Contribution ID: **71**

Type: **not specified**

Building a fast nvme passthrough

Thursday, 23 September 2021 10:00 (45 minutes)

New storage features, especially in NVMe, are emerging fast. It
takes time and a good deal of consensus-building for a device-feature
to move up the ladders of kernel I/O stack and show-up to user-space.
This presents challenges for early technology adopters.

The passthrough interface allows such features to be usable (at least
in native way) without having to build block-generic commands,
in-kernel users, emulations and file-generic user-interfaces. That said,
even though passthrough interface cuts through layers of
abstraction and reaches to NVMe fast, it has remained tied to
synchronous ioctl interface, making it virtually useless for fast I/O path.

In this talk I will present the elements towards building a scalable
passthrough that can be readily used to play with new NVMe features.
More specifically, recent upstream efforts involving:

- Emergence of per-namespace char interface, that remains available/usable even for unsupported features and new command-sets[1]
- Async-ioctl facility 'uring_cmd' that Jens proposed in io_uring [2].
- Async nvme-passthrough that I put up over 'uring_cmd' [3]

Performance evaluation comparing this new interface with existing ones
will be provided.
I would like to gather the feedback on the design-decisions, and discuss
how best to go about infusing more perf-centric advancements (e.g.
async polling, register-buffer etc.) into this path.

[1] https://lore.kernel.org/linux-nvme/20210421074504.57750-1-minwoo.im.dev@gmail.com/
[2] https://lore.kernel.org/linux-nvme/20210317221027.366780-1-axboe@kernel.dk/
[3] https://lore.kernel.org/linux-nvme/20210325170540.59619-1-joshi.k@samsung.com/

I agree to abide by the anti-harassment policy

I agree

Primary author: JOSHI, kanchan

Presenter: JOSHI, kanchan

Session Classification: Kernel Summit

Track Classification: Kernel Summit