

Virtualized disaggregated block storage in IPDK using a containerized simulation environment

Artsiom Koltun

artsiom.koltun@intel.com

Filip Szufnarowski

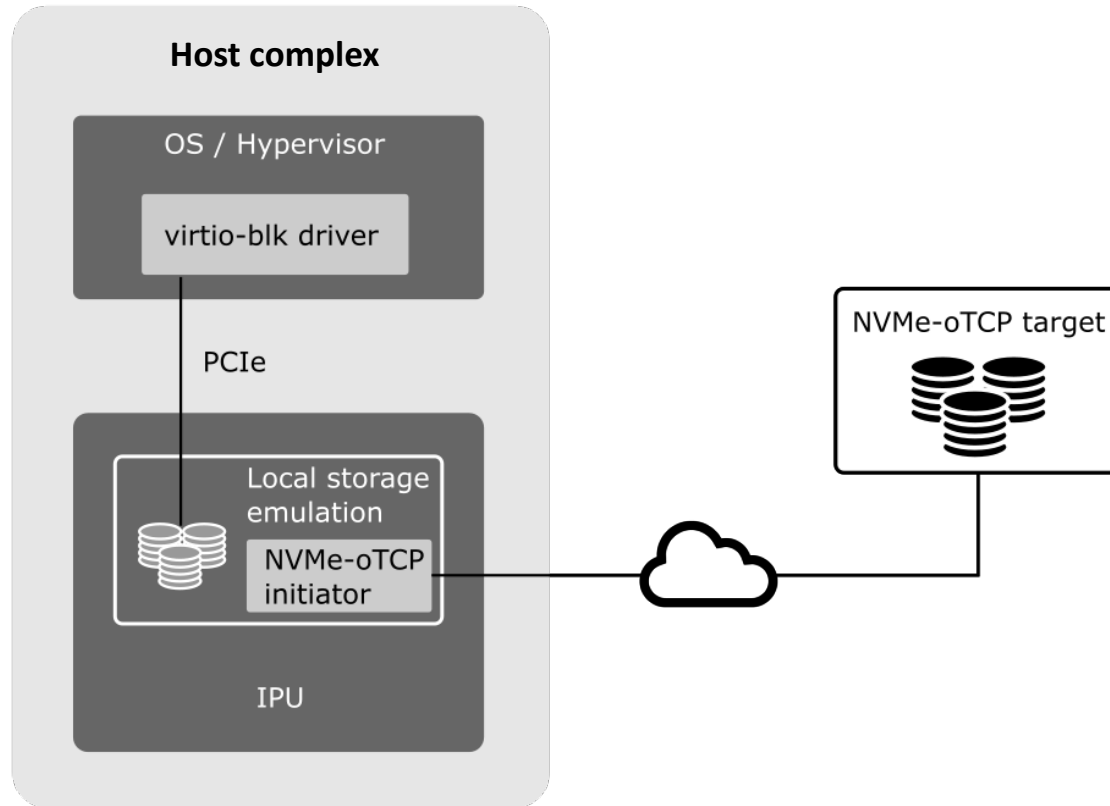
filip.szufnarowski@intel.com

June 12, 2022

Agenda

- | | |
|----------------|--------------|
| 1. THEORY | / BACKGROUND |
| 2. PRACTISE | / LIVE DEMO |
| 3. NEAR FUTURE | / NEXT STEPS |

THEORY — IPU STORAGE EMULATION

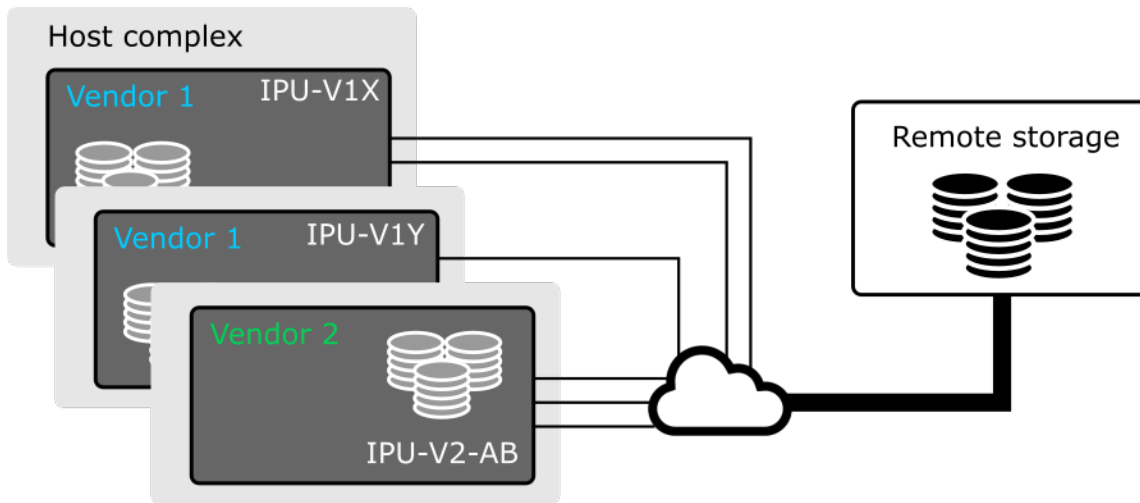


Problems to solve:

- provide abstraction over HW-storage resources
- enable storage disaggregation from local system
- remain transparent to any client-side SW incl. drivers

Today our focus is on host-facing **virtio-blk** interface and remote storage connection via **NVMe-oTCP** protocol.

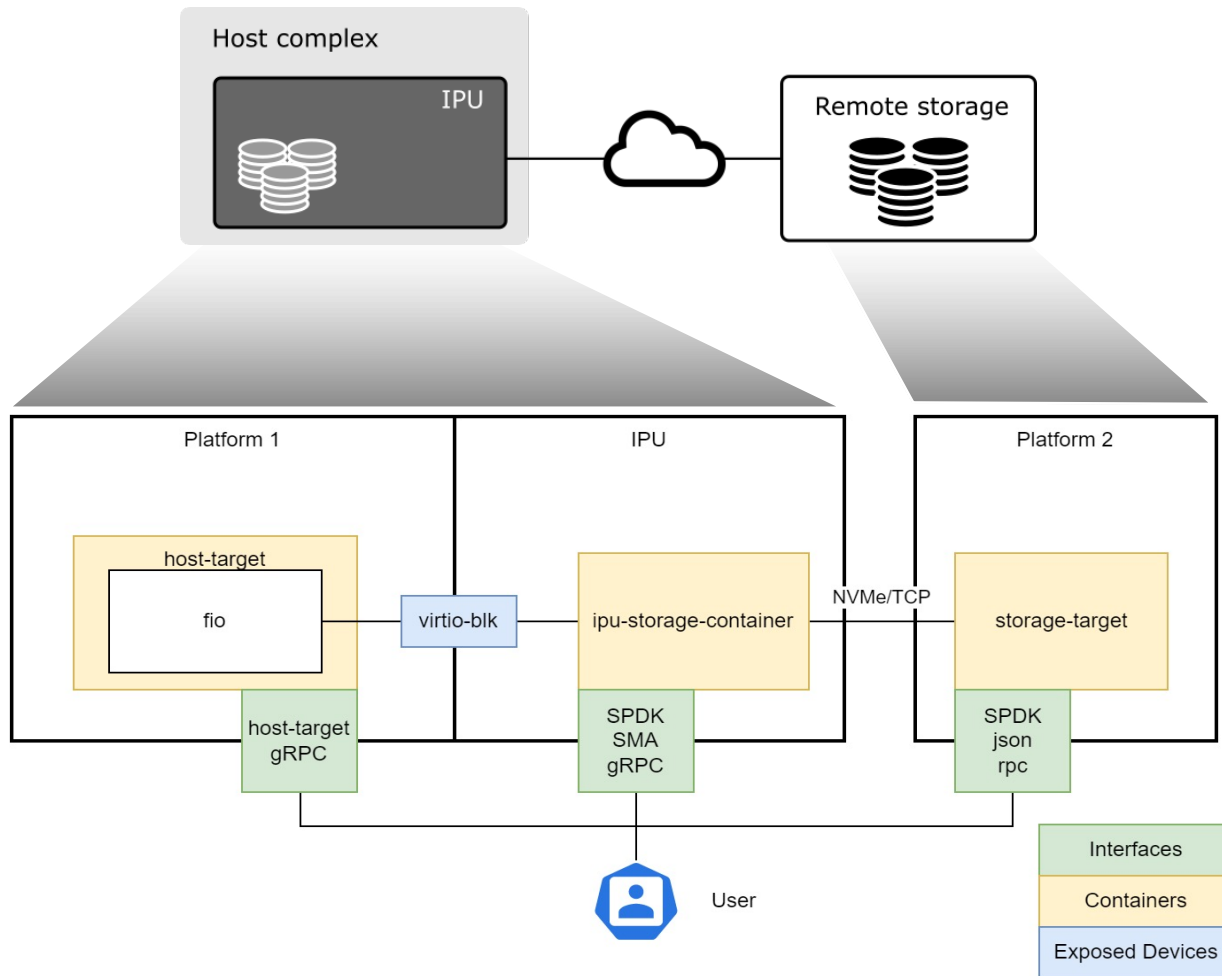
THEORY — PERFORMANCE AND SCALABILITY



IPU with IPDK transparently provides local storage abstraction and remote access to it for host-side applications while in addition ...

- preserving/boosting performance,
- enabling hot-plug/flexibility,
- remaining HW- and vendor-agnostic,

THEORY — USE OF CONTAINERS

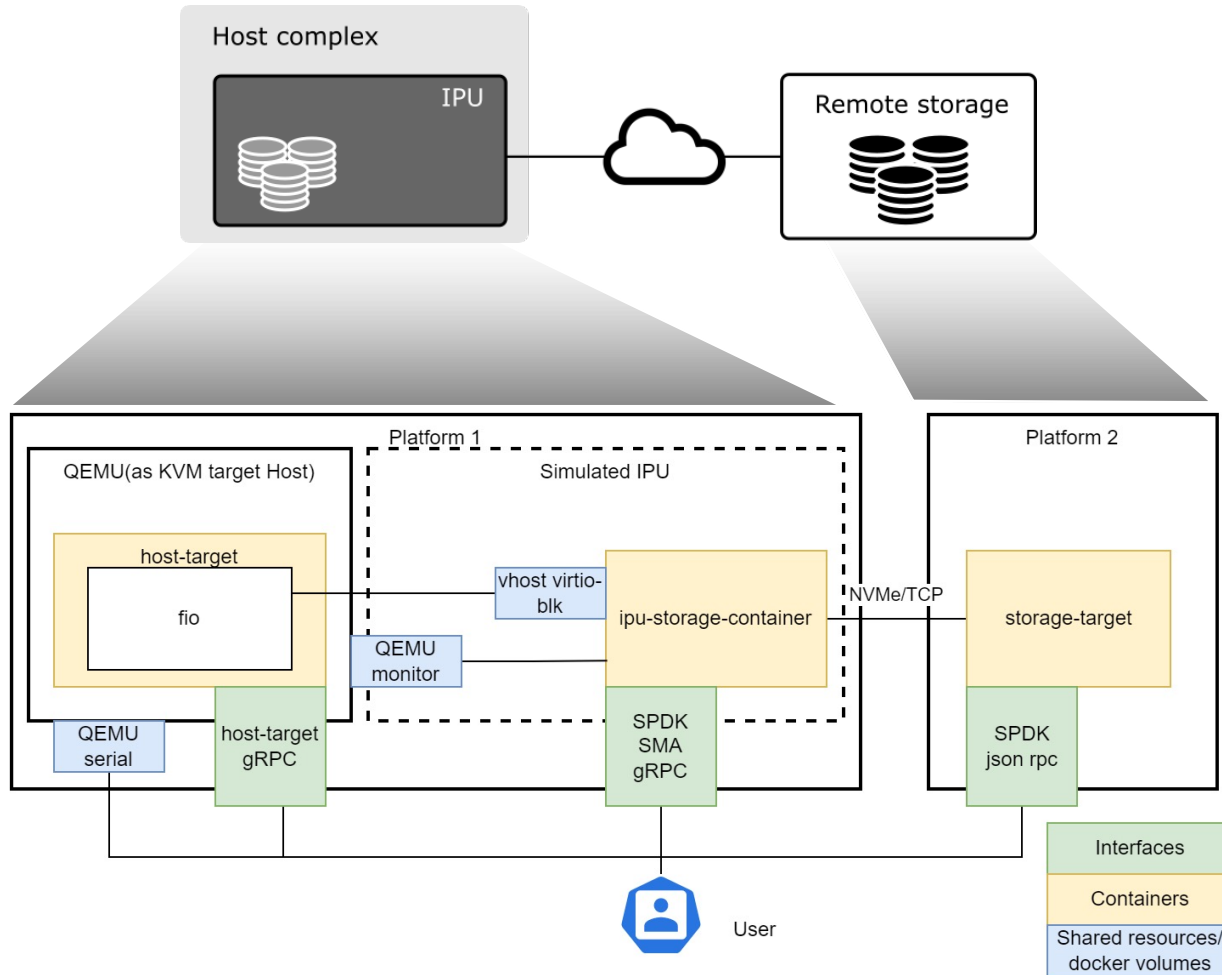


IPU with IPDK transparently provides local storage abstraction and remote access to it for host-side applications while in addition ...

- preserving/boosting performance,
- enabling hot-plug/flexibility,
- remaining HW- and vendor-agnostic,
- being easily deployable and testable,

Solution based on containers.

THEORY — IPU SIMULATION



IPU with IPDK transparently provides local storage abstraction and remote access to it for host-side applications while in addition ...

- preserving/boosting performance,
- enabling hot-plug/flexibility,
- remaining HW- and vendor-agnostic,
- being easily testable,
- enabling rapid-prototyping.

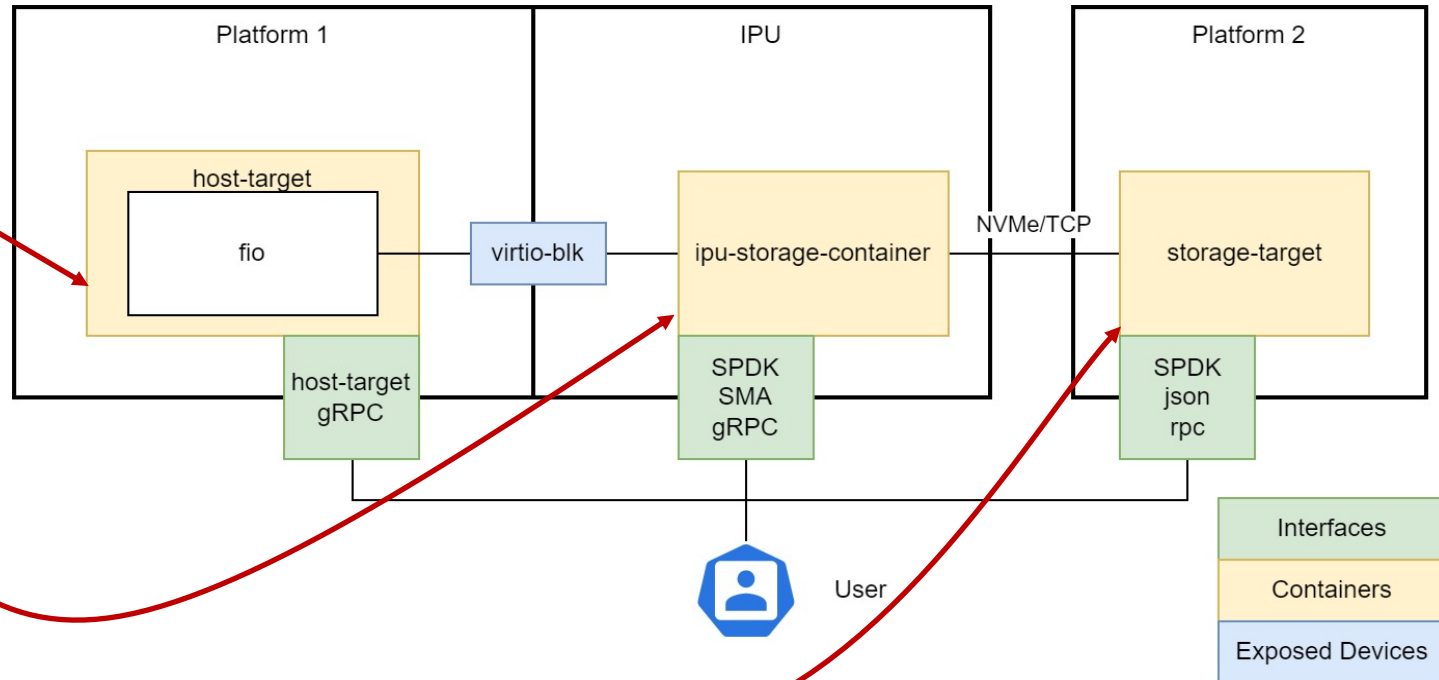
SW-based IPU-target provided.

THEORY – DETAIL VIEW HW-SOLUTION

host-target – runs fio payload over the exposed virtio-blk devices within host.

ipu-storage-container – runs SPDK SMA which connects to storage-target NVMe/TCP device and exposes virtio-blk to host

storage-target – responsible for creating ramdrive bdevs and exposes them over NVMe/TCP.



WIP FOR ENABLEMENT OF MULTIPLE HW-PLATFORMS.

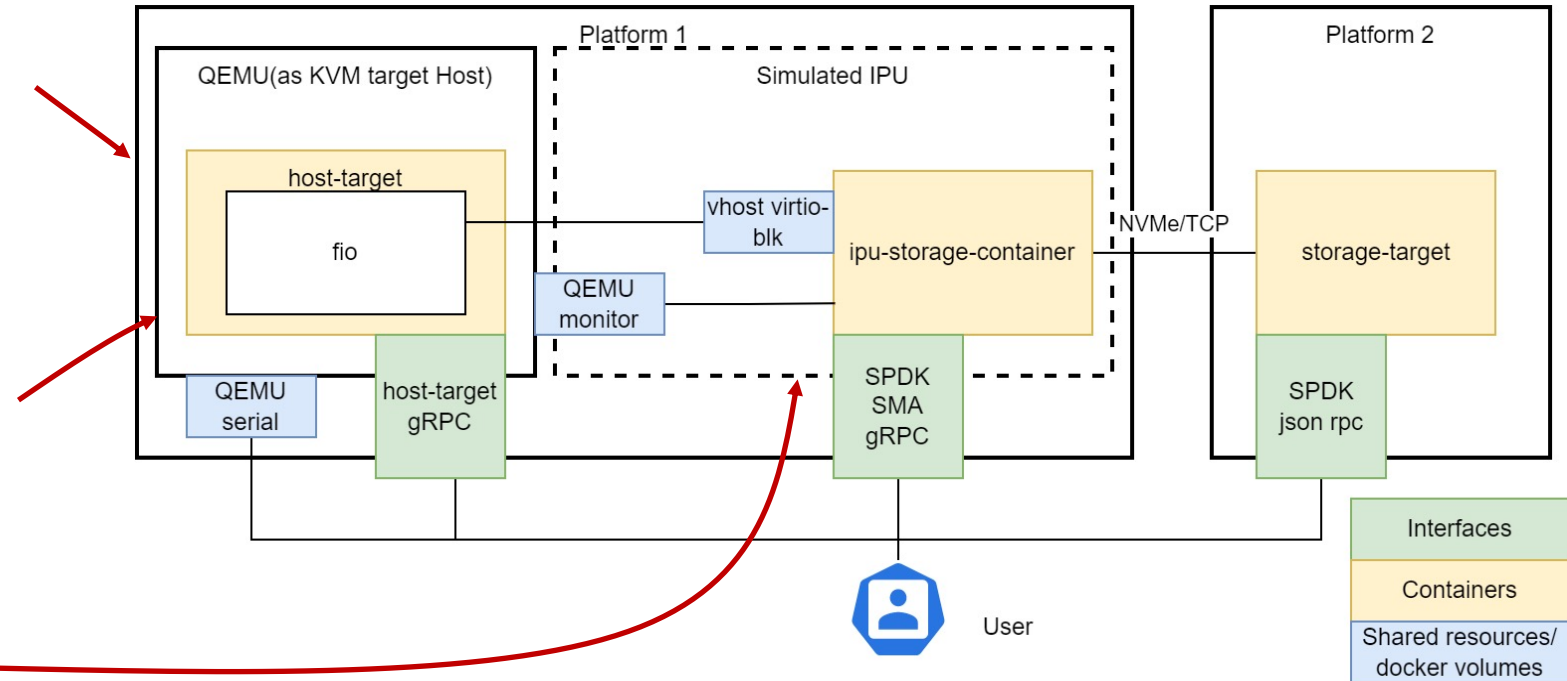
- CONTACT INTEL FOR DETAILS -

THEORY — DETAIL VIEW SW-SOLUTION

Platform 1 contains components to represent HW configuration case in SW

QEMU instance represents physical host attached to HW IPU in sw-solution

Simulated IPU is represented by running dedicated container Interacting with QEMU instance



THIS SOLUTION WILL BE SHOWN IN TODAY'S DEMO.

PRACTISE — INTERACTIVE DEMO

Scenarios to be covered during the demo:

- Hot-plug/unplug 64 virtio-blk disks backed by NVMe/TCP devices to KVM target
- Run traffic over one of the attached virtio-blk disks

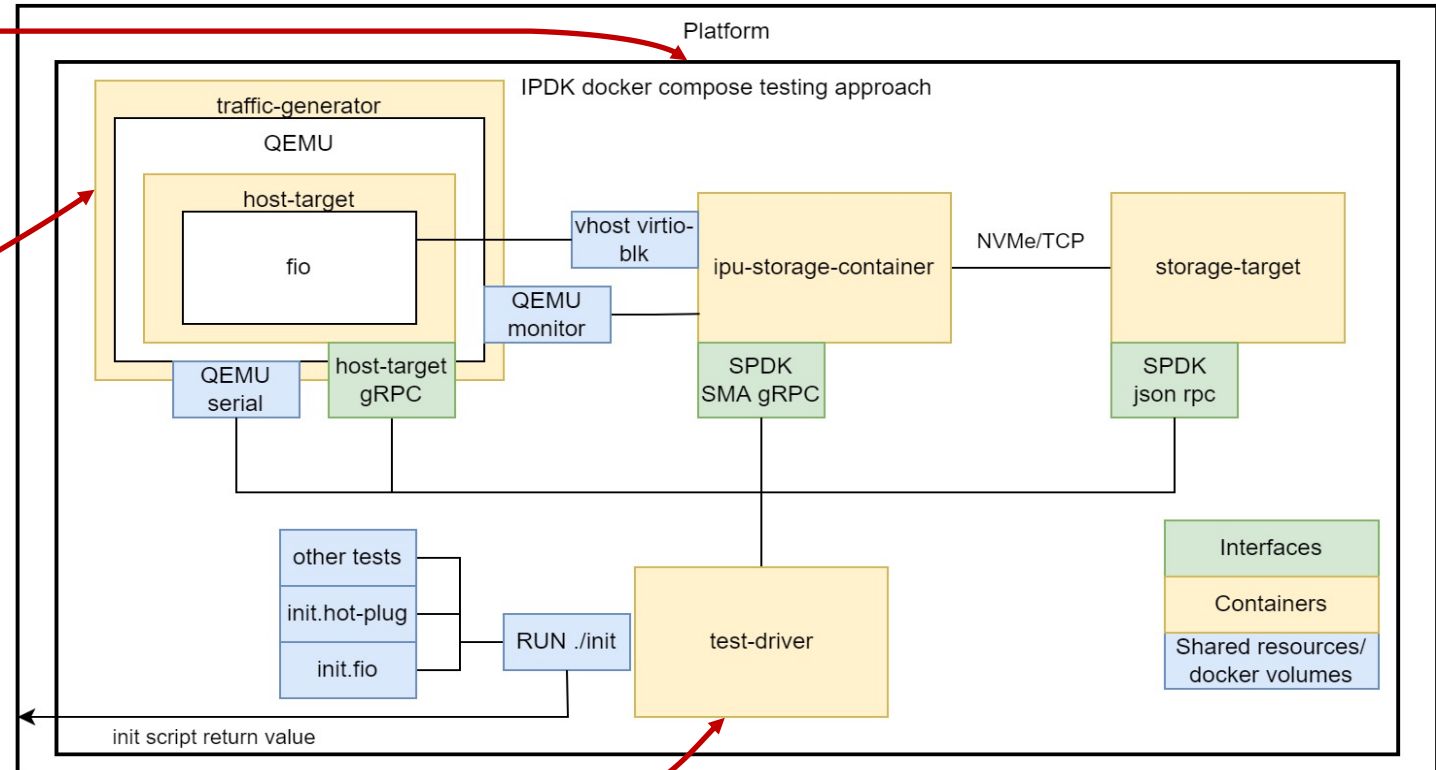
... now it is demo time 😊

PRACTISE – INTEGRATION TESTS INSIDE

docker-compose is used to run all containers on single platform

traffic-generator – contains Fedora image running in qemu. Exposes QEMU monitor and serial interfaces.

test-driver – runs test scenarios, issue configuration commands to all other components, check their states.



PRACTISE — INTEGRATION TESTS DEMO

- Attach a virtio-blk, detach it and re-attach again
- Run fio traffic over an attached virtio-blk
- Attach 64 virtio-blk disks and detach them

... it is demo time again 😊

PRACTISE — IT IS ALL ON GITHUB

You can find all demo code and setup instructions on GH

<http://www.github.com/ipdk-io>

- [Recipes how to run the scenarios manually](#)
- [Integration tests for KVM](#)

For detail questions or HW-platform support reach out via:

- Slack <http://www.slack.com/ipdk> (#storage channel)
- E-mail dan.daly@intel.com

NEAR FUTURE — IMPROVEMENTS TO COME

- Support for NVMe as host-facing interface
- Support for multiple HW-accelerated IPU's
- Data-at-rest encryption on the host
- QoS — bandwidth and rate limiters
- Kubernetes CSI plugin
- Common management interface for virtual devices (vPorts and vDisks) based on OpenConfig
- Virtualization use cases and Live Migration of VMs
- ... and many more to come ...

Besides — it is FOSS and **we would like to hear from you!**

Thank you.

IPDK.io: Infrastructure Programmer Development Kit
Collaborate with the community on Github & Slack

Infrastructure Programmer

Development Kit (IPDK) is an open source, vendor agnostic framework of drivers and APIs for infrastructure offload and management that runs on a CPU, IPU, DPU or switch. IPDK runs in Linux and uses a set of well-established tools such as SPDK, DPDK and P4 to enable network virtualization, storage virtualization, workload provisioning, root-of-trust and offload capabilities found in the platform. IPDK provides a common platform for increasing performance, optimizing resources and securing the infrastructure as a sub-project of [Open Programmable Infrastructure](#), a [Linux Foundation Project](#).

