Subject: Re: Network namespaces a path to mergable code.
Posted by ebiederm on Thu, 29 Jun 2006 00:25:40 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <dlezcano@fr.ibm.com> writes:

> Andrey Savochkin wrote:
>
>> Ok, fine.
>> Now I'm working on socket code.
>> We still have a question about implicit vs explicit function parameters.
>> This question becomes more important for sockets: if we want to allow to use
>> sockets belonging to namespaces other than the current one, we need to do
>> something about it.
>> One possible option to resolve this question is to show 2 relatively short
>> patches just introducing namespaces for sockets in 2 ways: with explicit
>> function parameters and using implicit current context.
>> Then people can compare them and vote.
>> Do you think it's worth the effort?
>>
>
> The attached patch can have some part interesting for you for the socket
> tagging. It is in the IPV4 isolation (part 5/6). With this and the private
> routing table you will probably have a good IPV4 isolation.
> This patch partially isolates ipv4 by adding the network namespace
> structure in the structure sock, bind bucket and skbuf.

Ugh. skbuf sounds very wrong. Per packet overhead?

> When a socket
> is created, the pointer to the network namespace is stored in the
> struct sock and the socket belongs to the namespace by this way. That
> allows to identify sockets related to a namespace for lookup and
> procfs.
>
> The lookup is extended with a network namespace pointer, in
> order to identify listen points binded to the same port. That allows
> to have several applications binded to INADDR_ANY:port in different
> network namespace without conflicting. The bind is checked against
> port and network namespace.

Yes. If we don't duplicate the hash table we need to extend the lookup.

> When an outgoing packet has the loopback destination addres, the
> skbuff is filled with the network namespace. So the loopback packets
> never go outside the namespace. This approach facilitate the migration
> of loopback because identification is done by network namespace and
> not by address. The loopback has been benchmarked by tbench and the

> overhead is roughly 1.5 %

Ugh. 1.5% is noticeable.

I think it is cheaper to have one loopback device per namespace.
Which removes the need for a skbuff tag.

Eric