Receive side scaling (RSS) with eBPF in QEMU and virtio-net

Yan Vugenfirer - CEO, Daynix

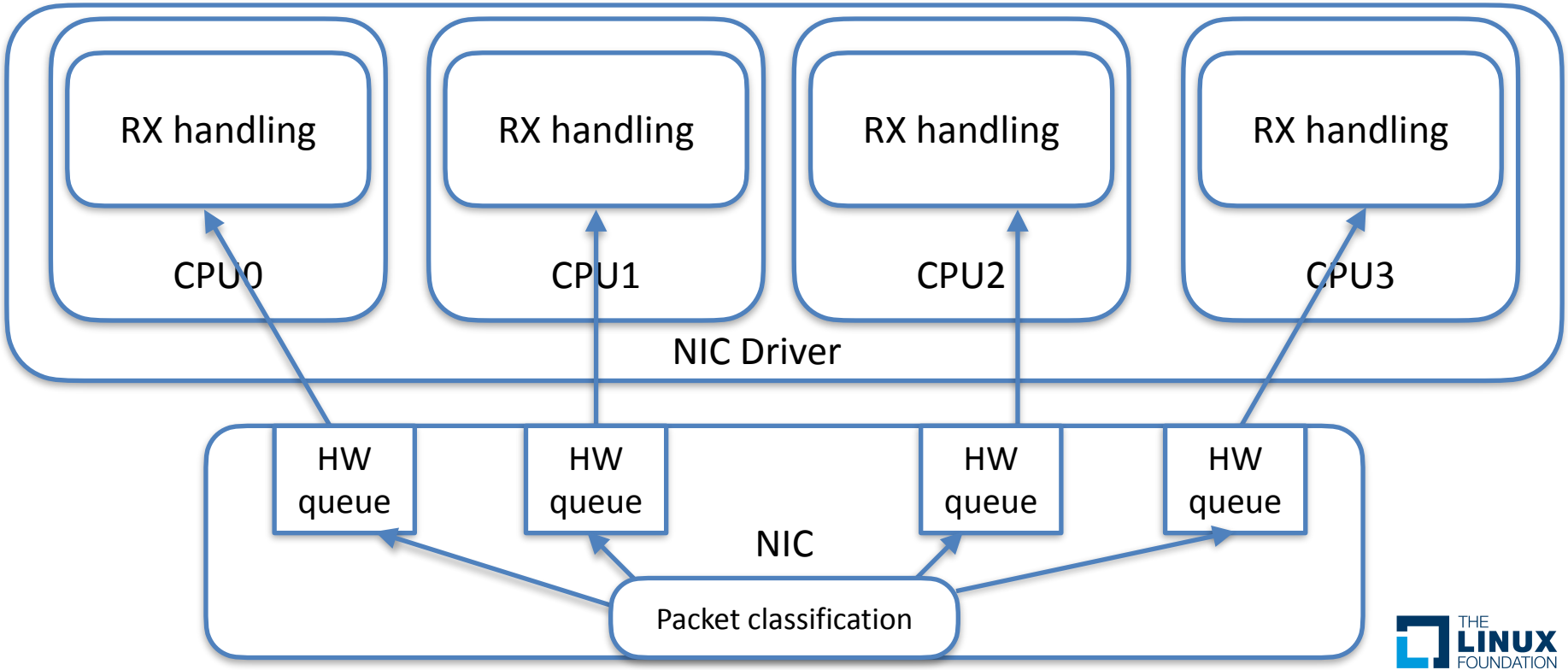
Agenda

- What is RSS?
- History: RSS and virtio-net
- What is eBPF?
- Using eBPF for packet steering (RSS) in virtio-net

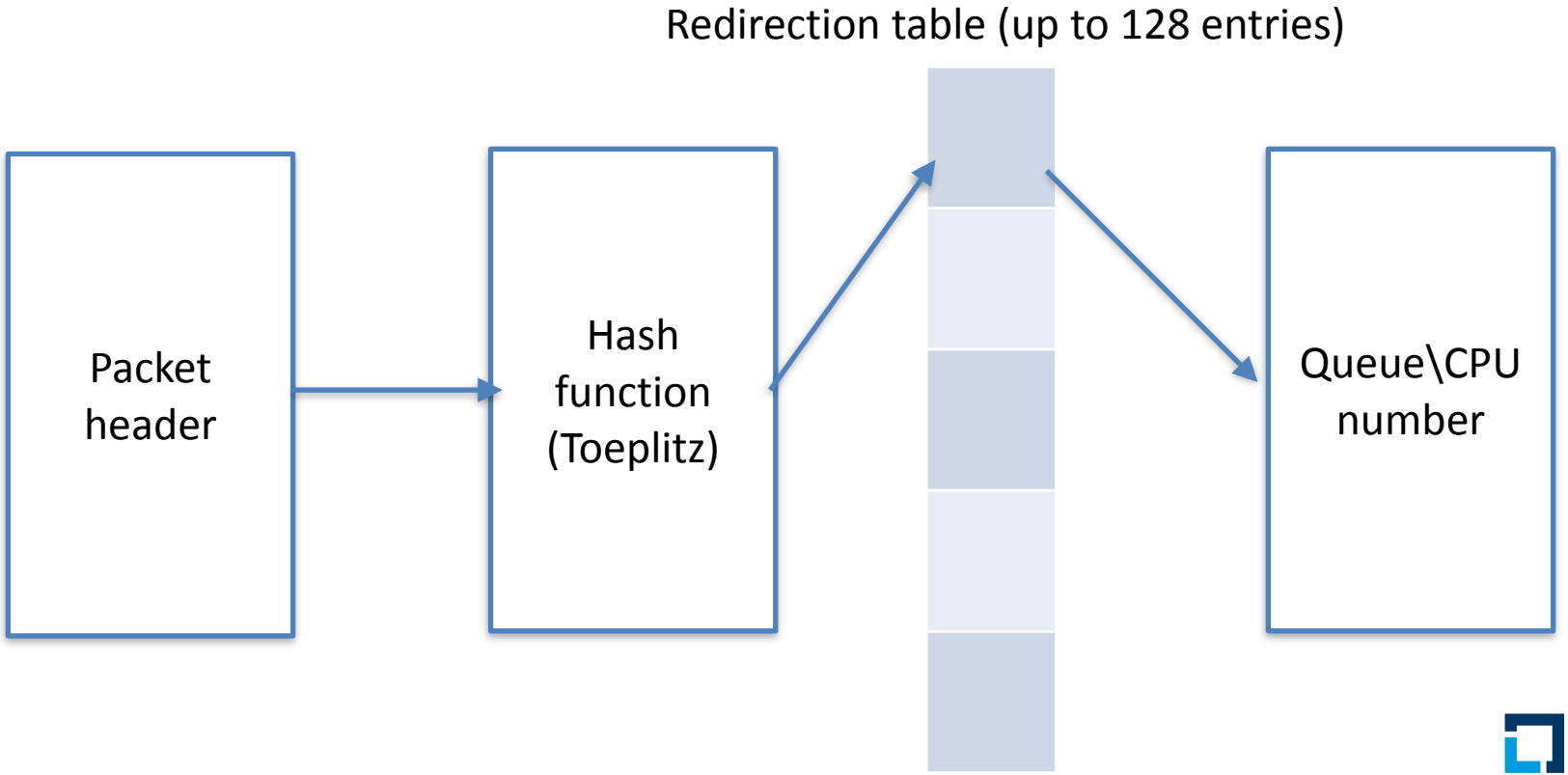
What is RSS?

- Receive side scaling - distribution of packets' processing among CPUs
  - A NIC uses a hashing function to compute a hash value over a defined area
  - Hash value is used to index an indirection table
  - The values in the indirection table are used to assign the received data to a CPU
  - With MSI support, a NIC can also interrupt the associated CPU

What is RSS?



What is RSS?

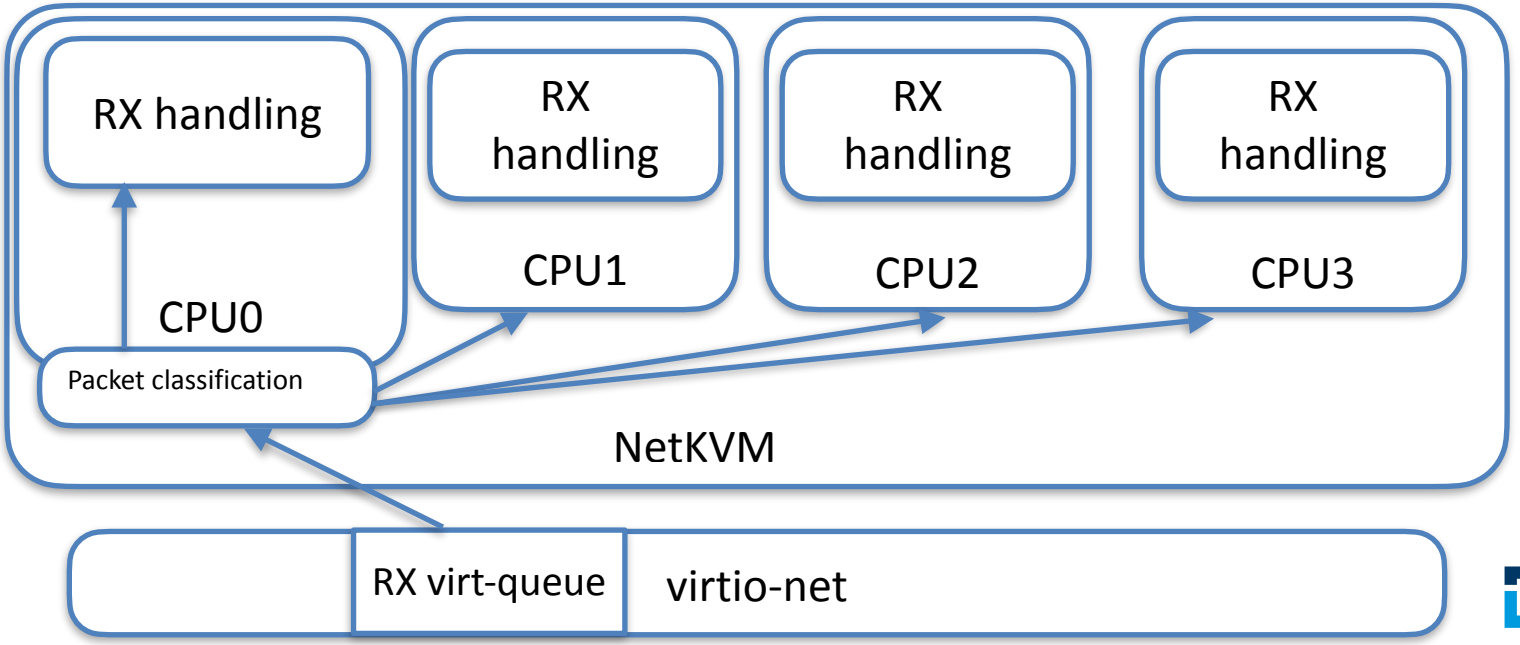


History: RSS and virtio-net

- Let’s use RSS with virtio-net!
  - Utilisation CPUs for packet processing
  - Cash locality for network applications
  - Microsoft WHQL requirement for high speed devices

History: RSS and virtio-net

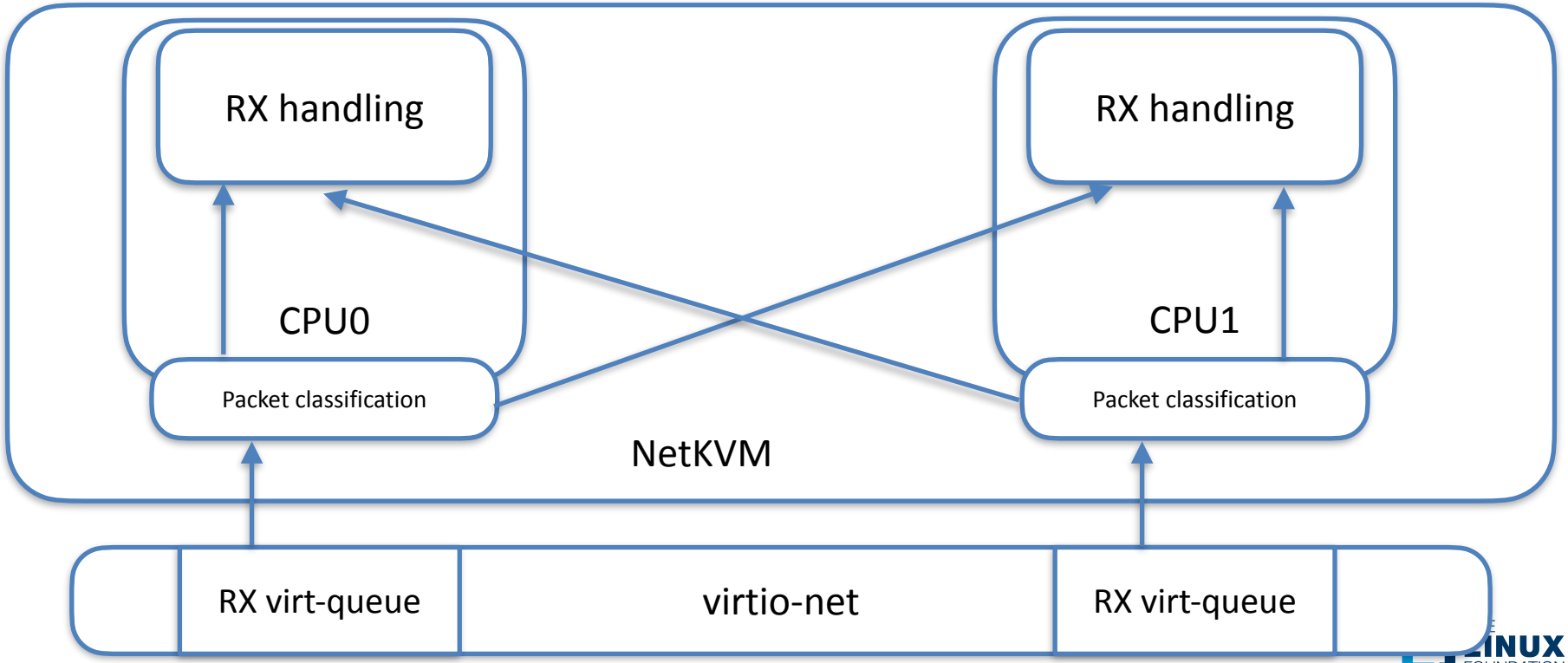
- No multi-queue in virtio
  - SW implementation in Windows guest driver (similar to RFS in Linux)



History: RSS and virtio-net

- virtio-net became multi queue device
  - Due to Windows requirements - hybrid model. Interrupt received on specific CPU core, but could be rescheduled to another
    - Works good for TCP
    - Might not work for UDP
  - Support legacy interrupts for old OSes

History: RSS and virtio-net



History: RSS and virtio-net

- virtio spec changes
  - Set steering mode
  - Pass the device redirection tables
  - Set hash value in virtio-net-hdr
  - No inter-processor interrupts due to re-scheduling
  - Vision: HW will do all the heavy work
- Implementations
  - SW only POC in QEMU
  - eBPF

virtio spec changes - capabilities

- VIRTIO_NET_F_RSS
  - VIRTIO_NET_F_MQ must be set

virtio spec changes - device configuration

- virtio_net_config

```
struct virtio_net_config {
    u8 mac[6];
    le16 status;
    le16 max_virtqueue_pairs;
    le16 mtu;
    le32 speed;
    u8 duplex;
    u8 rss_max_key_size;
    le16 rss_max_indirection_table_length;
    le32 supported_hash_types;
};
```

virtio spec changes - setting RSS parameters

- VIRTIO_NET_CTRL_MQ_RSS_CONFIG

```
struct virtio_net_rss_config {
    le32 hash_types;
    le16 indirection_table_mask;
    le16 unclassified_queue;
    le16 indirection_table[indirection_table_length];
    le16 max_tx_vq;
    u8 hash_key_length;
    u8 hash_key_data[hash_key_length];
};
```

virtio spec changes - virtio-net-hdr

```
struct virtio_net_hdr {
    u8 flags;
    u8 gso_type;
    le16 hdr_len;
    le16 gso_size;
    le16 csum_start;
    le16 csum_offset;
    le16 num_buffers;
    le32 hash_value;       (Only if
    VIRTIO_NET_F_HASH_REPORT negotiated)
    le16 hash_report;      (Only if
    VIRTIO_NET_F_HASH_REPORT negotiated)
    le16 padding_reserved; (Only if
    VIRTIO_NET_F_HASH_REPORT negotiated)
};
```

What is eBPF?

- Enable running sandboxed code in Linux kernel
- The code can be loaded at run time
- Used for security, tracing, networking, observability

How can eBPF help us?

- Calculate the RSS hash and return the queue index for incoming packets
- Populate the hash value in virtio_net_hdr (work in progress)

The “magic”

- Loading eBPF program using IOCTL TUNSETSTEERINGEBPF
- *tun_struct* has *steering_prog* field
- If eBPF program for steering is loaded, *tun_select_queue* will call it with *bpf_prog_run_clear_cb*

Hash population (work in progress)

- Population from eBPF program
- virtio_net_hdr with additional fields
- Work in progress in kernel
  - Enlarge virtio_net_hdr in all kernel modules
  - Keep calculated hash in SKB and copy it to virtio_net_hdr

eBPF program source in QEMU

- tun_rss_steering_prog
  - tools/ebpf/rss.bpf.c
- Use clang to compile
  - tools/ebpf/Makefile.ebpf

eBPF program skeleton

- During QEMU compilation include file is populated with the compiled binary
  - bpftool gen skeleton rss.bpf.o > rss.bpf.skeleton.h
- Helpers to initialise maps (mechanism to share data between eBPF program and kernel\userspace)
  - Some changes to support libvirt - mmapping the shared data structure to user space (3 maps in current main branch without mmaping, 1 map in pending patches)

Configuration map

- The configuration map is BPF array map that contains everything required for RSS:
  - Supported hash flows: IPv4, TCPv4, UDPv4, IPv6, IPv6ex, TCPv6, UDPv6
  - Indirections table size (max 128)
  - Default queue
  - Toeplitz hash key - 40 bytes
  - Indirections table - 128 entries

Loading eBPF program

- Two mechanisms
  - QEMU using function in skeleton file. Calling bpf syscall
  - eBPF helper program (with libvirt) - QEMU gets file descriptors from libvirt with already loaded ebpf program and mapping of the ebpf map (patches under review)

Loading eBPF program

- Possible load failures
  - Kernel support. Current solution requires 5.8+
  - Without helper
    - QEMU process capabilities: CAP_BPF, CAP_NET_ADMIN
    - sysctl kernel.unprivileged_bpf_disabled=1
  - libbpf not present
  - In case of helper usage - mismatch between helper and QEMU
    - Stamp is a hash of skeleton include file

Fallback

- Built it QEMU RSS steering
  - Can be triggered also by live migration
  - Hash population is enabled in QEMU command line, because there is still not hash population from eBPF program

Live migration

- Known issue: migrating to old kernel
- eBPF load failure
- Fallback to in-QEMU RSS steering

QEMU command line

- Multi-queue should be enabled
- -smp with vCPU for each queue-pair
- -device virtio-net-pci, rss=on,hash=on,ebpf_rss_fds=<fd0,fd1>

QEMU command line

- rss=on
  - Try to load eBPF from skeleton or by using provided file descriptors
  - Fallback to “built-in” RSS steering in QEMU if cannot load eBPF program
- hash=on
  - Populate hash in virtio_net_hdr
- ebpf_rss_fds - optional, provide file descriptors for eBPF program and map

libvirt integration

- QEMU should run with least possible privileges
- eBPF helper
  - Stamping the helper during compilation time
- Redirection table mapping
- Additional command line options to provide file descriptors to QEMU
- Patches under review

Current status

- Initial support was merged to QEMU
- libvirt integration patches in QEMU and libvirt are under discussion on mailing lists
- Hash population by eBPF program - pending additional work for next set of patches

Pending patches

- QEMU libvirt integration: https://lists.nongnu.org/archive/html/qemu-devel/2021-07/msg03535.html
- libvirt patches: https://listman.redhat.com/archives/libvir-list/2021-July/msg00836.html
- RSS support in Linux virtio-net driver: https://lists.linuxfoundation.org/pipermail/virtualization/2021-August/055940.html
- In kernel hash calculation reporting to guest driver: https://lkml.org/lkml/2021/1/12/1329

virtio-net and eBPF future

- Packet filtering with vhost
- Security?

yan@daynix.com

KVM FORUM

Links

- https://www.kernel.org/doc/Documentation/networking/scaling.txt
- https://docs.microsoft.com/en-us/windows-hardware/drivers/network/introduction-to-receive-side-scaling
- https://ebpf.io
- https://docs.microsoft.com/en-us/windows-hardware/drivers/network/rss-with-a-single-hardware-receive-queue
- https://docs.microsoft.com/en-us/windows-hardware/drivers/network/rss-with-hardware-queuing
- https://docs.microsoft.com/en-us/windows-hardware/drivers/network/rss-with-message-signaled-interrupts
- https://docs.microsoft.com/en-us/windows-hardware/drivers/network/rss-hashing-functions