

User Space DMA

- Neil Horman
 - Red Hat

Whats been going on

- DPDK is a thing

- Provides High speed networking

- As vendor neutral as a user space SDK gets

- Basically a user space stack implementation

- Including the driver

Soooo....

- Drivers in user space is never good

- Security issues

- Hardware arbitration issues

- We all know this drill

Mitigating the risk

•John Fastabend has been working on a method to mitigate this

–Called the Bifurcated Driver

–Basically Just DMA straight to user space

–Really nice for several reasons

There are still problems

- Someone in the community is all up in arms about making sure process memory doesn't get trampled.
- That's being worked on, hopefully soon some IOMMU goodness will come to the rescue here
- DMA to user space still exposes hardware details that we don't want to deal with (namely DMA descriptor format)

What I would like to fix

- I think the next logical step here is to provide a method to make user space DMA descriptors generic
- Without sacrificing performance
- Read: No translation on syscall

Ideas

- .Use a VDSO area

- .Pros

- Existing infrastructure

- Well understood

- .Cons

- Needs to be a per net device solution

Ideas

- Dynamically created DSO file
- Registered by driver, dlopen-ed by application
- Pros
 - Nice per device solution
- Cons
 - Likely requires some significant build tomfoolery

Ideas

- Exported Pseudo code
- Maps fields from generic DMA descriptor to HW specific one
- Adds instructions to set non-standard fields
- Sort of Reverse-BPF
- Requires User space library to interpret
- Nice vendor agnostic solution

Thoughts

- Best approach?
- Things I'm missing?
- Comments regarding my charm and good looks?