Contribution ID: **414** Type: **not specified**

Per Netns RTNL

Wednesday, 18 September 2024 12:00 (30 minutes)

rtnl_lock() is the "Big Kernel Lock" used all over the networking subsystem.

It serialises various rtnetlink requests, including adding/removing/dumping networking devices, IPv4 and IPv6 addresses, routes, etc.

Since 4.14, there has been an infrastructure not to hold rtnl_lock() for some types of requests, and a lot of work has been done to convert request handlers to RTNL-free. For example, since 6.9, IPv6 addresses and IPv4 routes can be dumped under RCU instead of rtnl_lock().

While significant improvements have been made on the reader side, rtnl_lock() is still a huge pain on the writer side.

One of our services creates thousands of network namespaces and a small number of devices in each netns. Even though the rtnetlink requests are issued per netns concurrently in userspace, they are serialised in the kernel, so setting up a single host takes 10+ minutes.

This talk gives a short refresher of rtnl_lock(), introduces recent updates to lower RTNL pressure, and suggests changes, per-netns RTNL, focusing on gaining more concurrency for many netns workloads.

Primary author: IWASHIMA, Kuniyuki (Amazon Web Services)

Presenter: IWASHIMA, Kuniyuki (Amazon Web Services)

Session Classification: Networking Track

Track Classification: Networking Track