Subject: Re: Which of the virtualization approaches is more suitable for kernel?
Posted by ebiederm on Mon, 27 Feb 2006 21:14:20 GMT
View Forum Message <> Reply to Message

Dave Hansen <haveblue@us.ibm.com> writes:

> On Fri, 2006-02-24 at 14:44 -0700, Eric W. Biederman wrote:
>> We can start on a broad front, looking at several different things.
>> But I suggest the first thing we all look at is SYSVIPC. It is
>> currently a clearly recognized namespace in the kernel so the scope is
>> well defined. SYSVIPC is just complicated enough to have a
>> non-trivial implementation while at the same time being simple enough
>> that we can go through the code in exhausting detail. Getting the
>> group dynamics working properly.
>
> Here's a quick stab at the ipc/msg.c portion of this work. The basic
> approach was to move msg_ids, msg_bytes, and msg_hdrs into a structure,
> put a pointer to that structure in the task_struct and then dynamically
> allocate it.
>
> There is still only one system-wide one of these for now. It can
> obviously be extended, though. :)
>
> This is a very simple, brute-force, hack-until-it-compiles-and-boots
> approach. (I just realized that I didn't check the return of the alloc
> properly.)
>
> Is this the form that we'd like these patches to take? Any comments
> about the naming? Do we want to keep the _namespace nomenclature, or
> does the "context" that I used here make more sense

I think from 10,000 feet the form is about right.

I like the namespace nomenclature. (It can be shorted to _space or _ns).
In part because it shortens well, and in part because it emphasizes that
we are *just* dealing with the names.

You split the resolution at just ipc_msgs. When I really think it should
be everything ipcs deals with.

Performing the assignment inside the tasklist_lock is not something we
want to do in do_fork().

So it looks like a good start. There are a lot of details yet to be filled
in, proc, sysctl, cleanup on namespace release. (We can still provide
the create destroy methods even if we don't hook the up).

I think in this case I would put the actual namespace structure

definition in util.h, and just put a struct ipc_ns in sched.h.
sysvipc is isolated enough that nothing outside of the ipc/
directory needs to know the implementation details.

It probably makes sense to have a statically structure and
to set the pointer initially in init_task.h

Until we reach the point where we can multiple instances that
even removes the need to have a pointer copy in do_fork()
as that happens already as part of the structure copy.

Eric
