Subject: [RFD] L2 Network namespace infrastructure
Posted by ebiederm on Fri, 22 Jun 2007 19:39:24 GMT
View Forum Message <> Reply to Message

Currently all of the prerequisite work for implementing a network
namespace (i.e. virtualization of the network stack with one kernel)
has already been merged or is in the process of being merged.

Therefore it is now time for a bit of high level design review of
the network namespace work and time to begin sending patches.

-- User space semantics

If you are in a different network namespace it looks like you have a
separate independent copy of the network stack.

User visible kernel structures that will appear to be per network
namespace include network devices, routing tables, sockets, and
netfilter rules.

-- The basic design

There will be a network namespace structure that holds the global
variables for a network namespace, making those global variables
per network namespace.

One of those per network namespace global variables will be the
loopback device. Which means the network namespace a packet resides
in can be found simply by examining the network device or the socket
the packet is traversing.

Either a pointer to this global structure will be passed into
the functions that need to reference per network namespace variables
or a structure that is already passed in (such as the network device)
will be modified to contain a pointer to the network namespace
structure.

Depending upon the data structure it will either be modified to hold
a per entry network namespace pointer or it there will be a separate
copy per network namespace. For large global data structures like
the ipv4 routing cache hash table adding an additional pointer to the
entries appears the more reasonable solution.

The initialization and cleanup functions will be refactored into
functions that do the work on a network namespace basis and functions
that perform truly global initialization and cleanup. And a
registration mechanism will be available to register functions that

are per network namespace.

It is a namespace so like the other namespaces that have been implemented a clone flag will exist to create the namespace during clone or unshare.

There will be an additional network stack feature that will allow you to migrate network devices between namespaces.

When complete all of the features of the network stack ipv4, ipv6, decnet, sysctls, virtual devices, routing tables, scheduling, ipsec, netfilter, etc should be able to operate in a per network namespace fashion.

--- The implementation plan

The plan for implementing this is to first get network namespace infrastructure merged. So that pieces of the network stack can be made to operate in a per network namespace fashion.

Then the plan is to proceed as if we are doing a global kernel lock removal. For each layer of the networking stack pass down the per network namespace parameter to the functions and modify the functions to verify they are only operating on the initial network namespace. Then one piece at a time update the code to handle working in multiple network namespaces, and push the network namespace information down to the lower levels.

This plan calls for a lot of patches that are essentially noise. But the result is simple and generally obviously correct patches, that can be easily reviewed, and can be safely merged one at a time, and don't impose any additional ongoing maintenance overhead.

In my current proof of concept patchset it takes about 100 patches before ipv4 is up and working.

--- Performance

In initial measurements the only performance overhead we have been able to measure is getting the packet to the network namespace. Going through ethernet bridging or routing seems to trigger copies of the packet that slow things down. When packets go directly to the network namespace no performance penalty has yet been measured.

--- The question

At the design level does this approach sound reasonable?

Eric

p.s. I will follow up shortly with a patch that is one implementation of the basic network namespace infrastructure. Feel free to cut it to shreds (as it is likely overkill) but it should help put the pieces of what I am talking about into perspective.

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers