Contribution ID: **303** Type: **not specified**

Making Networking Queues a First Class Citizen in the Kernel

Tuesday 10 September 2019 15:45 (45 minutes)

XDP (the eXpress Data Path) is a new method in Linux to process packets at L2 and L3 with really high performance. XDP has already been deployed for use cases involving ingress packet filtering, or transmission back through the ingress interface, are already well supported today. However, as we expand the use cases that involve the XDP_REDIRECT action, e.g., to send packets to other devices, or zero-copy them to userspace sockets, it becomes challenging to retain the high performance of the simpler operating modes.

One of the keys to get good performance for these advanced use cases, is effective use of dedicated hardware queues (on both Rx and Tx), as this makes it possible to split traffic over multiple CPUs, with no synchronization overhead in the fast path. The problem with using hardware queues like this is that they are a constrained resource, but are hidden from the rest of the kernel: Currently, each driver allocates queues according to its own whims, and users have little or no control over how the queues are used or configured.

In this presentation we discuss an abstraction that makes it possible to keep track of queues in a vendor-neutral way: We implement a new submodule in the Linux networking core that drivers can register their queues to. Other pieces of code can then allocate and free individual queues (or sets of them) satisfying certain properties (e.g., “a Tx/Rx pair”, or “one queue per core”). This submodule also makes sure that the queues get IDs that are hardware independent, so that they can easily be used by other components. We show how this could be exposed to userspace, and how it can interact with the existing REDIRECT primitives, such as device maps.

Finally if there is time, we would like to discuss a related problem: often a userspace program wants to express its configuration not in terms of queue IDs, but in terms of a set of packets it wants to process (e.g., by specifying an IP address). So how do we change user space APIs that use queue IDs to be able to use something more meaningful such as properties of the packet flow that a user wants? To solve this second problem, we propose to introduce a new bind option in AF_XDP that takes a simple description of the traffic that is desired (e.g. “VLAN ID 2”, “IP address fc00:dead:cafe::1”, or “all traffic on a netdev”). This hides queue IDs from userspace, but will use the new queue logic internally to allocate and configure an appropriate queue.

I agree to abide by the anti-harassment policy

Yes

I confirm that I am already registered for LPC 2019

Primary authors: KARLSSON, Magnus (Intel); TÖPEL, Björn (Intel); DANGAARD BROUER, Jesper (RedHat); HØILAND-JØRGENSEN, Toke (RedHat); KICINSKI, Jakub (Netronome); MIKITYANSKIY, Maxim (Mellanox)

Presenters: KARLSSON, Magnus (Intel); TÖPEL, Björn (Intel); DANGAARD BROUER, Jesper (RedHat); HØILAND-JØRGENSEN, Toke (RedHat); KICINSKI, Jakub (Netronome); MIKITYANSKIY, Maxim (Mellanox)

Session Classification: Networking Summit Track