

Re: RFC: ipath ioctls and their replacements

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-01/msg07593.html>

- *From:* ebiederm@xxxxxxxxxxxxx (Eric W. Biederman)
 - *Date:* Thu, 19 Jan 2006 11:20:48 -0700
-

"Bryan O'Sullivan" <bos@xxxxxxxxxxxxx> writes:

> On Thu, 2006-01-19 at 01:25 -0700, Eric W. Biederman wrote:
>
>> Do the infiniband verbs not allow dealing with a unreliable datagram
>> protocol?
>
> Eric, I think you are misunderstanding what we are actually trying to
> do. We already implement IB verbs and the various IB networking
> protocols in our drivers, at a layer that is not at all related to the
> one that is currently festooned with ioctls.
>
> The ioctl discussion pertains to lower-level direct user access to the
> hardware, for a protocol that bypasses the entire IB stack and just
> happens to send UD-compliant datagrams over the wire.

I'm surprised. I didn't think your native datagrams were complaint above the link level with any of the IB protocols in the kernel.

In any case that is not what I am saying. I am saying that I think that if the IB/rdma/networking layer does not do a good job of supporting you it is a failure there. Your driver looks ugly because there is not a sufficiently good helper layer. For high performance non-IP targeted networking cards you aren't doing anything terribly exotic. Could you please detail why you can't use the IB/rdma whatever helper layer, is insufficient to do what you need.

If it is byzantine and heavy weight that concern needs to be addressed. I agree the normal software stack is pretty tall.

> I'm actually pretty satisfied with the feedback I've already gotten from
> Greg K-H and davem.
>
>> We need some generic subsystem support to do this.
>
> I am more than happy to put together generic support, provided I see
> other drivers that could take advantage of it being considered for
> submission. Right now, I do not - in general - see this happening.

Re: RFC: ipath ioctls and their replacements

Right now it largely seems to be a chicken and the egg problem. There is a large portion of the HPC community that doesn't believe they are interesting to the rest of the world or that the rest of the world is interesting to them so they do their own thing leading to support problems.

There are other drivers for linux right now, that the vendors are not too concerned about closed source that potentially code. I can think of at least 3 other networking fabrics out there. Heck the kernel already has a myrinet driver in it. Currently it only supports

I also know there is another infiniband adapter that only provides raw packet access like yours does.

I'm sick and tired of drivers having to invent all of the user space glue elements, for HPC.

> I know that some other drivers need to do user page pinning, and I'm
> happy to try to find a generic solution that is common to IB and drivers
> unrelated to IB.

Which is the RDMA thing. And looking at the code and I don't see how

>>> RCVCTRL enables/disables receipt of packets.
>>
>> Again this is a generic problem, and the generic interfaces are broken
>> if you can't do this.
>
> The SIOCSIFFLAGS ioctl, which I assume is the generic interface you
> refer to (it's the one used by iproute, at any rate), has poor overlap
> with what we need (it supports a pile of stuff that we don't care about,
> and we require a pile of stuff it doesn't support), and I don't feel
> inclined to try using it in any case.

But SIOCSIFFLAGS is not implemented by a driver. It is implemented by the networking subsystem. It requires a network device to make sense in any case.

>>> SET_PKEY sets a partition key, essentially telling hardware
>>> which packets are interesting to userspace.
>>
>> I'm pretty certain this should be something that should be set
>> at open time.
>
> It might be possible to make it fit into whatever replaces USERINIT, or
> else we can use a netlink message of its own.
>
>>> UPDM_TID and FREE_TID are used for RDMA context management.
>>>
>>> WAIT waits for incoming packets, and can clearly be replaced by
>>> file_ops->poll.

Re: RFC: ipath ioctls and their replacements

>>>
>>> GETCOUNTERS, GETUNITCOUNTERS and GETSTATS can all be replaced by
>>> files in sysfs.
>>
>> This whole section just cries out for a network/rdma/ib/kernel-by-pass
>> layer that is that any interesting network driver can use.
>
> No, it doesn't. Our chip's approach to remote memory access doesn't
> even slightly resemble that of other comparable chips. In addition, our
> counters are entirely device-specific, and I'm already planning to move
> them to sysfs. The sysfs move gets them out of ioctl-land, and there's
> no point in trying to do anything beyond that.

Agreed, counters and sysfs are a good match. But the generic networking layer already has support for counters that are different for every device. That helper really needs to export those counters to sysfs as well as ethtool but the support already exists for more typical networking.

The problem actually gets pretty simple when you need to design an interface to support generic kernel-by-pass over using arbitrary protocols. There are so few things in common those things that are in common stick out.

>> Infiniband stack, it's there use it.
>
> No. If you're running a full IB stack, we provide the usual IB subnet
> management facilities, and you can run OpenSM to manage your subnet. If
> you're *not*, which is the case I'm concerned with here, it makes no
> sense to replicate the byzantine IB management interfaces in order to do
> a handful of simple things that aren't even tied to the higher-level IB
> protocols.

Is it the stack that is byzantine? Or the interface too it.
What I thinking ultimately is there should be something about as simple as af_packet in the kernel (but at the IB/rdma) layer that gives you the help you need.

>> There are a couple of choices here.
>
> Yes, we'll use the firmware interface, as Greg suggested.

I will have to look. That one doesn't sound familiar... Do we really have 4 wheels in the kernel?

>> There is also an interface in /proc or /sys I forget which
>> that let's you select the individual bar for a pci device.
>
> Yes, we'll use that.
>
> Thanks for your comments.

Re: RFC: ipath ioctls and their replacements

Welcome, and thanks for your patience with this process.

Eric

–

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>

- **Follow-Ups:**

- ◆ **[Re: RFC: ipath ioctls and their replacements](#)**

- ◇ From: Bryan O'Sullivan

- ◆ **[Re: \[openib-general\] Re: RFC: ipath ioctls and their replacements](#)**

- ◇ From: Sean Hefty

- **References:**

- ◆ **[RFC: ipath ioctls and their replacements](#)**

- ◇ From: Bryan O'Sullivan

- ◆ **[Re: RFC: ipath ioctls and their replacements](#)**

- ◇ From: Eric W. Biederman

- ◆ **[Re: RFC: ipath ioctls and their replacements](#)**

- ◇ From: Bryan O'Sullivan

- Prev by Date: **[Re: scsi cmd slab leak? \(Was Re: \[ck\] Anyone been having OOM killer problems lately?\)](#)**

- Next by Date: **[Re: RFC: OSS driver removal, a slightly different approach](#)**

- Previous by thread: **[Re: RFC: ipath ioctls and their replacements](#)**

- Next by thread: **[Re: \[openib-general\] Re: RFC: ipath ioctls and their replacements](#)**

- Index(es):

- ◆ **[Date](#)**

- ◆ **[Thread](#)**