OVN Offload: The Next Generation starring OpenShift and BlueField-2

Dan Winship, Red Hat

Open vSwitch Fall 2021 Conference

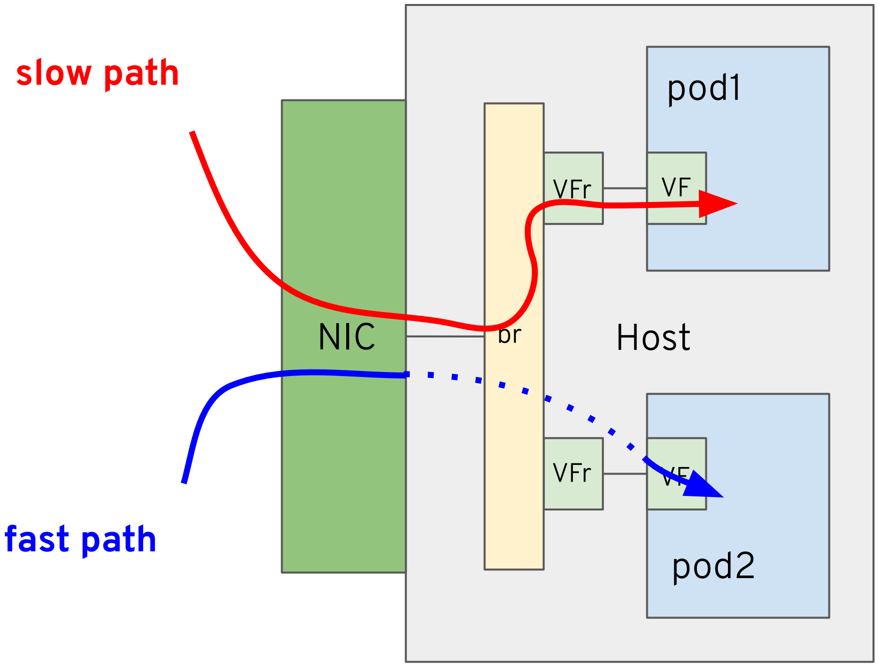
Intro

- Basic OVS offload is mostly a solved problem
- Now let’s offload MORE!
- I’ll be talking about OpenShift and the OVN-Kubernetes CNI plugin, but the general idea applies to anything using OVN (or raw OVS).
- Also, I’ll be talking about the NVIDIA BlueField-2 because that’s what we’re working with now, but some other vendors have released / will release similar NICs.

OVN-Kubernetes Offload (Currently)

- OVN-Kubernetes on each node configures OVS to do hardware offload
- Instead of veth pairs, pods are connected to the OVS bridge using SR-IOV VF/VF representor pairs. The pod gets a VF as its primary network interface, and ovn-kubernetes attaches the corresponding VF representor to the OVS bridge.
- OVS offloads matched flows to the NIC, then future matching packets can be delivered directly from the NIC to the pod’s VF without ever passing through the host network namespace.

OVN-Kubernetes Offload (Currently)



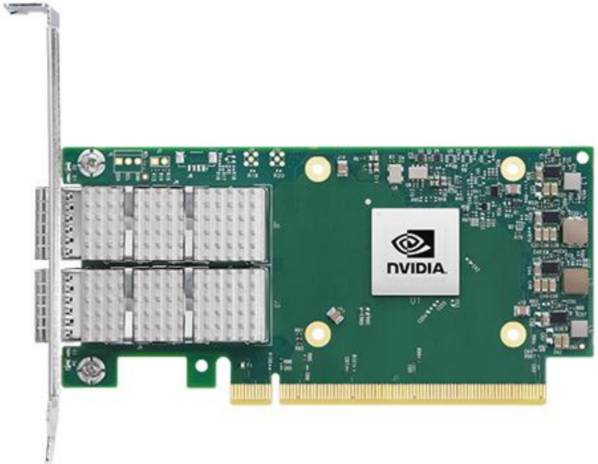
OVN-Kubernetes Offload (Currently)

- But this is really just “OVS offload”. Each node still runs `ovnkube-node`, `ovn-controller`, `ovs-vswitchd`, and `ovsdb-server`.
- These also use up RAM and CPU that could be devoted to user workloads.
- (There are other reasons for wanting OVN/ovn-kubernetes offload too, but we’ll get to that later…)

NVIDIA BlueField-2

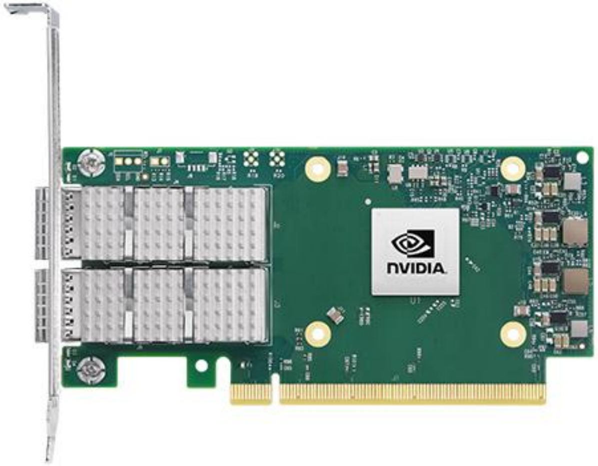
- What is a BlueField-2?

NVIDIA BlueField-2



You take a ConnectX-6...

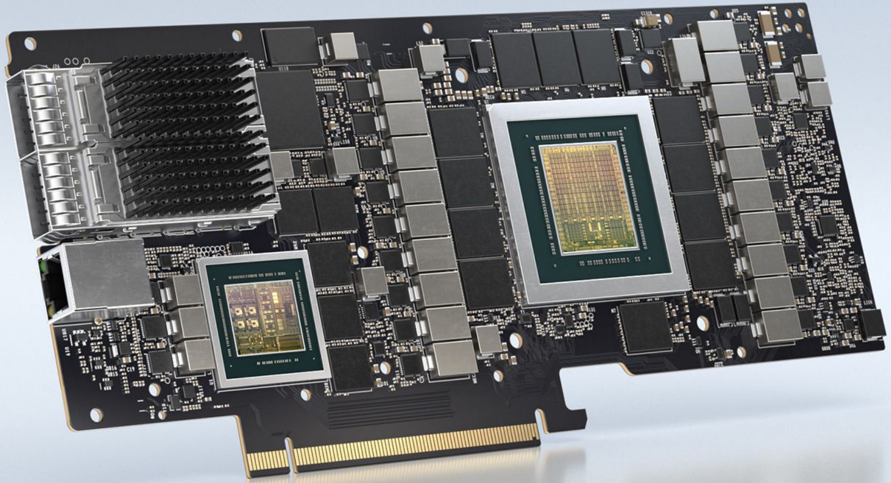
NVIDIA BlueField-2



+

... and then glue a Raspberry Pi onto it

NVIDIA BlueField-2



NVIDIA BlueField-2

- The networking capabilities of a ConnectX-6, plus an ARM system to manage the NIC separately from its host (x86) system.
- The ARM system acts as a man-in-the-middle between the host and the external network.
  - It controls what packets make it into and out of the host.
  - Can run arbitrary additional software as well.
- NVIDIA calls this a “DPU” — Data Processing Unit
  - (“Smart NIC” / “DPU” terminology is not yet consistent between vendors.)

NVIDIA BlueField-2

- The ARM SoC is a fully-generic ARM64 system
  - Up to 8 Armv8 A72 cores
  - 8/16/32GB RAM
  - 32GB flash disk, plus M.2 connector if you need real storage
  - 2 high speed network ports plus separate 1G management port
- The BlueField-specific device drivers have all been upstreamed, so it can run any Linux distribution with a new enough kernel.

OVN-Kubernetes Offload with BF-2

- Move most of the rest of the OVN and ovn-kubernetes processes off the host onto the NIC to free up resources on the host.
  - Maybe move some other stuff too? eg, CoreDNS pods
  - Don’t want to overload the NIC too much though; weaker processor, slower disk
- Take advantage of the “man-in-the-middle”-ness of the DPU for additional security and monitoring functions…
  - The host has no privileged access to the ARM system. So even if the host cluster is compromised, an attacker would not be able to bypass the monitoring and/or restrictions imposed by the ARM system.

OVN BF-2 Offload Architecture

- So how do we do this?
- We need some way to install and maintain the software on the NICs, including the base OS, ovn-kubernetes, and eventually arbitrary end-user workloads.
- Hey! We know how to do that!

OVN BF-2 Offload Architecture

- We manage the NIC ARM systems as an OpenShift cluster
  - Install RH CoreOS on each NIC using the normal OpenShift bare-metal installer
  - Open vSwitch is part of RHCOS, ovn-kubernetes is run on the NICs as a Kubernetes DaemonSet
- The NIC cluster is a separate OpenShift cluster from the cluster that the x86 hosts are part of (so that access to the NICs can be independent of access to the nodes).

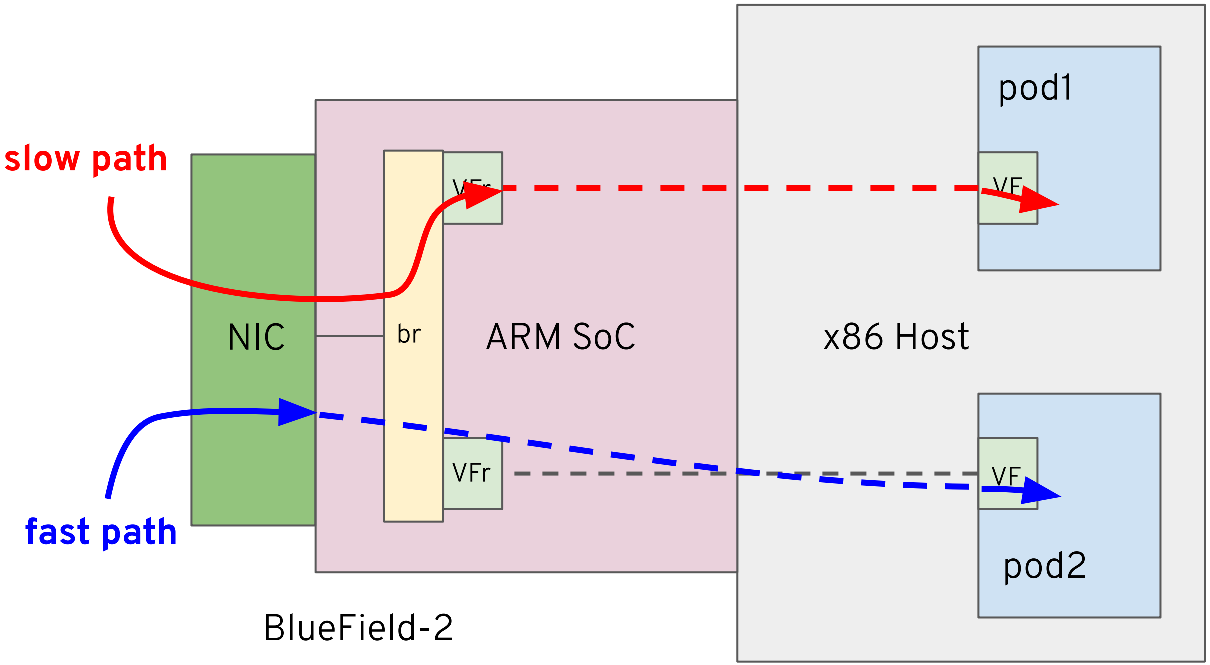
OVN BF-2 Offload Architecture

- The OVN databases and ovn-kubernetes master components will run in the host cluster, just like normal.
- The NIC cluster will run a new “dpu-network-operator” that sets things up so that the ovn-kubernetes node components in the NIC cluster can talk to the Kubernetes apiserver, ovn-kubernetes master components, and OVN databases in the host cluster.

OVN BF-2 Offload Architecture

- On the host nodes, we don’t run OVS or ovn-controller, and we run ovnkube-node in a special “dpu-host” mode, where it basically only does the CNI pod setup parts, and it uses SR-IOV VFs rather than veths.
- On the NIC nodes, we run ovnkube-node in “dpu” mode, which runs basically everything *except* the CNI pod setup parts, and instead just watches for pods being created on the host, and attaches the corresponding VF representors to its OVS bridge.
- (The ARM nodes are also configured to no “normal” OVS offload.)

OVN Offload with BlueField-2



x86 Host doesn’t
have to route the
traffic even on
the slow path.

Status in OpenShift

- “Proof of Concept” version earlier this year
  - Held together with duct tape, but it worked
- “Dev Preview” in OCP 4.10 (early 2022)
  - Running OCP on the NICs; overall architecture is mostly correct
  - But not very polished, and upgrades will be messy

Plans for “Tech Preview”

- Nicer install/update process
    - Install NICs and hosts all at once
    - Upgrade the clusters in unison so the NICs only reboot after the hosts have been drained of pods
    - Get rid of the need for separate ARM servers to run as the master nodes
- Lower resource usage in the NIC cluster (although BF-3 is bigger than BF-2 and now they’re talking about BF-4…)
- A few more features (eg IPsec offload) that aren’t yet supported by upstream kernels.

Plans for GA / Future

- User workloads in the NIC cluster
  - Monitoring
  - Setting up VPN tunnels?
  - … I dunno, it’s a Kubernetes cluster, the user can run whatever pods they want. That’s the point.
- HyperShift
  - New architecture for OpenShift to better support users with multiple clusters
  - Simplifies some of the “host cluster” vs “NIC cluster” stuff

Thank you

in linkedin.com/company/red-hat

▶ youtube.com/user/RedHatVideos

f facebook.com/redhatinc

twitter.com/RedHat

Epilogue

The Proof of Concept and most of the Developer Preview work were done by:

Fabrizio D’Angelo, Eric Garver, Peng Liu, Billy McFall, Balazs Nemeth, Zenghui Shi

(plus some NVIDIA/Mellanox people who did some of the upstream ovn-kubernetes work)