
Subject: Re: [PATCH] [NETNS49] support for per/namespace routing cache cleanup

Posted by [Daniel Lezcano](#) on Wed, 17 Oct 2007 15:05:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

Denis V. Lunev wrote:

> Daniel Lezcano wrote:

>> Denis V. Lunev wrote:

>>> Daniel Lezcano wrote:

>>>> Denis V. Lunev wrote:

>>>>> /proc/sys/net/route/flush should be accessible inside the net

>>>>> namespace.

>>>>> Though, the complete opening of this file will result in a DoS or

>>>>> significant entire host slowdown if a namespace process will

>>>>> continually

>>>>> flush routes.

>>>>>

>>>>> This patch introduces per/namespace route flush facility.

>>>>>

>>>>> Each namespace wanted to flush a cache copies global generation

>>>>> count to

>>>>> itself and starts the timer. The cache is dropped for a specific

>>>>> namespace

>>>>> iff the namespace counter is greater or equal global ones.

>>>>>

>>>>> So, in general, unwanted namespaces do nothing. They hold very old low

>>>>> counter and they are unaffected by the requested cleanup.

>>>>>

>>>>> Signed-of-by: Denis V. Lunev <den@openvz.org>

>>>>>

>>>> That's right and that will happen when manipulating ip addresses of

>>>> the network devices too. But I am not comfortable with your

>>>> patchset. It touches the routing flush function too hardly and it

>>>> uses current->nsproxy->net_ns.

>>>>>

>>>> IMHO we should have two flush functions. One taking a network

>>>> namespace parameter and one without the network namespace parameter.

>>>> The first one is called when a write to

>>>> /proc/sys/net/ipv4/route/flush is done (we must use the network

>>>> namespace of the writer) or when a interface address is changed or

>>>> shutdown|up. The last one is called by the timer, so we have a

>>>> global timer flushing the routing cache for all the namespaces.

>>>>>

>>> we can't :(The unfortunate thing is that the actual cleanup is

>>> called indirectly and asynchronously. The user _schedule_ the garbage

>>> collector to run NOW and we are moving over a large routing cache.

>>> Really large.

>>>>>

>>> The idea to iterate over the list of each namespace to flush is bad.
>>> We are in atomic context. The list is protected by the mutex.

Oh, by the way, I forgot something important you spotted with the list protected by the mutex.

When looking at `ipv6/fib_hash.c` with Benjamin, we need to browse the network namespaces list for the garbage collecting, but we are in an interrupt handler, so I can not use `rtnl_lock`.

Why is not possible to protect the list with a simple spinlock ? so we can call `spin_lock_bh` when we are in interrupt handler.
