STORAGE DEVELOPER CONFERENCE
SDC 22
Fremont, CA
September 12-15, 2022
BY Developers FOR Developers

A SNIA Event

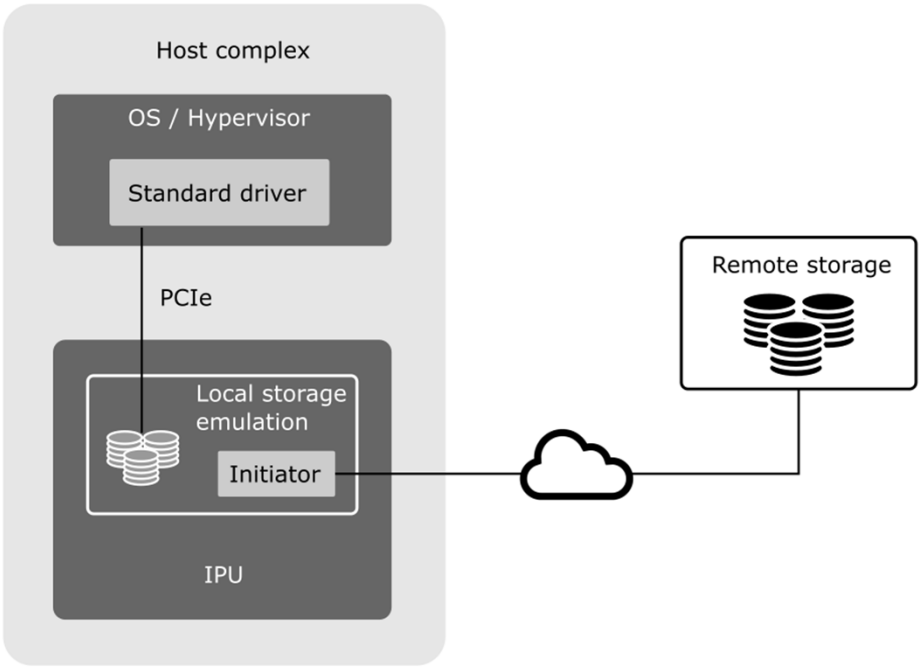
Storage Virtualization and HW-agnostic Acceleration using IPDK and xPU

Presented by Deb Chatterjee, Senior Director of Engineering, **Intel Corporation**

Agenda

- Emulation of Local Storage
- Introduction to xPU
- Introduction to IPDK
- Storage virtualization on IPDK SW target
- Moving storage usecases to IPU as IPDK HW target
- Next steps

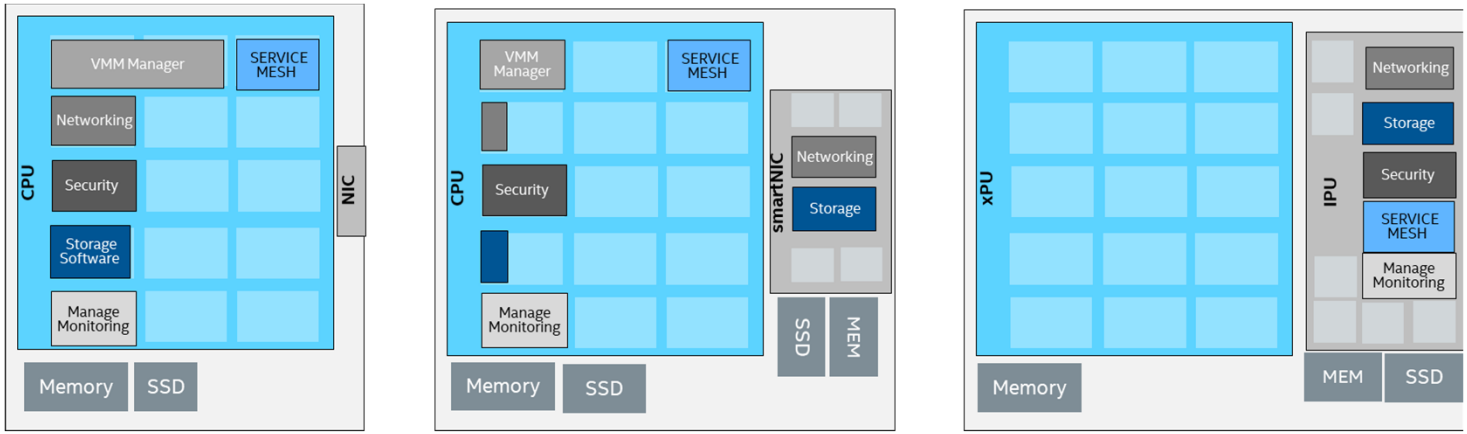
Local storage emulation



- Abstraction over HW-storage resources
- Enablement of storage disaggregation
- Programmability
- Transparency to client-side SW incl. drivers
- Performance with dedicated HW
- Improved security

IPU provides storage to host by connecting to the remote disk and virtio-blk device using SPDK back-end application running in IPU.

Emergence of the xPU

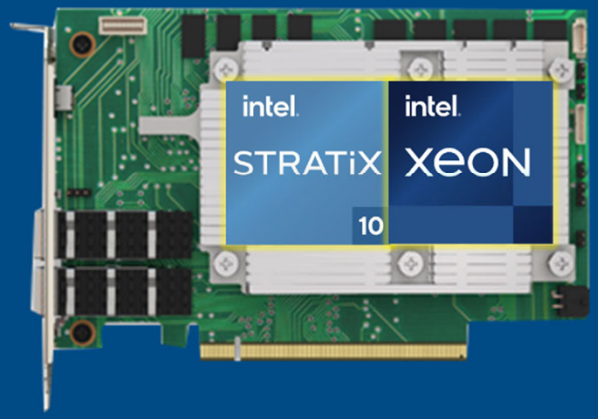


1. Efficient high-performance software programmable multi-core processors
2. Flexible and programmable acceleration engines
3. Stream based interfaces for high performance data movement to attached xPU (within Intel we call it IPU; other companies may call it DPU or some other name)

IPU – Big Spring Canyon - Intel® FPGA IPU C5000X-PL

Features

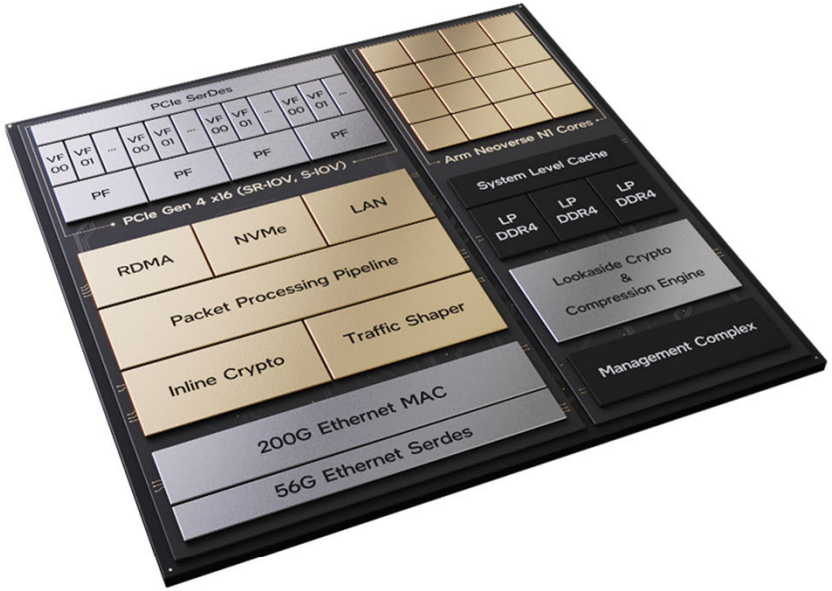
- Intel® Xeon® D processor and FPGA SmartNIC combined platform
- Virtual networking and storage on single card
- Standard shell for virtio-net, virtio-blk, NVMe, RoCE
- Software and hardware programmable



Benefits

- Large software ecosystem
- High performance
- Standard software
- Versatile across data center
- Easier getting started
- Resilient to future change
- Customizable

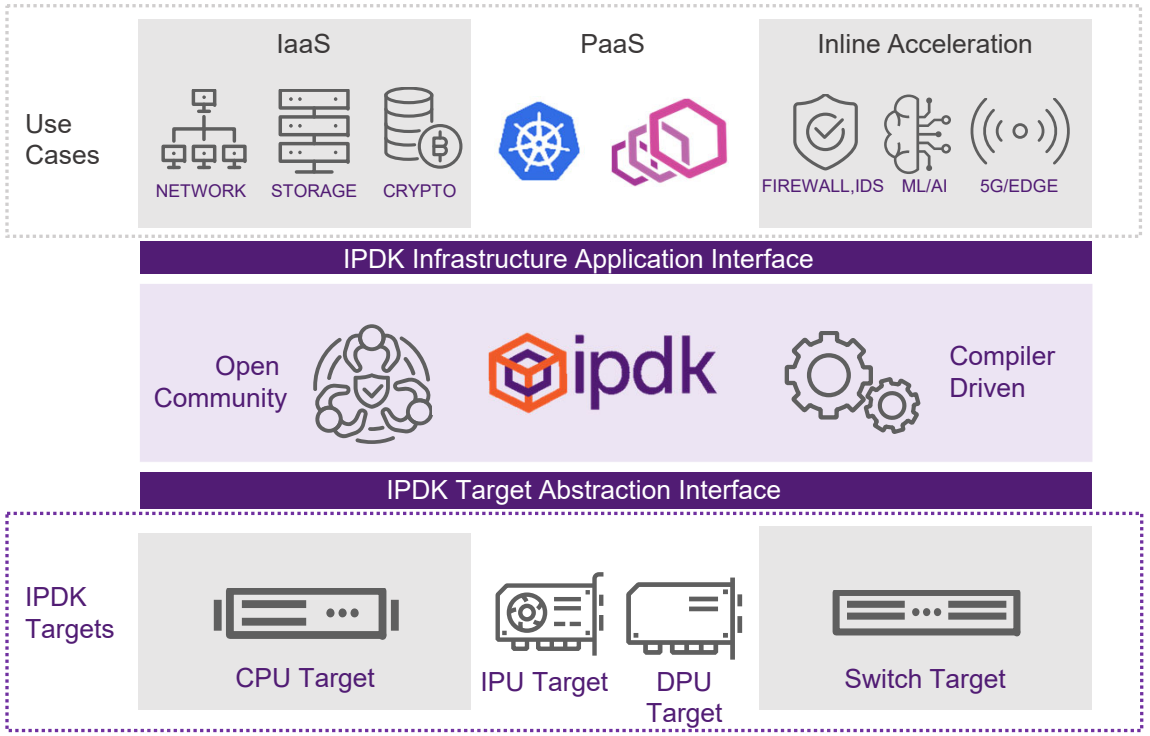
IPU - Intel® Mount Evans 200G ASIC IPU



Hyperscale Ready	Co-designed with a top cloud provider Integrated learnings from multiple gen. of FPGA IPU High performance under real world load Security and isolation from the ground up
Technology Innovation	Best-in-Class Programmable Packet Processing Engine NVMe storage interface scaled up from Intel Optane Next Generation Reliable Transport Advanced crypto and compression accel.
Software	SW/HW/Accel co-design P4 Studio based on Barefoot technology Linux OS leveraging DPDK, SPDK & IPDK eco-systems Enable broad adoption of IPUs

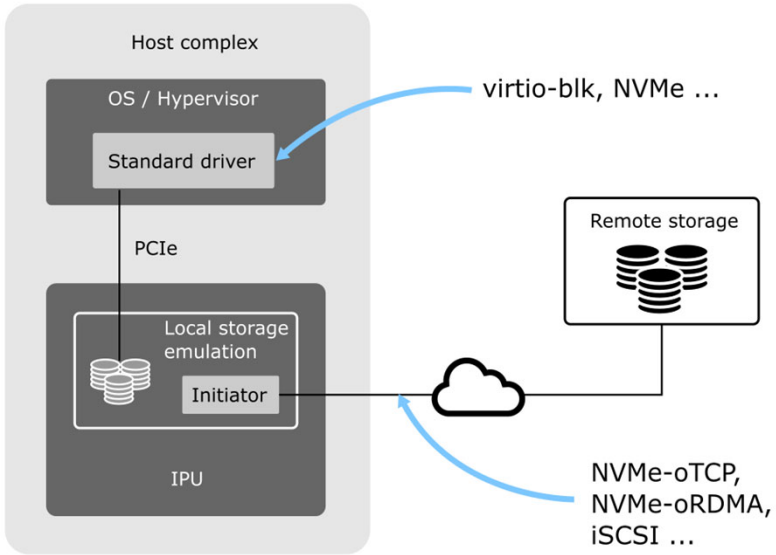
Infrastructure programmer development kit (IPDK)

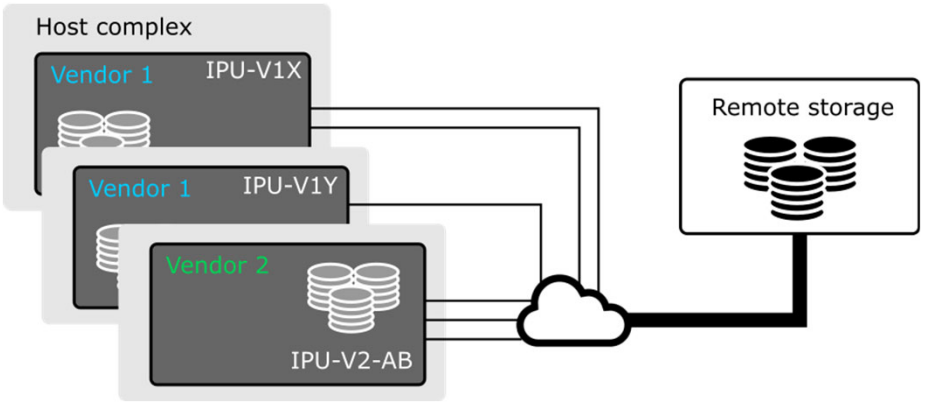
- IPDK is a development framework
- **community-driven**
- **target agnostic**
- runs on CPU, IPU, DPU or switch.



IPDK storage usecase running on IPU

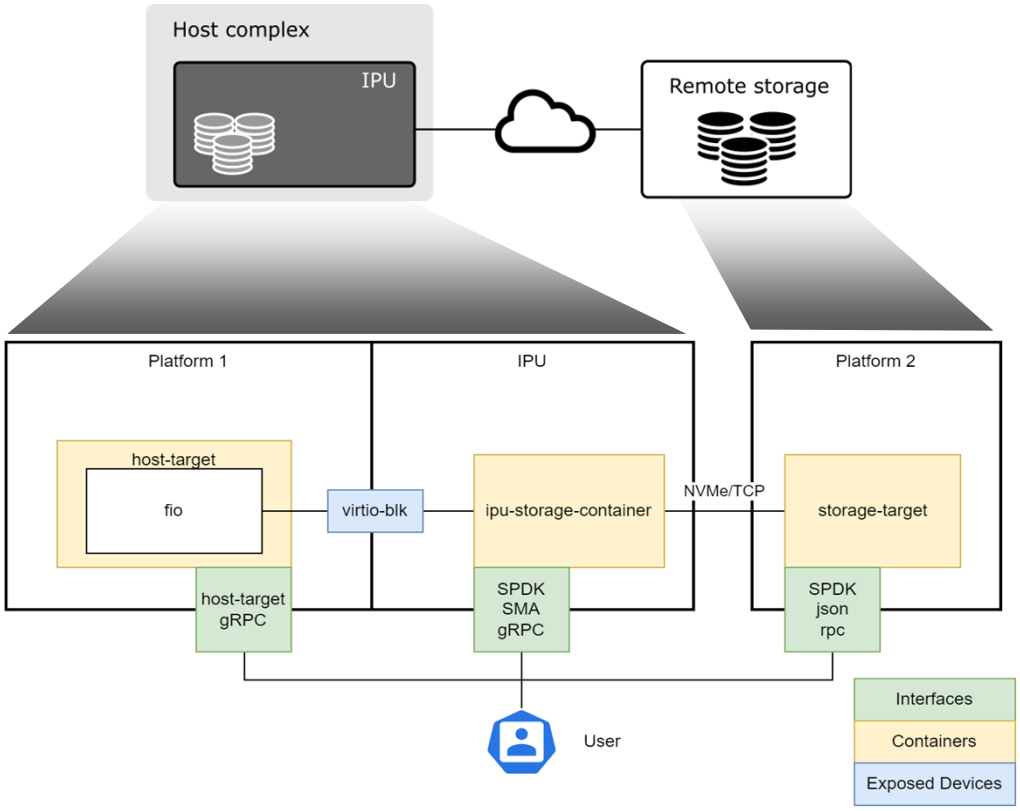
- IPDK supports multitude of storage protocols
- virtio-blk and NVMe as host-facing protocols





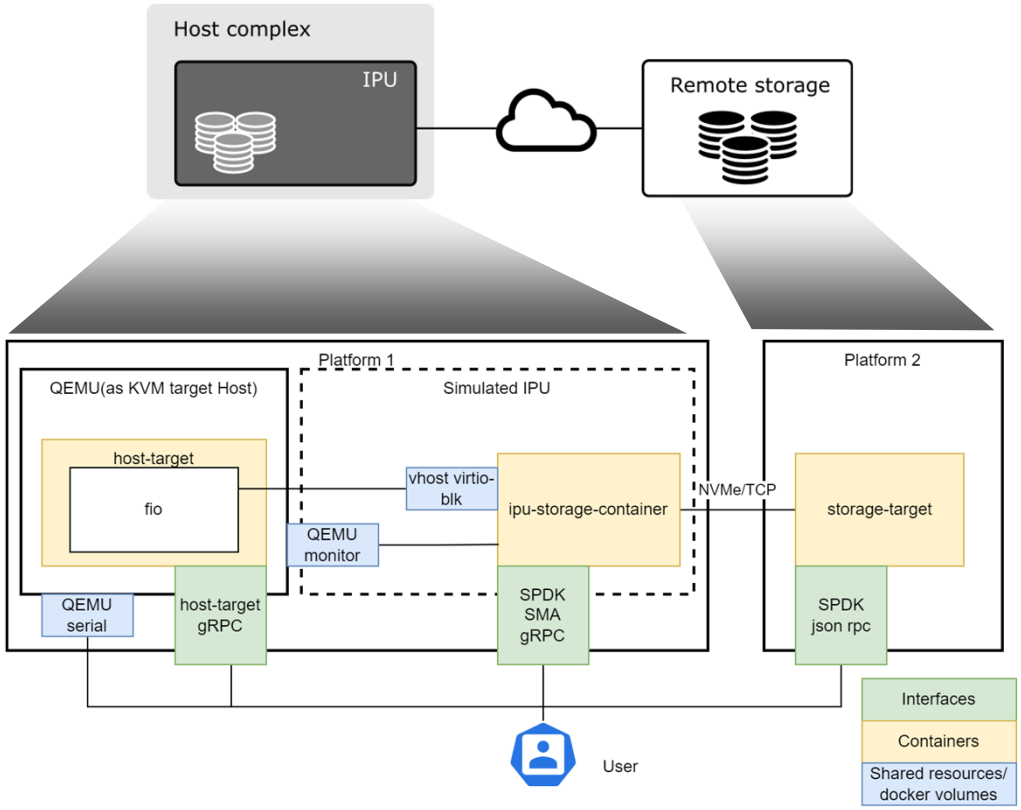
- HW- and vendor-agnostic by design
- Accelerated by dedicated HW-processing (offload of storage stack, crypto/compression…)

Container-based development environment of IPDK



- Containers as integrated SW-stacks
- Easily deployable
- Easily testable

Storage virtualization using IPDK SW target



Pure SW target that can run on any general-purpose CPU.

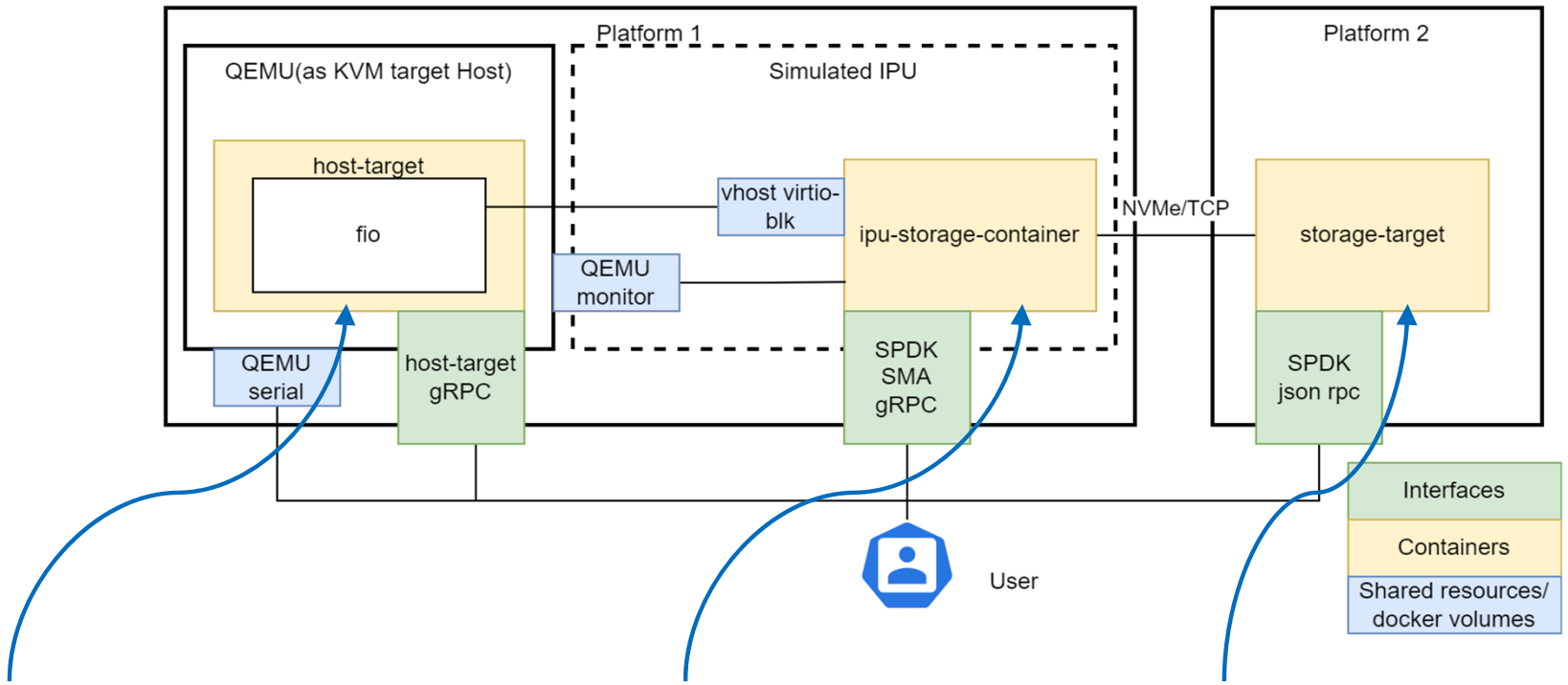
Platform 1 contains components to represent HW configuration case in SW

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

IPDK SW target is within the dedicated container interacting with QEMU instance

Enabling rapid prototyping without HW

IPDK container functionalities



host-target – runs traffic (e.g. 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.

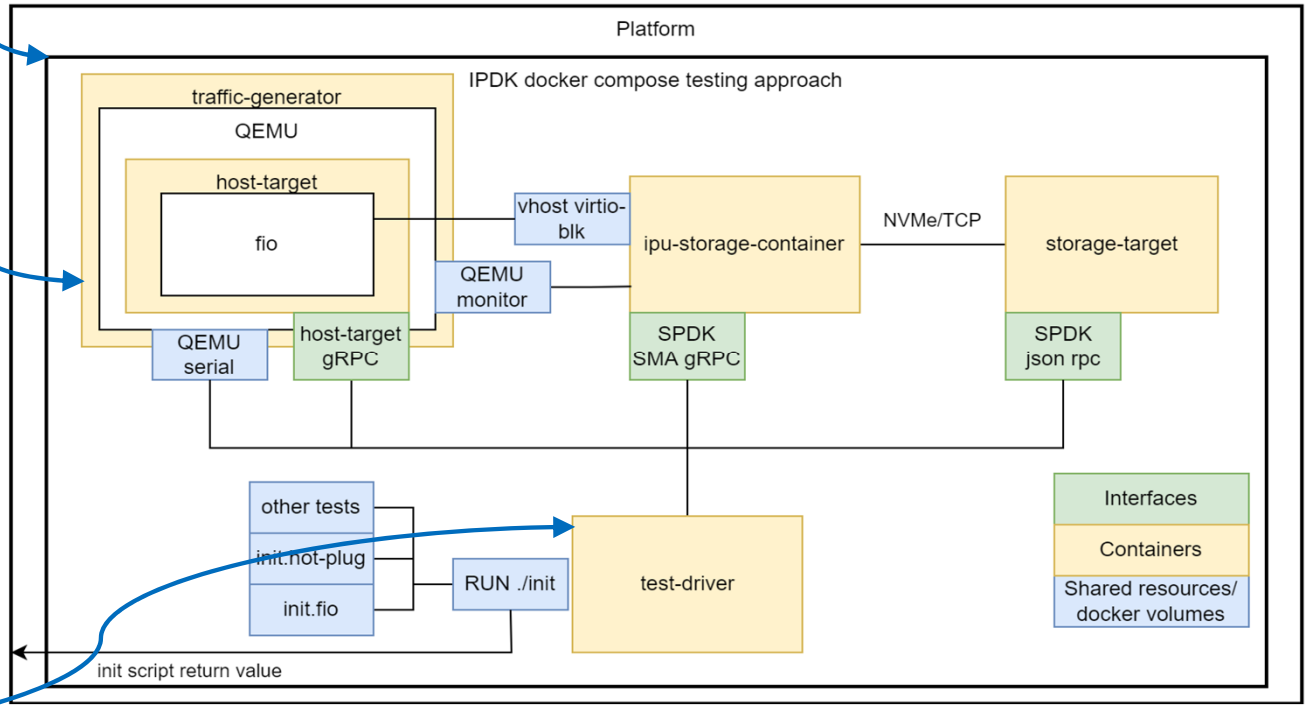
storage-target – responsible for creating ramdrive bdevs and exposes them over NVMe/TCP.

Integrated testing on a single machine

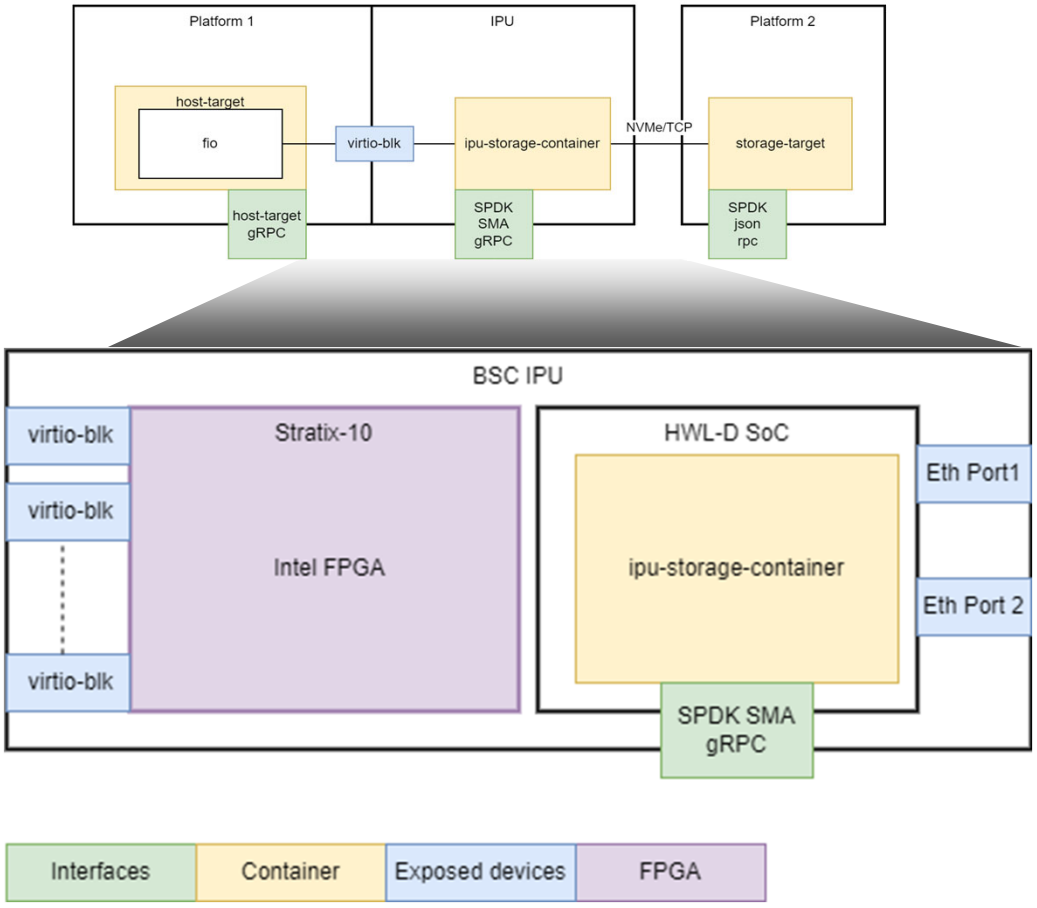
docker-compose is used to run all containers on single platform

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.



IPDK – Big Spring Canyon (BSC) IPU



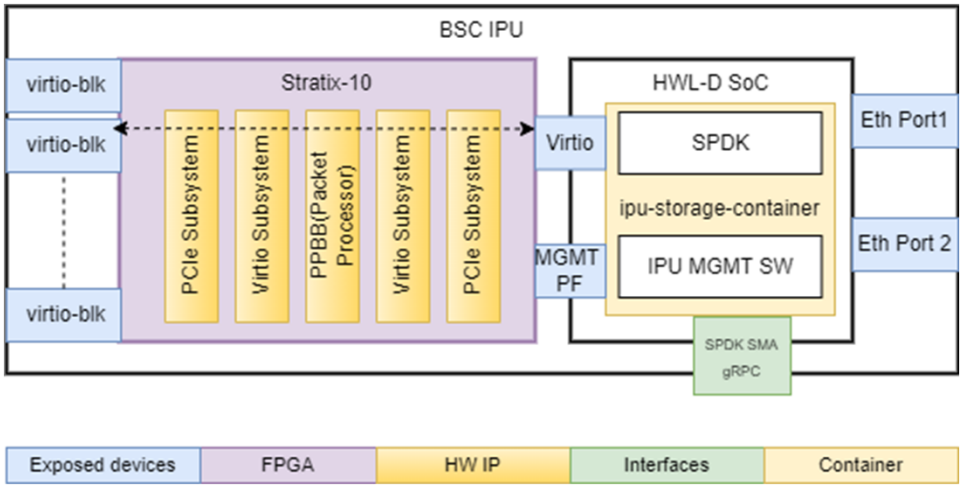
Big Spring Canyon IPU is a a Stratix-10-based FPGA IPU containing FPGA IPs and Hewitt Lake-D (HWL-D) SoC.

BSC supports storage offload up to 50 Gbps. The storage backend SW and control path software run in SoC.

The host-target and storage-target containers are HW-independent. The ipu-storage-container is HW-specific.

Exact same flow as the IPDK SW target

BSC IPU HW overview



Host-facing interface:

- 16 virtio-blk physical function (PFs)
- Each virtio-blk PF has up to 8 virtqueues

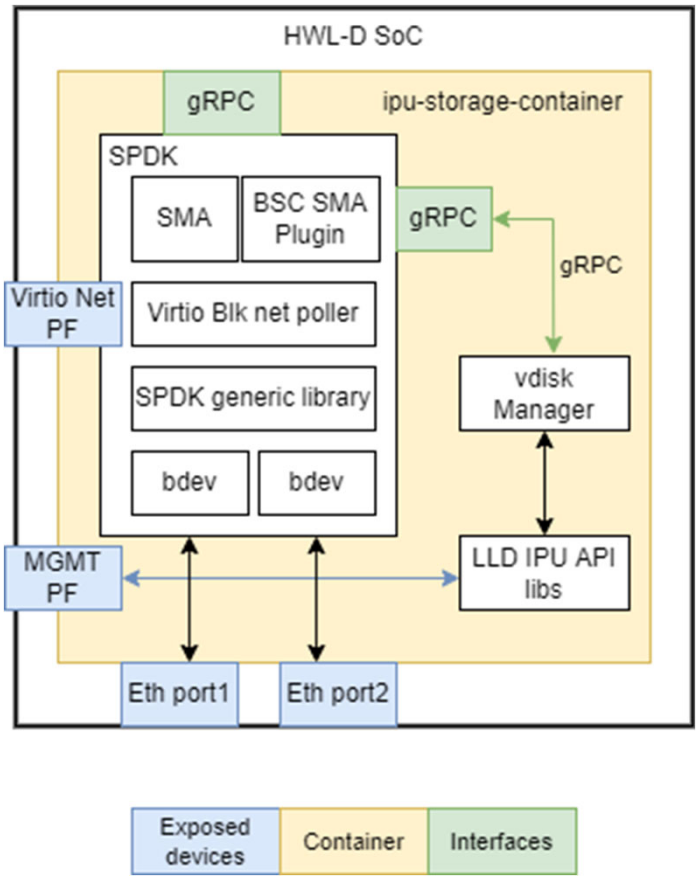
Packet Processor Block (PPBB):

- For mapping Host/SoC/Line interfaces.
- Add/Remove Intel Meta Tag

SoC-facing interface:

- 1 management PF for configuring IPU
- 1 virtio-net physical function for storage packet processing
- 2 virtio-net physical functions for 2x25G interface to connect to external server.

Big Spring Canyon IPU SW overview



Storage Management Agent (SMA) - gRPC interface for orchestrating SPDK applications.

Virtio Blk net poller: conduit between the host initiator and remote storage

LLD IPU APIs libs for configuring virtio-blk/-net interfaces of IPU and enabling hot-plug add/remove.

Examples of SW APIs

1. Storage-Target container for creating remote disk.
    a. create_and_expose_sybsystem_over_tcp -> (nvmf_create_subsystem, nvmf_create_transport, nvmf_subsystem_add_listener)
    b. create_ramdrive_and_attach_as_ns_to_subsystem -> (bdev_malloc_create, nvmf_subsystem_add_ns)
2. IPU-Storage-Container for configuring IPU and to start storage back-end application.
    a. SMA server to process storage config APIs
        i. CreateDevice
        ii. AttachVolume (not used)
        iii. DeleteDevice
    b. SMA plug-in(BSC/Vhost) to connect to platform specific IPU APIs
        i. create_device
        ii. delete_device
    c. IPU APIs
        i. ifc_add_port => Blk port default config
        ii. ifc_set_port_disk_capacity => configure capacity
        iii. ifc_add_sep => hotplug add
        iv. ifc_remove_sep => hotplug remove

Next steps

- Upstreaming of NVMe as host-facing interface
- Support for multiple HW-accelerated IPUs with performance benchmarking
- Data-at-rest encryption on the host
- QoS – bandwith 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
- ...

Contributors

Basavaraj (Basa) Kamadinni
Cloud SW Engineer
Intel Corp.

Artsiom (Artek) Koltun
Cloud Software Dev. Engineer
Intel Corp.

Filip Szufnarowski
Storage SW Eng. Manager
Intel Corp.

Anbuvelu (Anbu) Venkataraman
Cloud SW Engineer
Intel Corp.

References

Check the IPDK storage solution directly on GH:

https://github.com/ipdk-io/ipdk/tree/main/build/storage/recipes / simulated IPU setup instructions

https://github.com/ipdk-io/ipdk/tree/main/build/storage/tests/it / integration tests on a single host

Reach out for discussion or HW support details:

http://www.slack.com/ipdk (#storage channel) / for general discussion

dan.daly@intel.com / for HW-support requests

Please take a moment to rate this session.

Your feedback is important to us.

