

Re: RFC: ipath ioctls and their replacements

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-01/msg07453.html>

- *From:* ebiederm@xxxxxxxxxxxxx (Eric W. Biederman)
 - *Date:* Thu, 19 Jan 2006 01:25:39 -0700
-

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

- > When I posted the last round of ipath driver code for review, people
- > objected to the number of ioctls we had. I'd like to get feedback on
- > what would be acceptable replacements.

Roland you know the RDMA model best, are things so tied to the current crop of infiniband protocols that what the ipath code wants to do is not covered?

They clearly need subsystem support and what they are trying to do either isn't covered or they don't see how to use what is there. Do the infiniband verbs not allow dealing with a unreliable datagram protocol?

- > We have four kinds of ioctl right now:
- >
- > * Interfacing with userspace
- > * Infiniband subnet management
- > * Flash/EEPROM management
- > * Diagnostics
- >
- > There are currently 36 ioctls in total. I think that I can reduce this
- > number dramatically, but we're having some contentious internal debate
- > about whether and how some of the ioctls should be replaced. I'd like
- > to see what's most likely to get accepted. Obviously, we'd prefer the
- > number to be zero, but I don't think we can do that without submitting a
- > driver that isn't very useful.
- >
- > Unless I indicate otherwise, I cannot think of clean replacements for
- > the ioctls listed below, and would appreciate suggestions.
- >
- > For user access:
- >
- > Opening the /dev/ipath special file assigns an appropriate free
- > unit (chip) and port (context on a chip) to a user process.
- > Think of it as similar to /dev/ptmx for ttys, except there isn't
- > a devpts-like filesystem behind it. Once a process has
- > opened /dev/ipath, it needs to find out which unit and port it

Re: RFC: ipath ioctls and their replacements

- > has opened, so that it can access other attributes in /sys. To
- > do this, we provide a GETPORT ioctl.

We need some generic subsystem support to do this. If the kernel ib/rdma support is not enough to do this we need to build something. Dealing with NUMA affinity should not be something drivers need to invent.

- > USERINIT and BASEINFO work with mmap to set up direct access to
- > the hardware for user processes. We intend to turn these into a
- > single ioctl, USERINIT. This copies a substantial amount of
- > information to and from userspace.

I'm not certain but the concept sounds generic even if the information is not. This sounds like a job for the ib/rdma/kernel-bypass networking subsystem.

- > RCVCTRL enables/disables receipt of packets.

Again this is a generic problem, and the generic interfaces are broken if you can't do this. I know the linux network stack already provides this.

- > 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.

- > 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.
- >
- > 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. A device driver should not need to invent the interfaces for this kind of functionality.

- > For subnet management:
- >
- > GETLID, SET_LID, SET_MTU, SET_GUID, SET_MLID, GET_MLID,
- > GET_DEVSTATUS, GET_PORTINFO and GET_NODEINFO can all be replaced
- > by files in sysfs.
- >
- > SET_LINKSTATE changes the link state.
- >
- > SEND_SMA_PKT and RCV_SMA_PKT send and receive subnet management

Re: RFC: ipath ioctls and their replacements

- > packets. I *think* they could be replaced by read and write
- > methods on a new special file, although the semantics aren't a
- > super-clean match.

Infiniband stack, it's there use it.

If the Infiniband stack is too ugly to use or it is missing features then we need to fix it. So please complain about why you are have a hard time using the in-kernel infiniband stack, for this.

- > For EEPROM/flash management:
- >
- > READ_EEPROM reads the flash. WRITE_EEPROM writes it. I don't
- > see a standard way of doing this in the kernel; many drivers
- > provide their own private ioctls, some on dedicated special
- > files. I think that using read and write instead would be okay
- > (with a small qualm about semantics), but this idea makes an
- > influential coworker barf violently. I can't see how we could
- > use the ethtool flash interface: the low-level driver doesn't
- > look like a regular net device, and we support partial updates
- > of the flash.

There are a couple of choices here. Off the top of my head. Have your driver support an i2c device, have your driver export an mtd device, and ethtool are the most standard. Partly it depends on what you are trying to do.

Partial updates are not a problem. Just keep a cached copy and only write to those bytes that have changed.

- > For diagnostics:
- >
- > DIAGENTER and DIAGLEAVE put the driver into and out of diag
- > mode. These could be replaced by open/close of a special file.

This one does sound global to a device and a trivial parameter. sysfs does sound like the proper interface here. That makes it script controllable etc.

- > DIAGREAD and DIAGWRITE perform direct accesses to the device's
- > PCI memory space. I think these could be replaced by read and
- > write, but they are again subject to the make-coworker-barf
- > problem.

mmap(/dev/mem)

There is also an interface in /proc or /sys I forget which that let's you select the individual bar for a pci device. You don't need to do anything, in your driver to support this.

- > HTREAD and HTWRITE perform direct accesses to the device's PCI
- > config space. Same disagreement problem as DIAGREAD and
- > DIAGWRITE.

Re: RFC: ipath ioctls and their replacements

Again. This is generic functionality already provided by the kernel, no need to implement anything. lspci/setpci already handle this quite well.

- > SEND_DIAG_PKT can be replaced with whatever sends and receives
- > subnet management packets, as above.
- >
- > DIAG_RD_I2C is synonymous with READ_EEPROM, and will go away.
- >
- > Depending on how you look at it, we can slim our list of ioctls down to
- > somewhere between 6 and 10. This isn't zero, but it's not 36, either.
- > What do people think?

It's getting there. :)

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: Roland Dreier
- ◆ **Re: RFC: ipath ioctls and their replacements**
◇ From: Bryan O'Sullivan
- ◆ **Re: RFC: ipath ioctls and their replacements**
◇ From: David S. Miller

• *References:*

- ◆ **RFC: ipath ioctls and their replacements**
◇ From: Bryan O'Sullivan

- Prev by Date: **Re: - add-pselect-ppoll-system-call-implementation-tidy.patch removed from -mm tree**
- Next by Date: **[PATCH] 2.6.16-rc1-mm1 - produce useful info for kzalloc with DEBUG SLAB**
- Previous by thread: **Re: [openib-general] Re: RFC: ipath ioctls and their replacements**
- Next by thread: **Re: RFC: ipath ioctls and their replacements**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**