



Contribution ID: 304

Type: **not specified**

FW centric devices, NIC customization

Tuesday, 13 September 2022 16:00 (30 minutes)

For a long time now the industry has been building programmable processors into devices to run firmware code. This is a long standing design approach going back decades at this point. In some devices the firmware is effectively a fixed function and has little in the way of RAS features or configurability. However, a growing trend is to push significant complexity into these devices processors.

Storage has been doing FW centric devices for a long time now, and we can see some evolution there where standards based channels exist that carry device specific data. For instance, looking at nvme-cli we can see a range of generic channels carrying device specific RAS or configuration (smart-log, fw-log, error-log, fw-download). nvme-cli also supports entire device specific extensions to access unique functionality (nvme-intel-nvme-huawei-, nvme-micro-*)

<https://man.archlinux.org/man/community/nvme-cli/nvme.1.en>

This reflects the reality that standardization can only go so far. The large amount of FW code still needs RAS and configuration unique to each device's design to expose its full capability.

In the NIC world we have been seeing FW centric devices for a long time, starting with MIPS cores in early Broadcom devices, entire Linux OS's in early "offload NICs", to today's highly complex NIC focusing on complicated virtualization scenarios.

For a long time our goal with devlink has been to see a similar healthy mix of standards based multi-vendor APIs side by side with device specific APIs, similar to how nvme-clie is handling things on the storage side.

In this talk, we will explore options, upstream APIs and mainstream utilities to enjoy FW-centric NIC customizations.

We are focused on:

- 1) non-volatile device configuration and firmware update - static and preserved across reboots
- 2) Volatile device global firmware configuration - runtime.
- 3) Volatile per-function firmware configuration (PF/VF/SF) - runtime.
- 4) RAS features for FW - capture crash/fault data, read back logs, trigger device diagnostic modes, report device diagnostic data, device attestation

I agree to abide by the anti-harassment policy

Yes

Primary authors: GUNTHORPE , Jason (Nvidia); MAHAMEED, Saeed (Nvidia); BLOCH, Mark (Nvidia)

Presenters: MAHAMEED, Saeed (Nvidia); BLOCH, Mark (Nvidia)

Session Classification: eBPF & Networking

Track Classification: eBPF & Networking Track