more complete Switch Features…
Anti-spoof (Basic ACL)	Mirroring
	Encryption offloads
VLAN filtering/switching/pruning	Tunnel End point Offloads
	Metering and policing
Port VLAN	Egress and Ingress forwarding rules
	HW offloaded ACLs
L3-L4 based forwarding/drop rules	HW LAG
	HW Learning
Flow counters	Multiple control Domains (Infrastructure and Tenant)
	Control Plane offloads?
Tunnel based forwarding	
Broadcast/Multicast replication	

Future or In Progress Enhancements

- Adaptive Virtual Function Driver
- Live Migration with SR-IOV
- Para Virtualized solution with HW Acceleration
- Future thoughts: (Composable VFs)

Adaptive Virtual Function



Customer Needs:

- Need a single VF driver for all generations of Devices.

Solution:

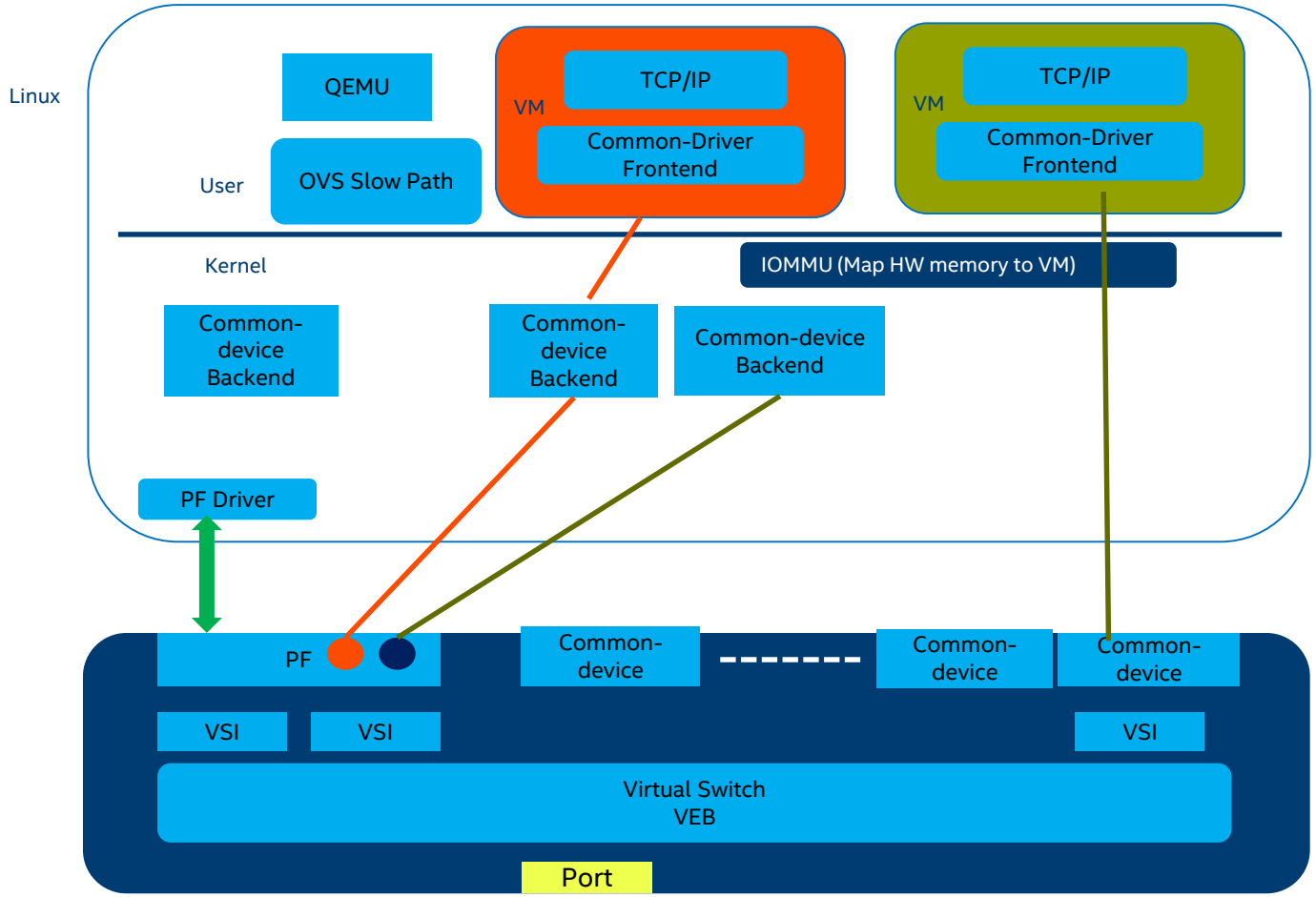
- Adaptive Virtual Function driver with Base feature set that is Forward compatible.
  - Base features
  - Negotiated advanced features

Benefits:

- Existing VM Images will run on the new hardware unchanged.

Para-virtualized with HW Acceleration

- Dedicated resources



Live Migration Issue with SR-IOV

Requirements:

- Tracking Dirty Pages
- Moving Switch state
- Hairpin (Switching redirect)
- Minimize Packet Drop rate (Implied)

Multiple Solutions:

- Hardware agnostic, Teaming Solution
- Hardware aware Solution

- Hardware agnostic Solution:
  - Integrated Teaming in software using emulated interface and pass through interface in the same driver.
  - Because of failover to emulated path, dirty pages tracked by PML.

- Hardware aware Solution:
  - IOMMU tracking dirty pages (platform level solution)
  - NIC based solution
    - Common driver for pass through and para-virtualized.
    - Shared Pages between VM and Hypervisor for the queue structures.
    - Doorbell is mapped in VM's address space in the fast path.
    - When ready to migrate, un-map the doorbells and interrupts
      - VM exit handlers are programmed appropriately.
      - Which then automatically result in VM exit and the exit handler will wake up the emulated path.

Future: Composable VFs

- Bring your own fast path, but a common management of devices and drivers in a device agnostic way.
- Get away from the PCIe Spec limitations, fabricated PCIe device for the VM.
- Too much pass through – separate the data fast path and Control path in Hardware. Control path can be in software, leaving only data fast path as pass through.

Network Performance Comparison

Other points of Comparison

- Flexibility in the Hardware implementation
- Offloads that require driver change vs. the ones that don’t
- Number of Control Domains possible (Infrastructure vs. Tenant)
- East-West traffic between VMs
- Overall Cost of maintenance

Conclusion

- SR-IOV is good but it can be improved.
- More software/hypervisor support is needed or already being worked on to reduce how much we do in the hardware for the slow path and/or control plane.
- Hypervisor should take into consideration both a device agnostic/emulated model and a device specific accelerated model for a better user experience and migratable solution.
- SwitchDev model works for an SR-IOV control plane, but still need to look into how well it will scale.
- XDP based acceleration to bypass hypervisor stack can solve some of the Network virtualization bottleneck in the emulated path.

Acknowledgement

- Dan Daly
- Liang Cunming
- John Fastabend
- Kiran Patil
- Mitch Williams
- Neerav Parikh


intel
experience
what's inside™

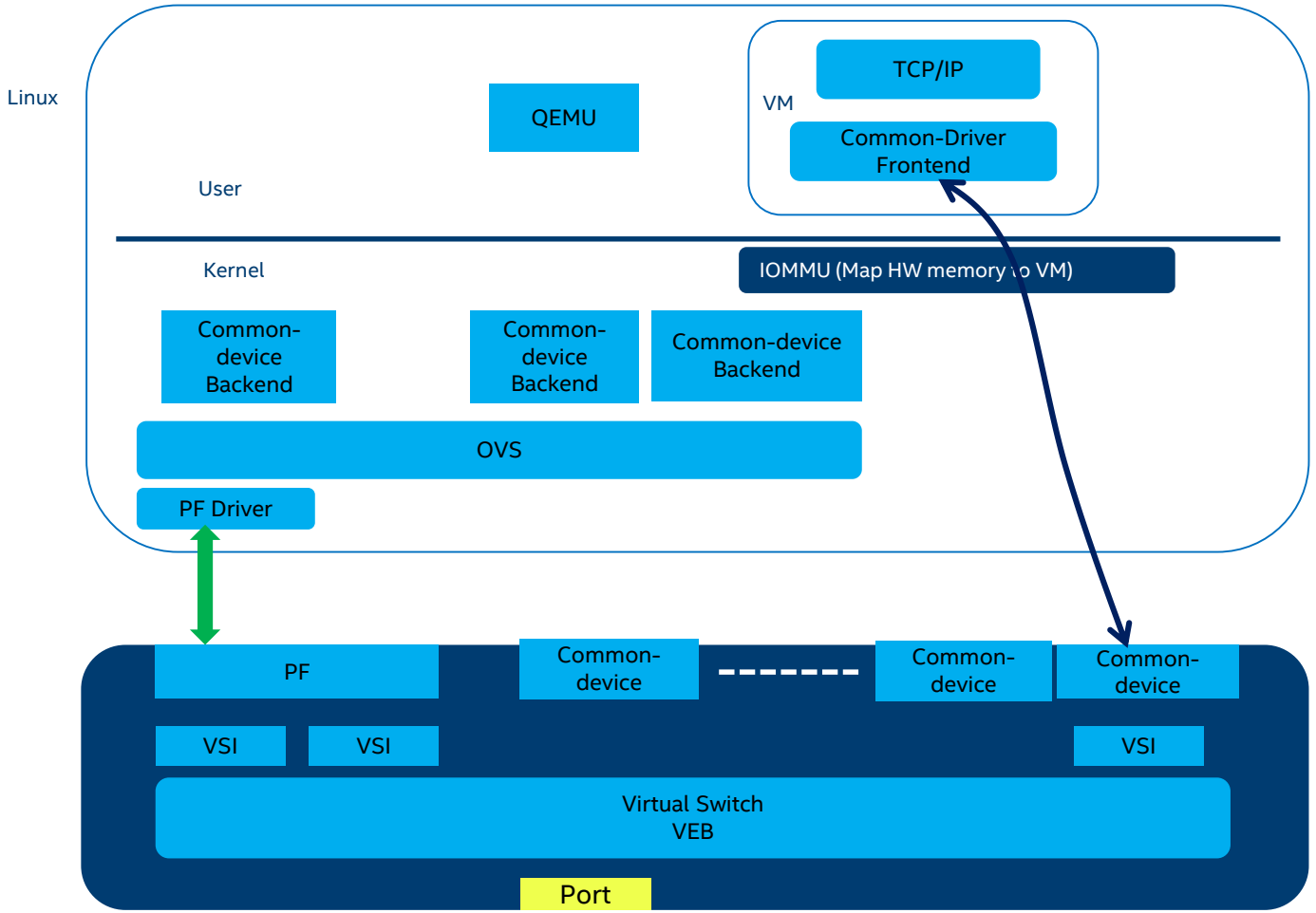
Take Away

- We at Intel are dedicated in raising the bar on Network Virtualization in terms of ease of use and feature richness for both the Cloud and Comms customers in a way that is scalable and performant.

- There is a need to rethink both the Hardware design and Software design to cater to this huge spectrum of use cases and Data explosion due to IoT.

- It requires community support and expertise.

Live Migration using Para virtualized (Fast path)



Live Migration using Para virtualized (Failover path)