2023 OFA Virtual Workshop

DPFS: DPU-Powered File System Virtualization

Peter-Jan Gootzen 🇨🇭🇳🇱, ***Jonas Pfefferle*** 🇺🇸, ***Radu Stoica*** 🇨🇭, ***Animesh Trivedi*** 🇳🇱

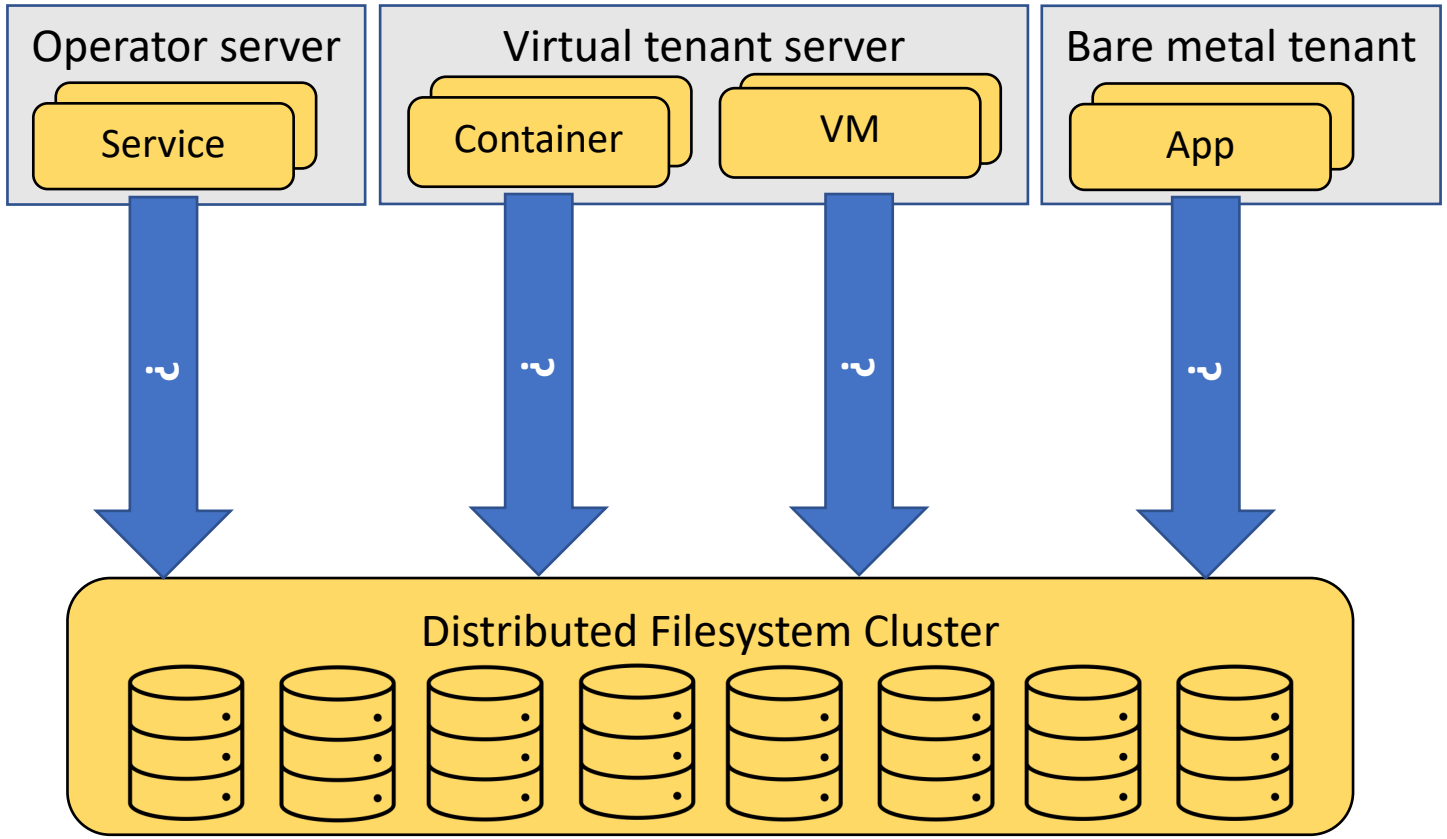
IBM Research
Zurich 🇨🇭 and Yorktown 🇺🇸

IBM Research

Vrije Universiteit Amsterdam 🇳🇱
AtLarge Research

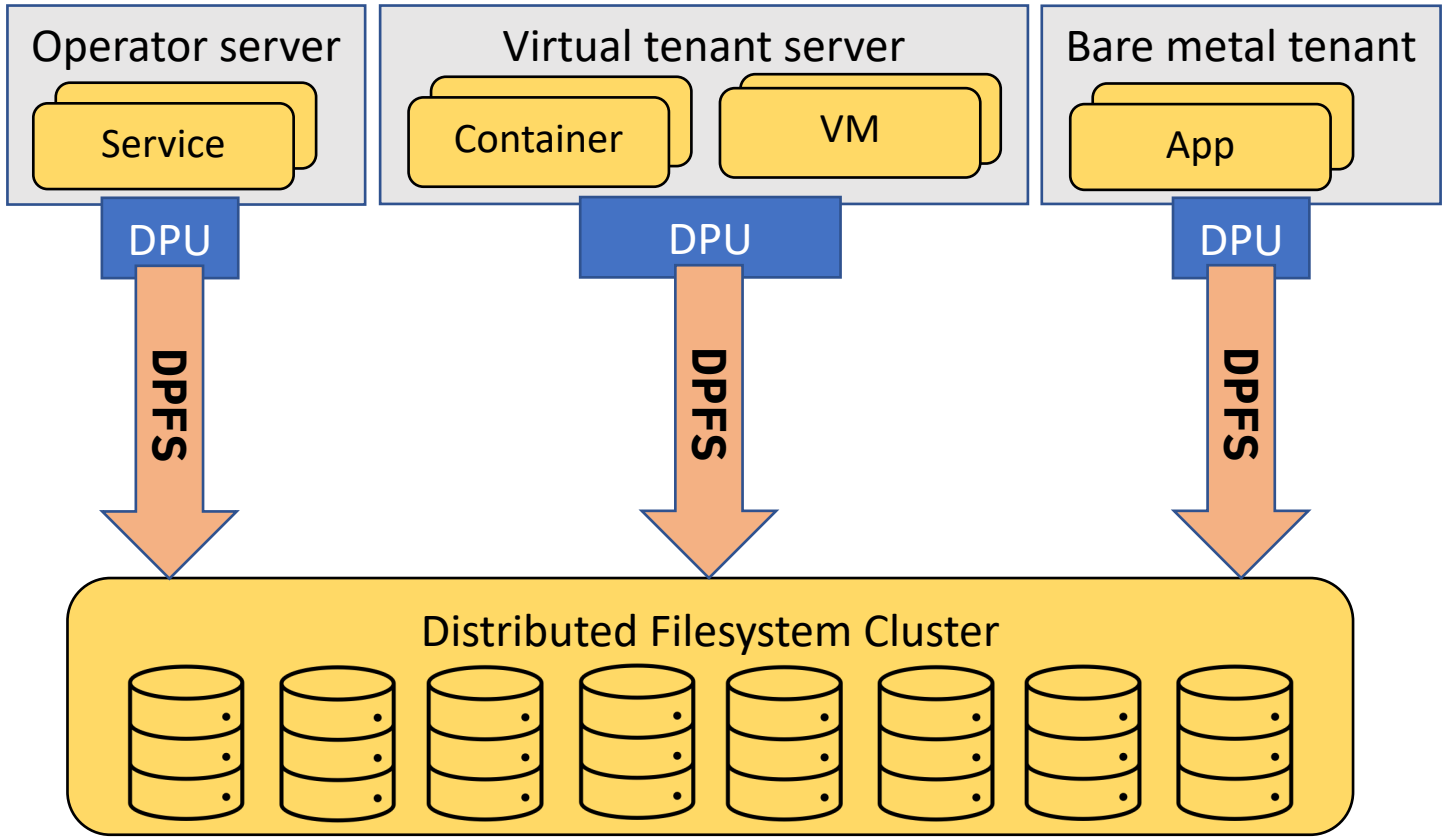
VU VRIJE UNIVERSITEIT AMSTERDAM

How to consume a FS service in the Cloud?



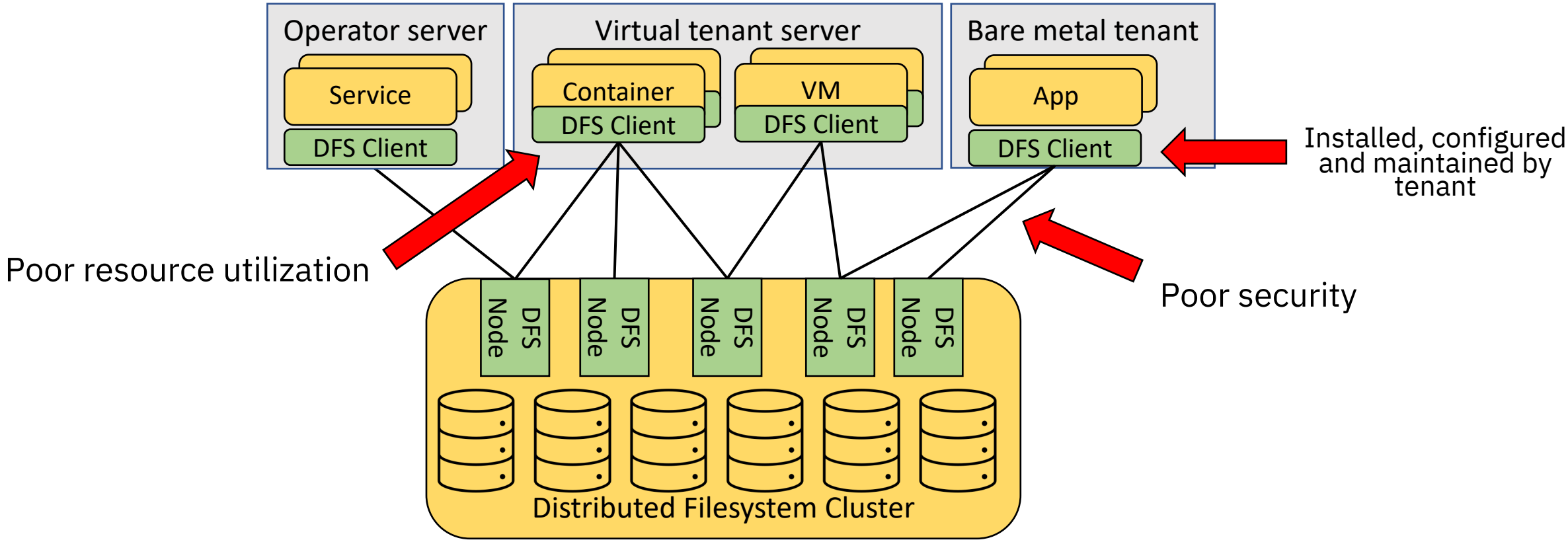
Efficiency			Management			Security	
Performance	Overhead	Multi-tenancy	Support all tenants	Client transparency	Operator control	Attack surface	Network isolation

DPU-Powered File System Virtualization



Efficiency			Management			Security	
Performance	Overhead	Multi-tenancy	Support all tenants	Client transparency	Operator control	Attack surface	Network isolation

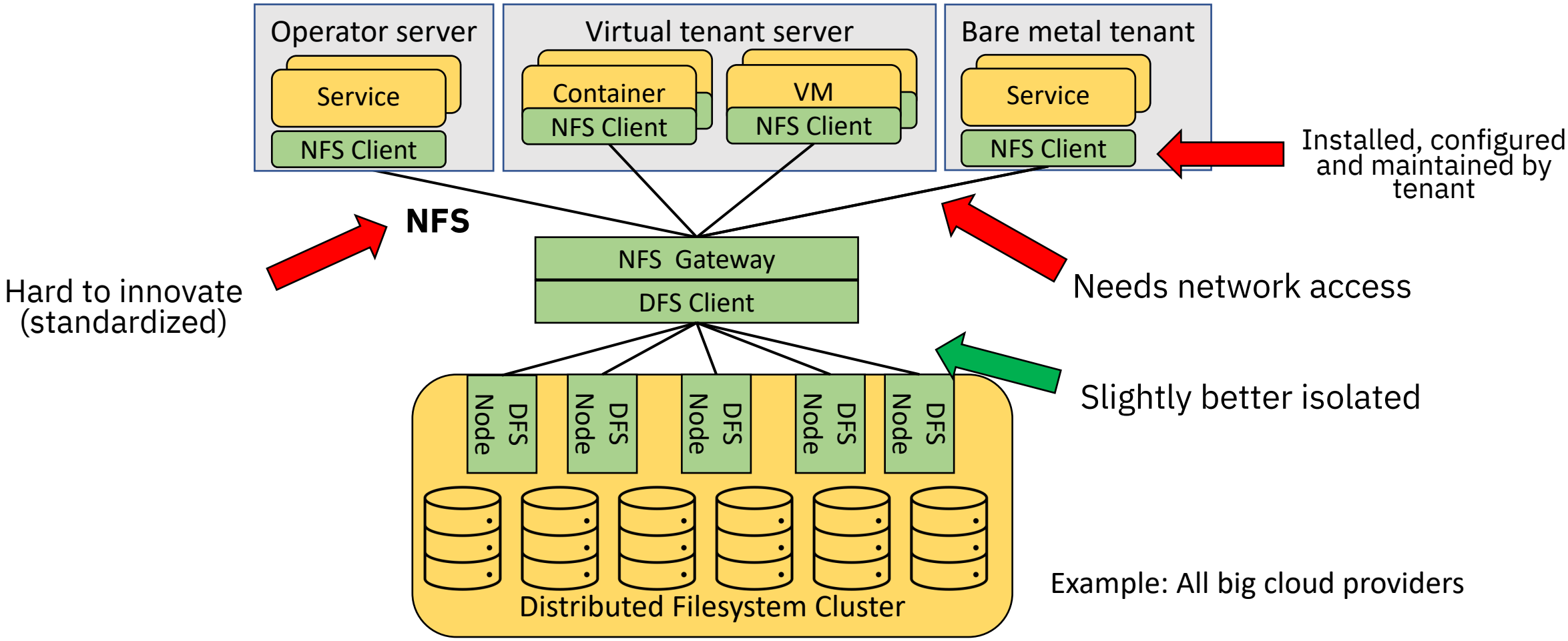
Option 1: *Traditional* Distributed File System client



Example: Spectrum Scale, Ceph etc.

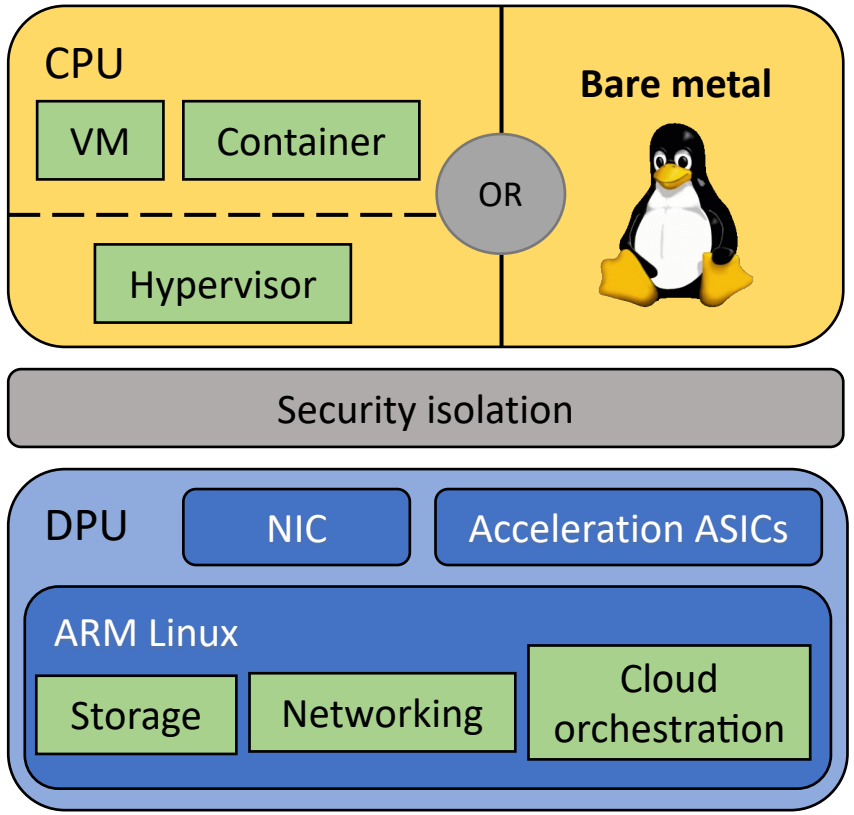
Efficiency	Management	Security

Option 2: NFS gateway for Cloud File Systems



Efficiency	Management	Security

The DPU-powered Cloud ☁️



- Also known as *SmartNIC* or *Infrastrucure Processing Unit (IPU)*
- "A NIC with compute and offload capabilities baked in"
- We focus on DPUs with a CPU

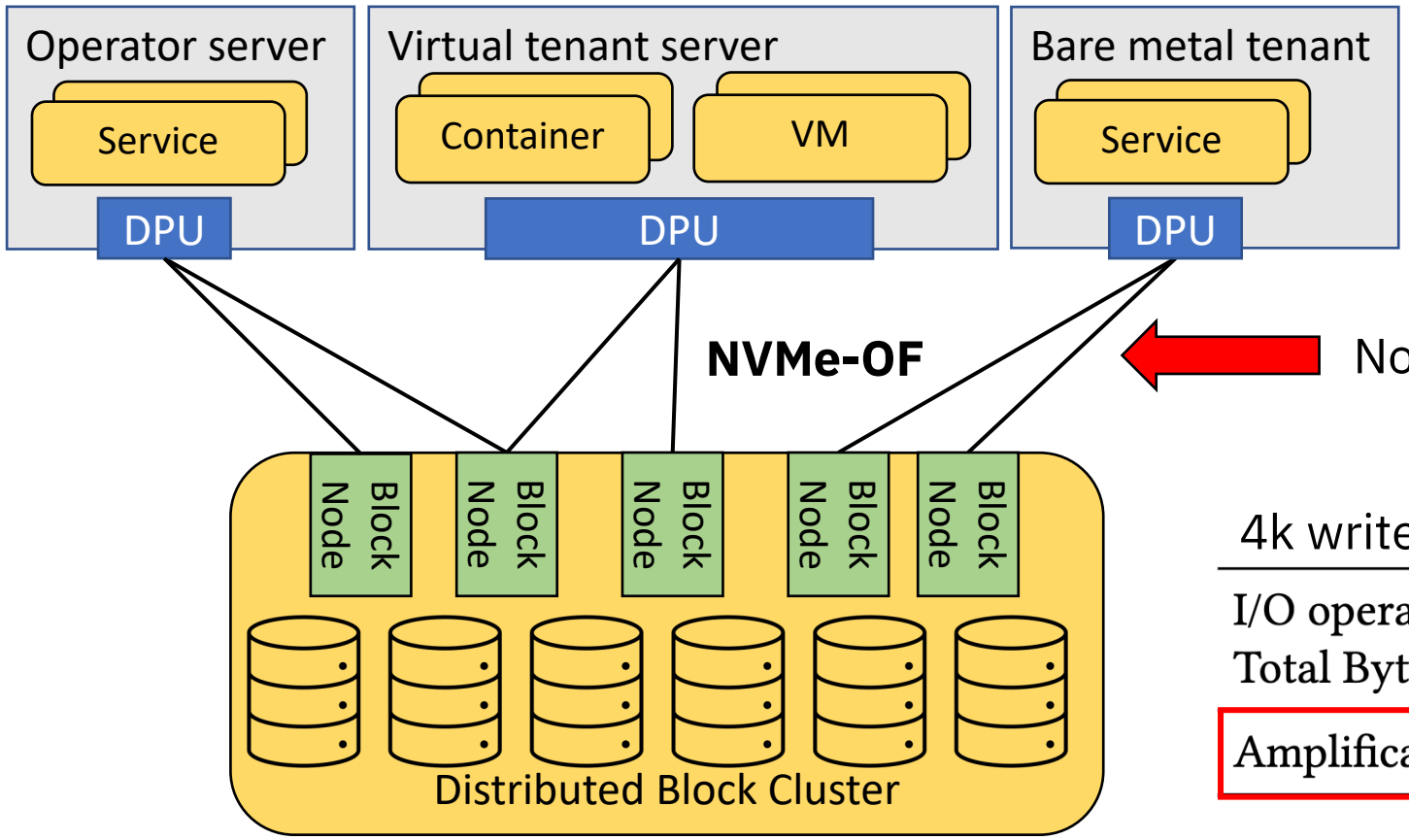
Offloading using DPUs:

✓ Block storage devices (NVMe and virtio-blk)

✓ Networking (virtio-net & programmable switch)

✗ File systems **Insert "DPFS"**

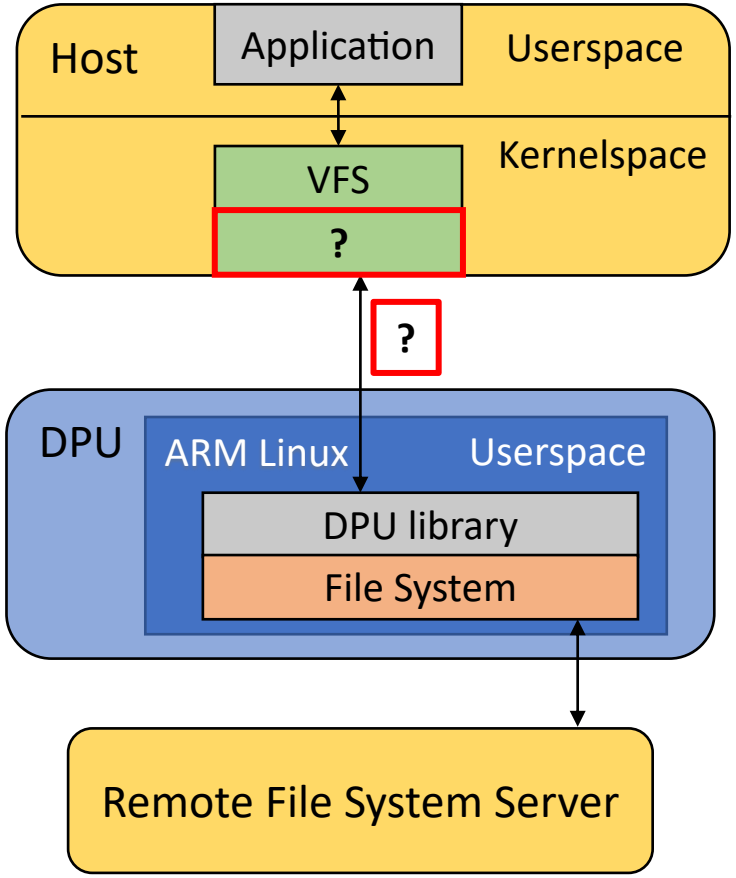
Option 3: Remote Block Storage



4k write	ext4	ext4 + NVMe-oF	XFS	Btrfs
I/O operations	5.2	13.7	3	4.6
Total Bytes (in KiB)	44.7	46.8	12	125.3
Amplification	11.2x	11.7x	3x	16x

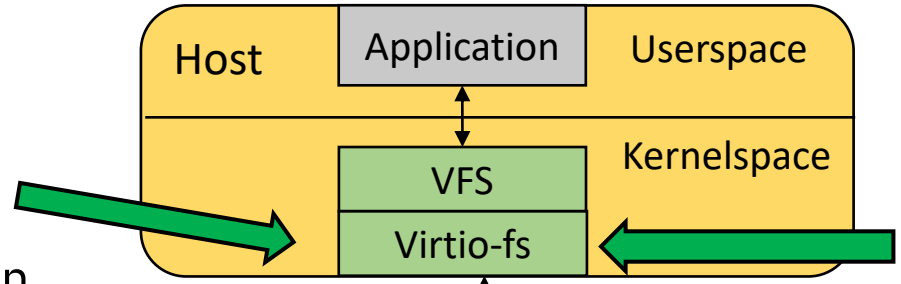
Efficiency | **Management** | **Security**

A DPU-powered abstraction for Cloud File Systems



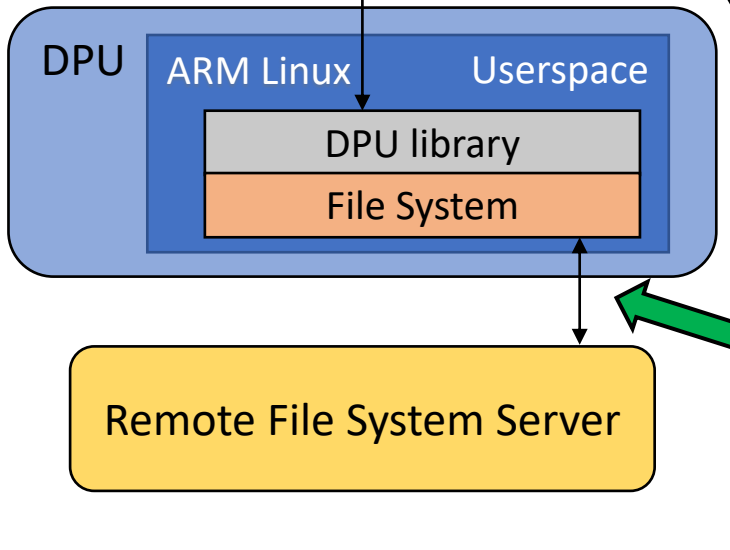
The **Virtio-fs** stack of DPFS

- No configuration
- Works on bare metal
- Transparent consumption of any FS



Virtio-fs = 13k LoC
vs.
(NFS+TCP/IP) = 181k LoC

Virtio-fs over PCIe



Multi-tenancy (SR-IOV)

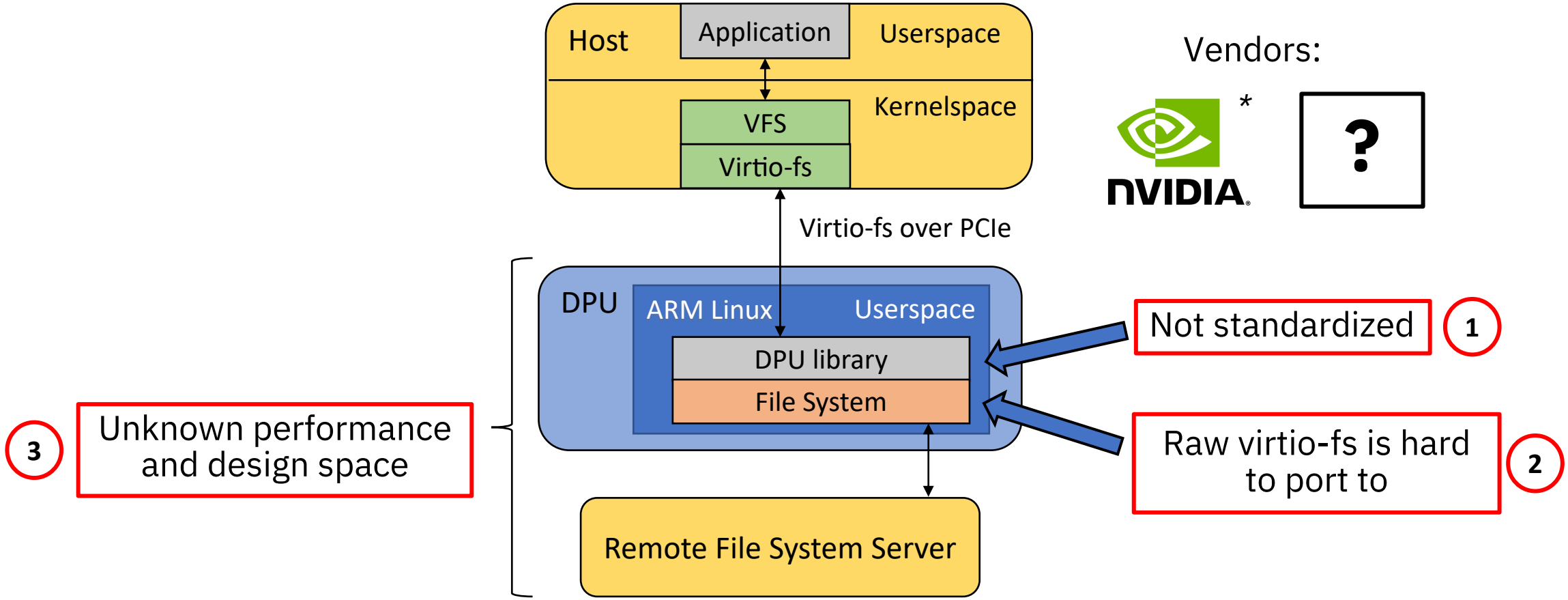
Maximum flexibility and full control for hardware specialization

Tenant completely isolated from FS client and network

Efficiency			Management			Security	
Performance	Overhead	Multi-tenancy	Support all tenants	Client transparency	Operator control	Attack surface	Network isolation

**Currently only available with the limited technical preview program of Nvidia BlueField.*

Challenges that **DPFS** solves

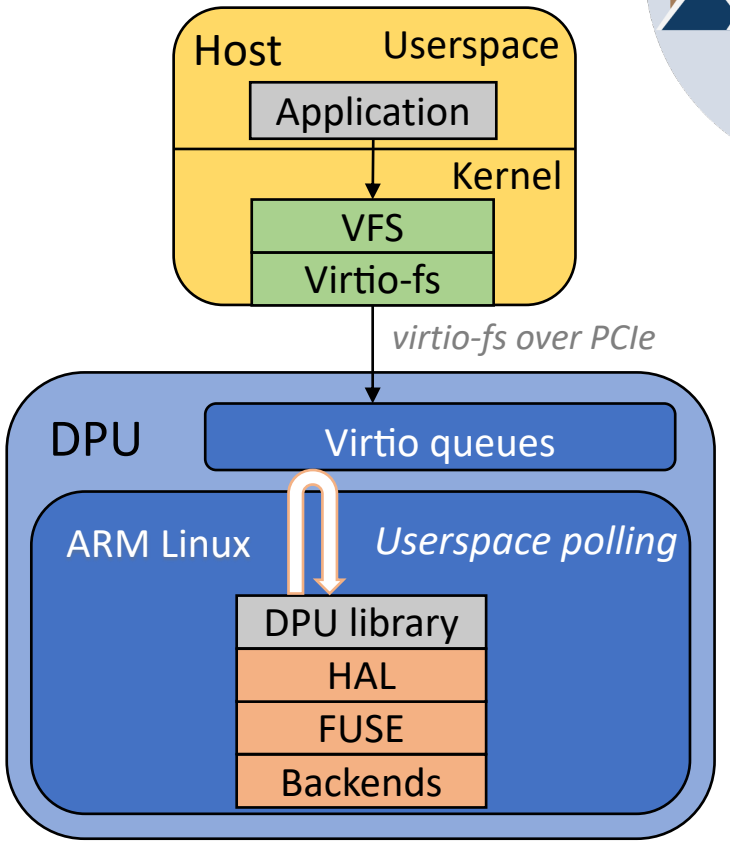


Kick-start open research and adoption!

The **DPFS** framework: **D**PU-**P**owered **F**ile **S**ystems

Architecture:

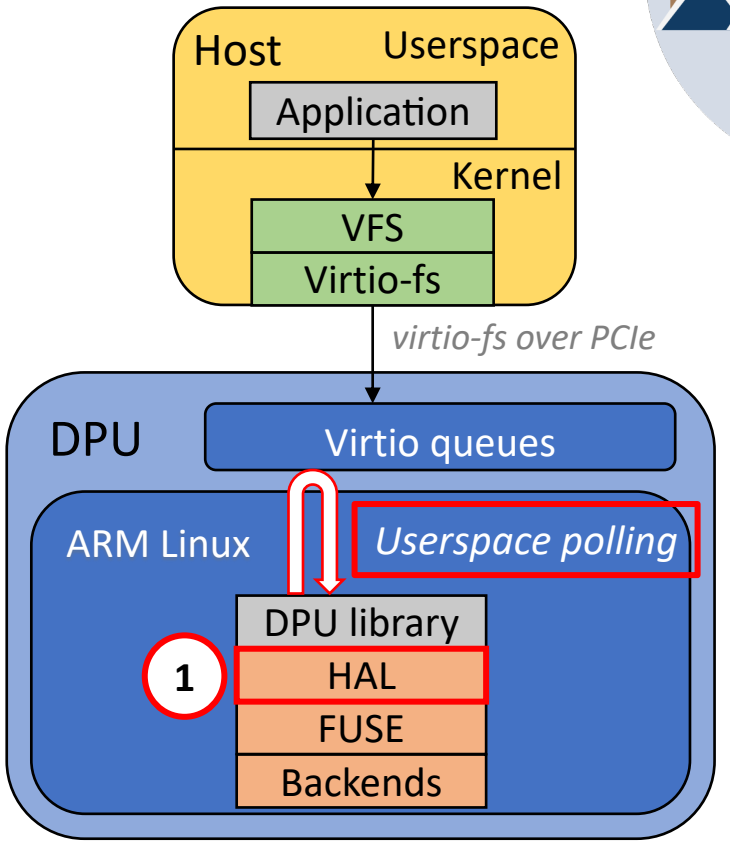
1. Hardware Abstraction Layer
2. FUSE API
3. Several backends



The **DPFS** framework: **D**PU-**P**owered **F**ile **S**ystems

Architecture:

1. **Hardware Abstraction Layer**
2. FUSE API
3. Several backends



Vendors:

*

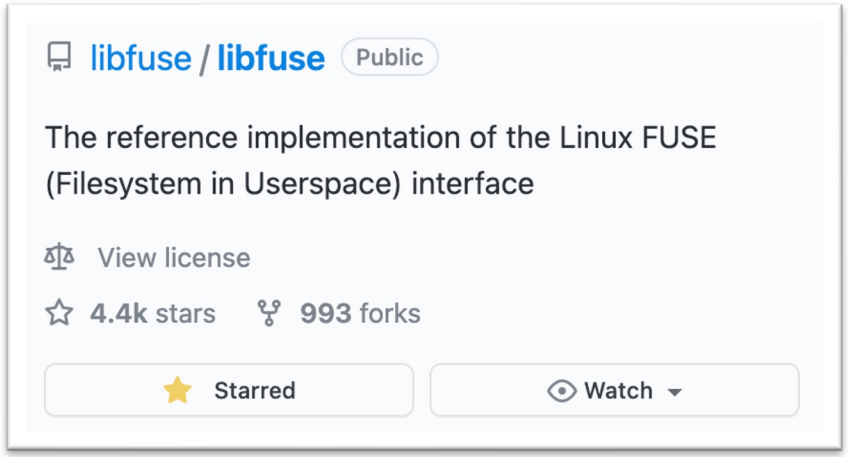
?

**Currently only available with the limited technical preview program of Nvidia BlueField.*

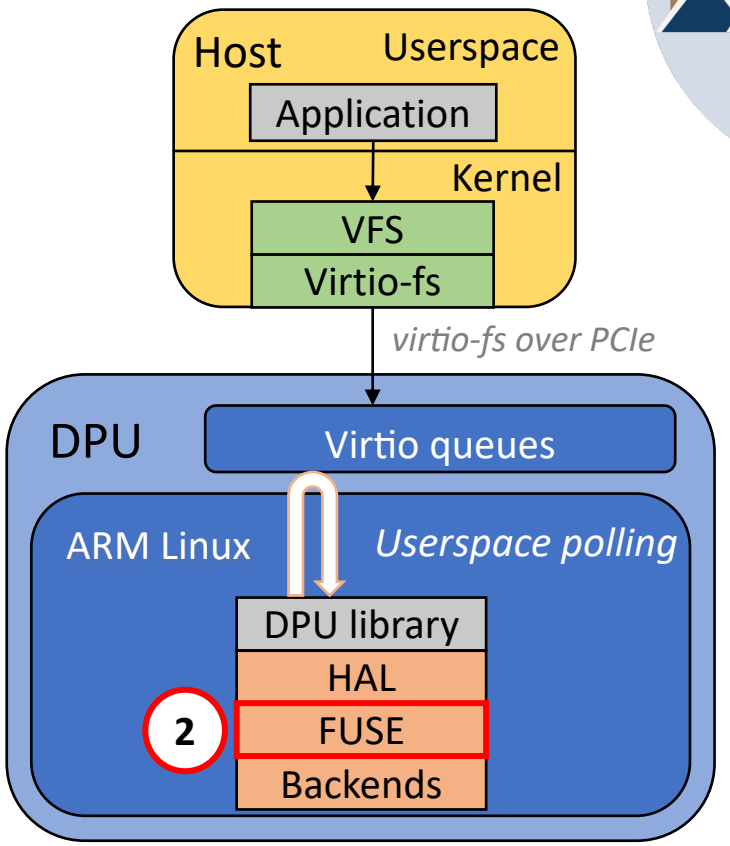
The **DPFS** framework: **D**PU-**P**owered **F**ile **S**ystems

Architecture:

1. Hardware Abstraction Layer
2. **FUSE API**
3. Several backends



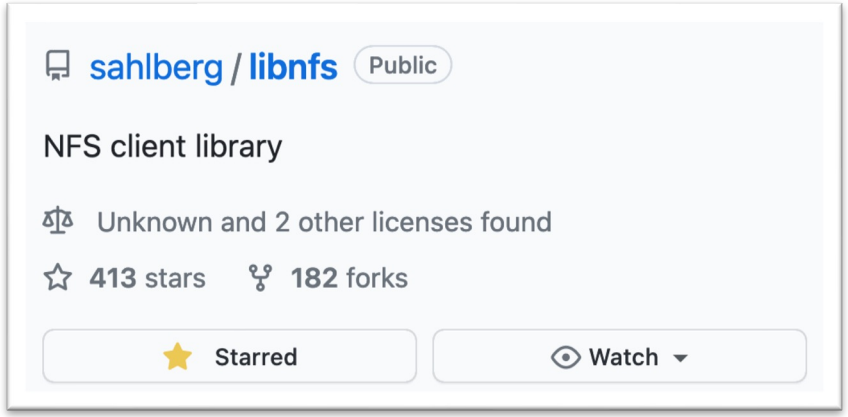
API ~equal, but no multithreading *yet*



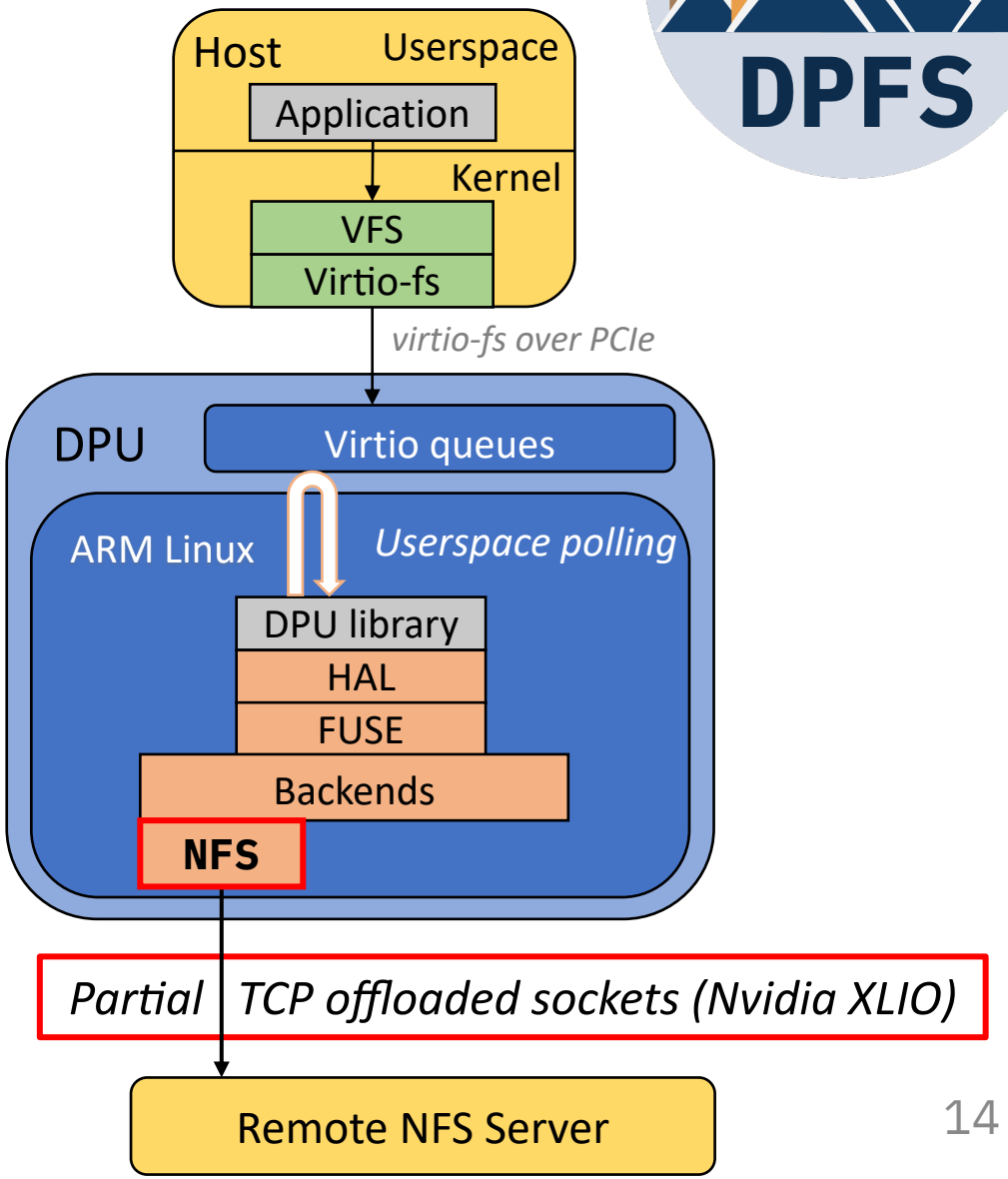
The **DPFS** framework: **D**PU-**P**owered **F**ile **S**ystems

Architecture:

1. Hardware Abstraction Layer
2. FUSE API
3a. **Several backends: NFS**



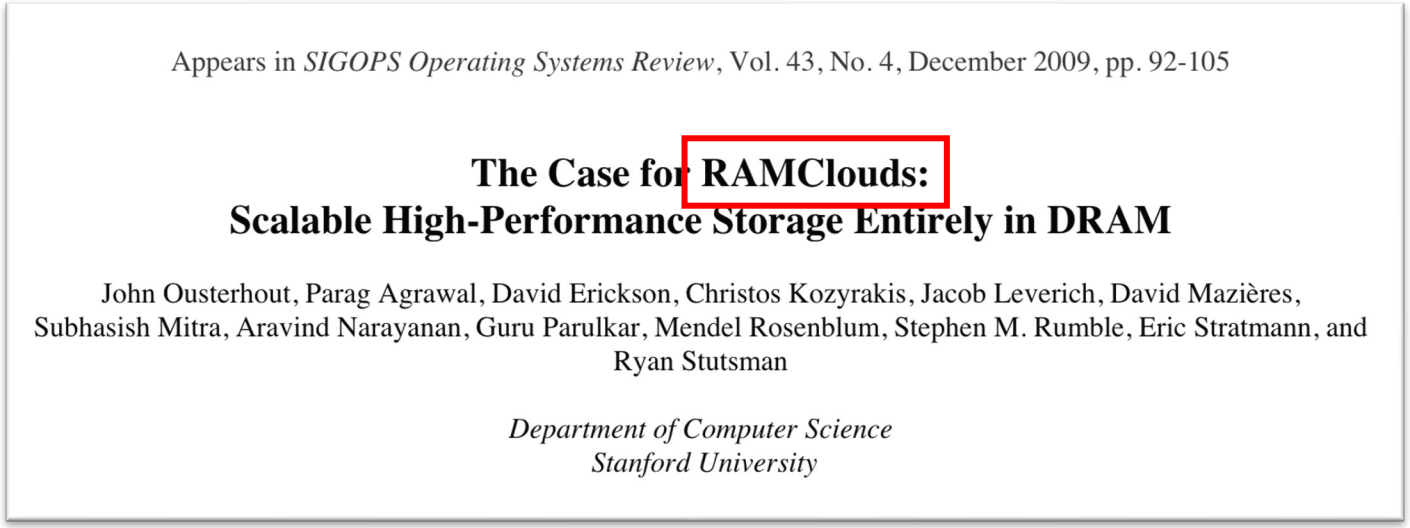
Userspace NFS v4.1



The **DPFS** framework: **D**PU-**P**owered **F**ile **S**ystems

Architecture:

1. Hardware Abstraction Layer
2. FUSE API
3b. **Several backends: NFS, KV**

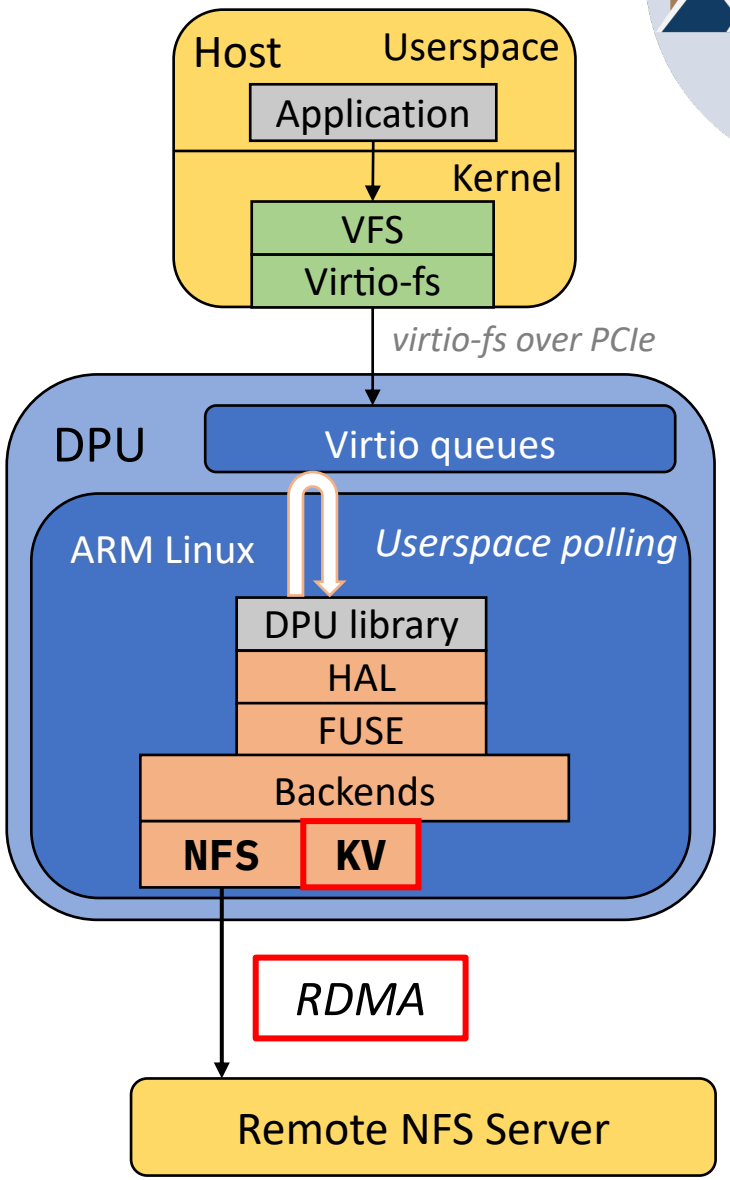


Appears in *SIGOPS Operating Systems Review*, Vol. 43, No. 4, December 2009, pp. 92-105

The Case for RAMClouds:
Scalable High-Performance Storage Entirely in DRAM

John Ousterhout, Parag Agrawal, David Erickson, Christos Kozyrakis, Jacob Leverich, David Mazières, Subhasish Mitra, Aravind Narayanan, Guru Parulkar, Mendel Rosenblum, Stephen M. Rumble, Eric Stratmann, and Ryan Stutsman

Department of Computer Science
Stanford University



Flat hierarchy

Optimized for 4k I/O and low latency

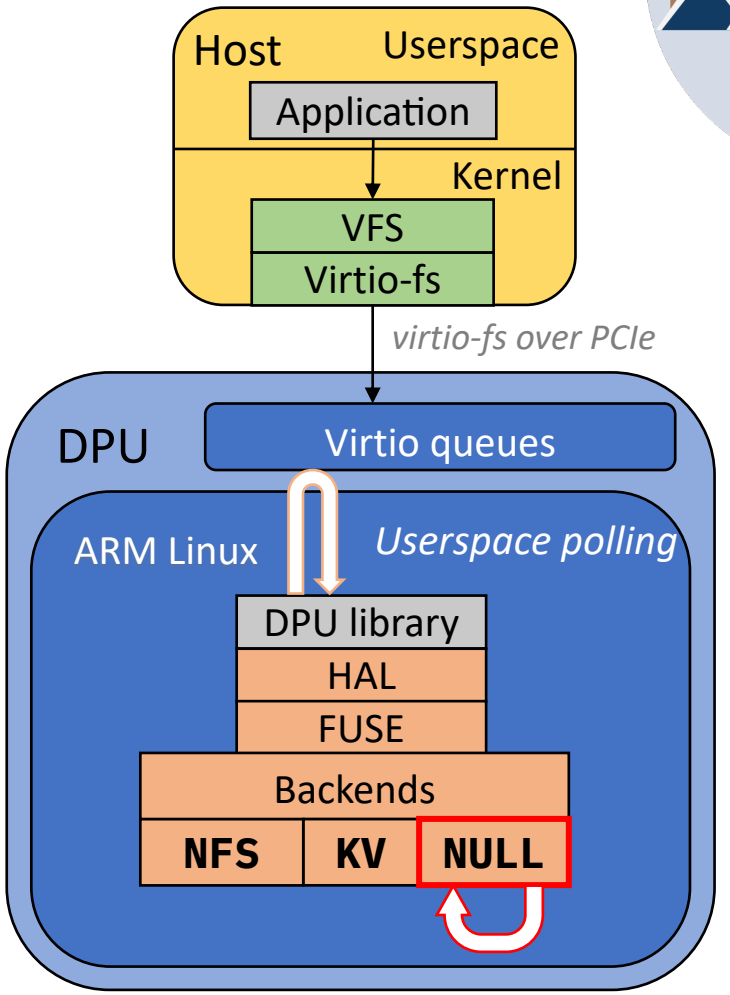
The **DPFS** framework: **D**PU-**P**owered **F**ile **S**ystems

Architecture:

1. Hardware Abstraction Layer
2. FUSE API
3c. **Several backends: NFS, KV, NULL**

Evaluates raw DPU performance:
latency and throughput

BlueField 2 vs BlueField 3 (soon)



Instantly returns any operation

Experimental evaluation

- Q1: Baseline DPU performance (DPFS-NULL)
- Q2: Throughput of DPFS-NFS (compared to Host NFS)
- Q3: Latency improvements with specialization (DPFS-NFS & -KV)
- Q4: Host CPU overhead analysis

Experimental setup

Host setup:

- 2x Intel Xeon E5-2630 v3, 2.2GHz, 8cores/socket
- 128GiB DDR4 1600
- Clean Ubuntu 22.04 (Linux 6.2) and fio 3.28
- NFS with optimized settings per Google Cloud (does more caching than DPFS)

DPU:

- Nvidia BlueField-2
- 8x A72 ARM cores (running Ubuntu 20.04 Linux)
- 16GB single-channel DDR4
- 100Gb/s ConnectX-6 network interface
- Exposes a single virtio-fs device to a single bare metal host

Q1: Baseline DPU performance (DPFS-NULL)

DPU setup:

- 1024 queue depth on the DPU
- Single core

Max TP = 7GB/s read and 5GB/s write

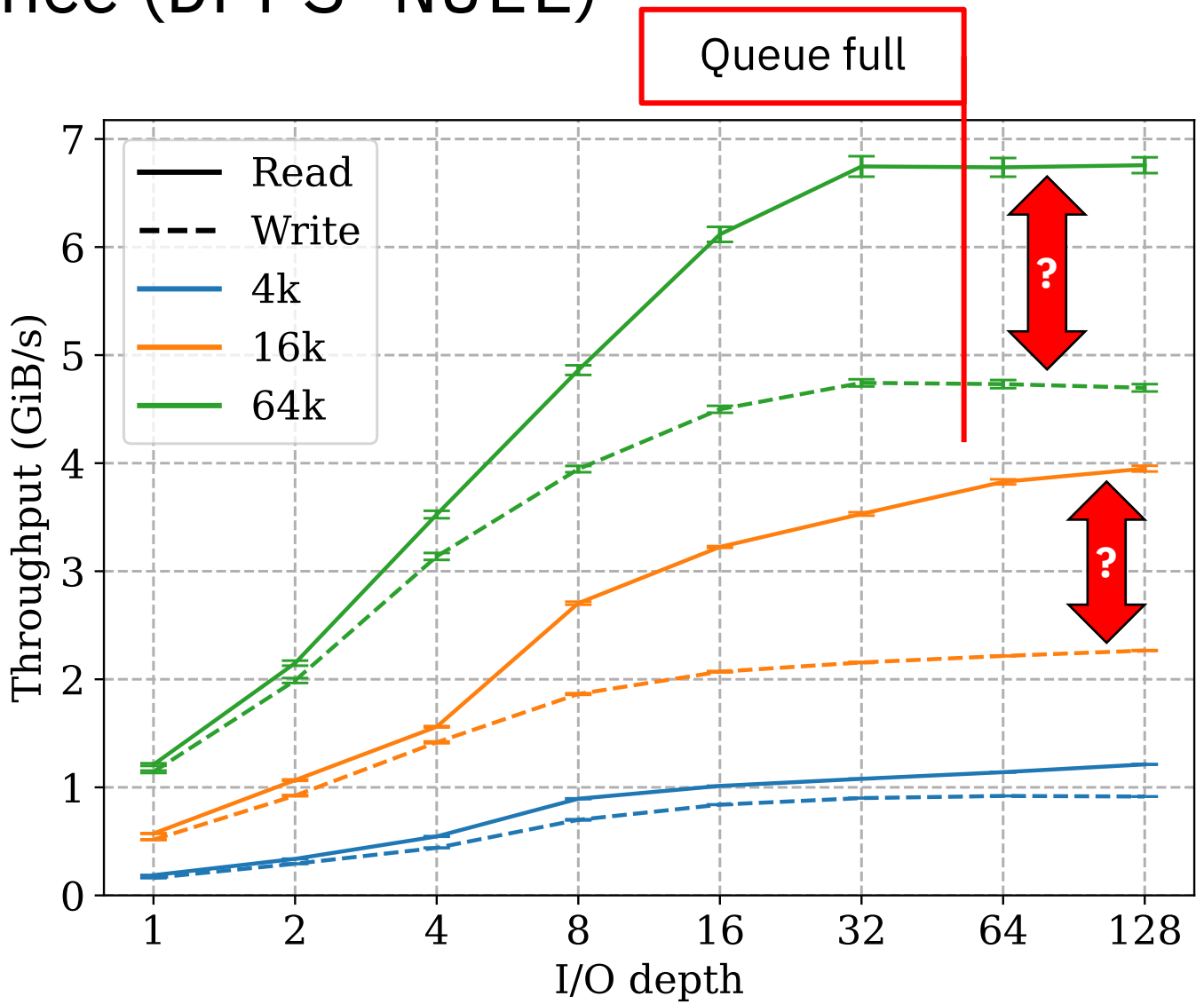
Large block sizes preferred

Read latency = 38.6µs

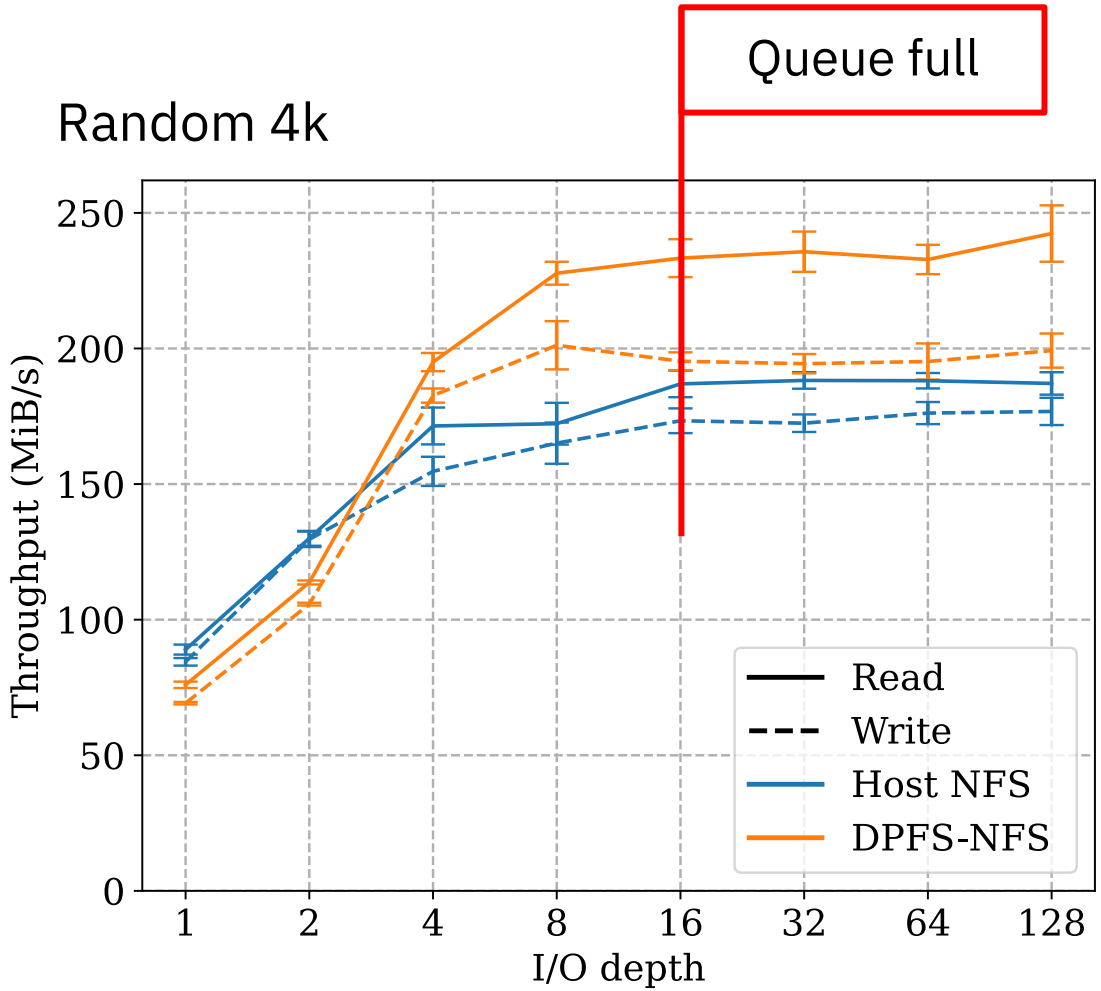
Write latency = 43.3µs

~40µs

Queue full and slow Arm A72 core fully saturated



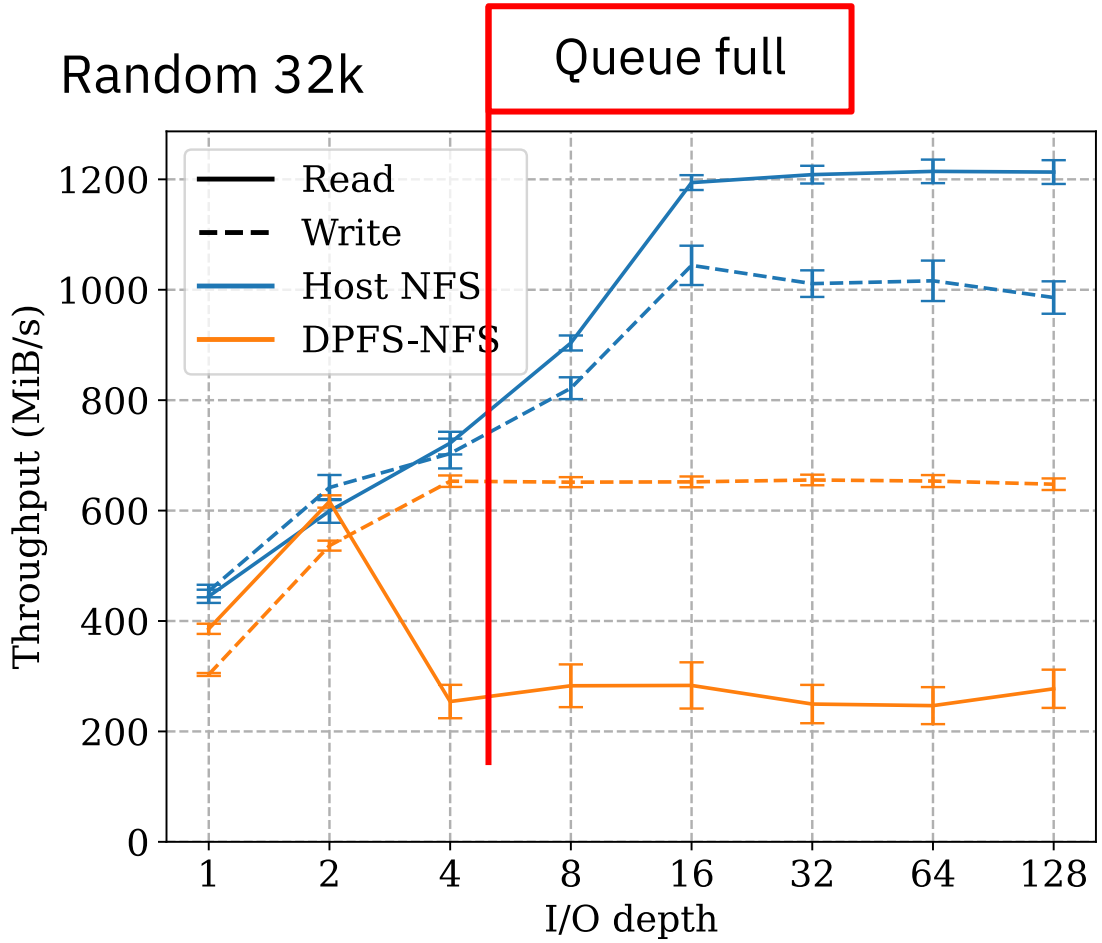
Q2: Throughput of DPFS-NFS



Bottleneck = TCP NFS I/O

DPU setup:

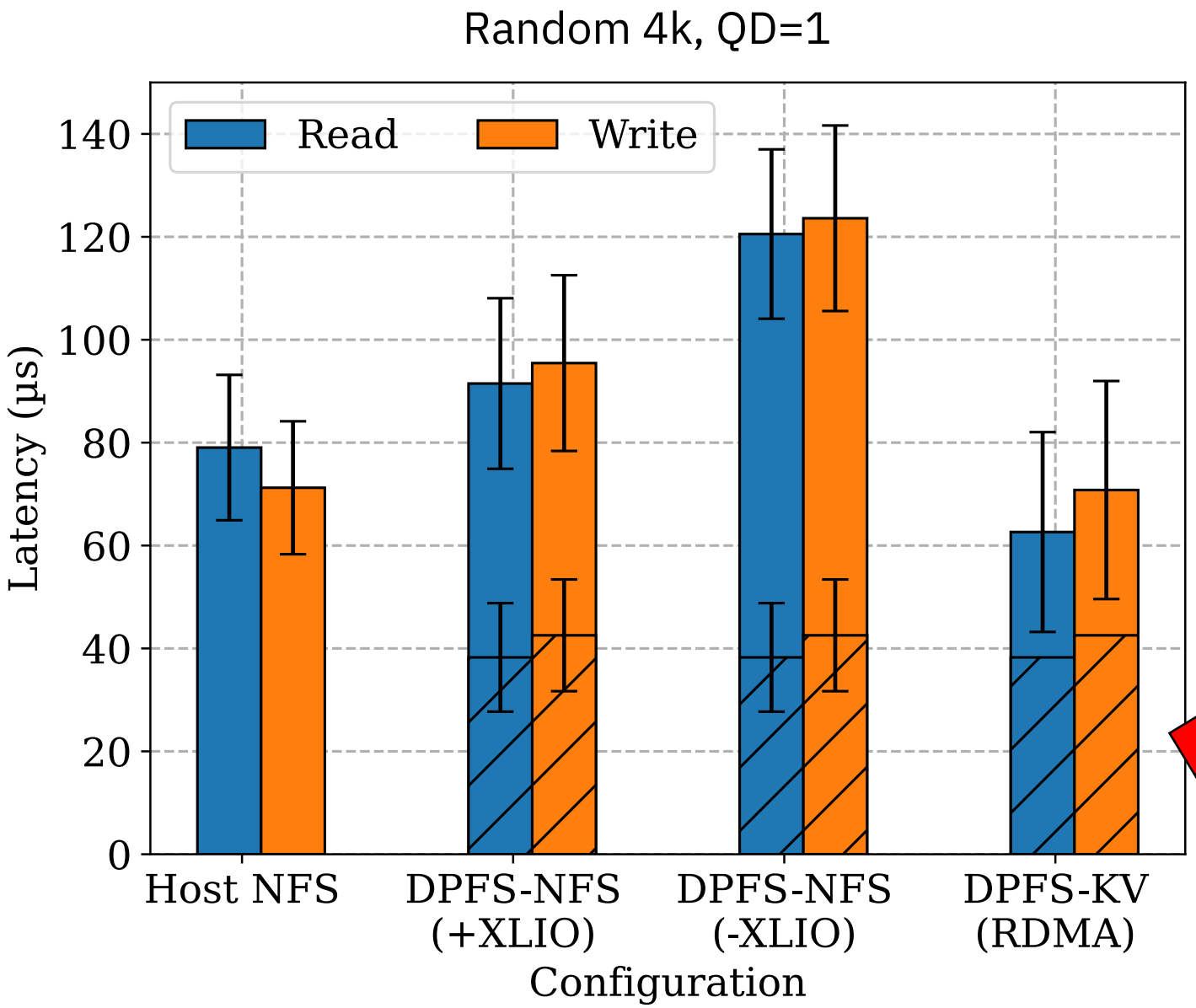
- 64 queue depth on the DPU (XLIO constrained)
- Single core (+ one core polling NFS completions)



Bottleneck = Limited queue depth (XLIO)

XLIO Read path *bad* with large BS & QD>=4

Q3: Latency improvements with specialization



Hardware specialization is key (e.g. TCP offloading or RDMA)

Baseline DPFS-NULL latency

Q4: Host CPU overhead analysis

Hypothesis:

Virtio-fs much lighter than NFS, so we expect big CPU savings.
(13k LoC vs 181k LoC)

Test setup:

- System-wide (kernel only) performance counters to account for TX and RX
- Take a 300s baseline, then perform a 300s stress test. Subtract the baseline from the stress test to only leave the instructions used for I/O.

	NFS	DPFS-NFS	+/-
Instructions/op	88,453	32,907	-62.80%
IPC	0.57	0.94	+64.21%
Branch miss rate	2.02	1.06	-47.42%
L1 dCache miss rate	8.82	3.82	-56.65%
dTLB miss rate	0.14	0.15	+7.14%
Savings in CPU cycles/op		**4.4×**	

Conclusions

- DPFS: a DPU-Powered File System Virtualization framework

Efficiency			Management			Security	
Performance	Overhead	Multi-tenancy	Support all tenants	Client transparency	Operator control	Attack surface	Network isolation

- Holistic solution for today's cloud file system needs.
- Up to 7GB/s throughput and base latency of ~40µs with DPU (single core)
- 4.4x host cycle savings and similar performance to host NFS
- Two hardware specialized backends: NFS and KV

Future work for DPFS

- Performance optimizations
  - *io_uring* file system backend for DPFS (DPU-local mirror)
  - Thread pooling in DPFS*
  - Multi-queue support in virtio-fs and DPFS*
- New RPC-based Virtio-fs backend
- Multi-tenancy performance evaluation
- Transition to faster DPUs (i.e. Nvidia BlueField-3)

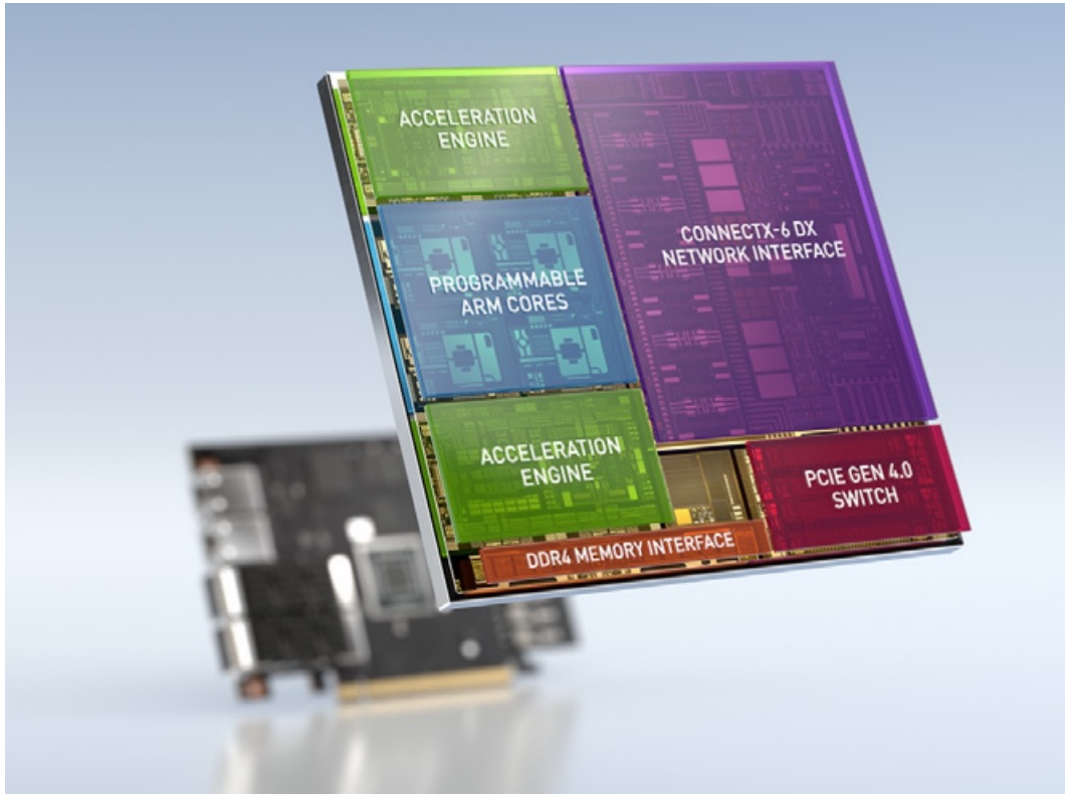
Thank you

Info and contact about the project at:
github.com/IBM/DPFS

Backup/extra

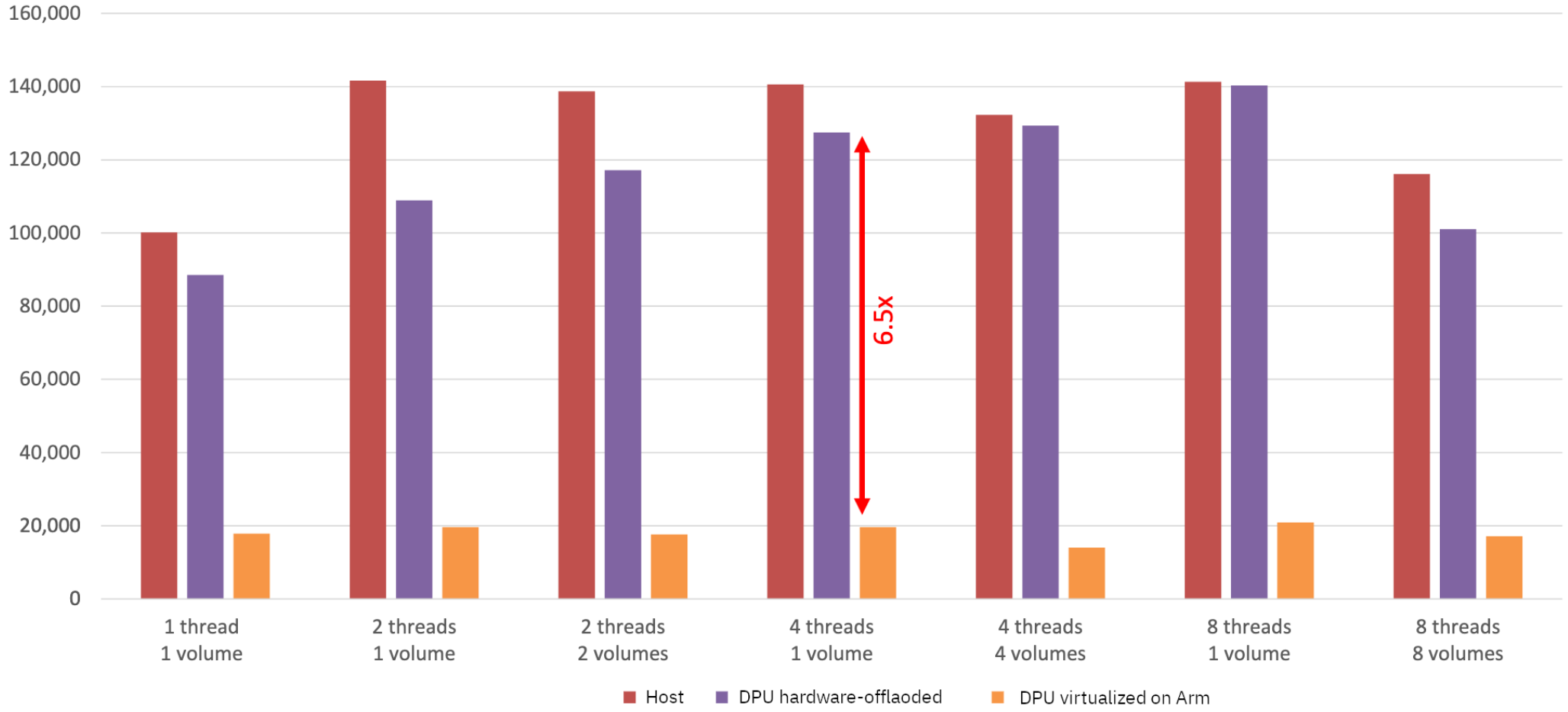
Nvidia BlueField-2 DPU

- 8x A72 ARM cores (running Ubuntu 20.04 Linux)
- 16GB single-channel DDR4
- 2x 100Gb/s ConnectX-6 network interface
- Hardware acceleration engines for:
  - Security
  - Networking
  - Storage
- Attached to host CPU over PCIe Gen 4.0
- Collaboration with Nvidia for limited technical feature preview



[Image source: nvidia.com]

OFA ‘21: DPU-offloaded Block storage

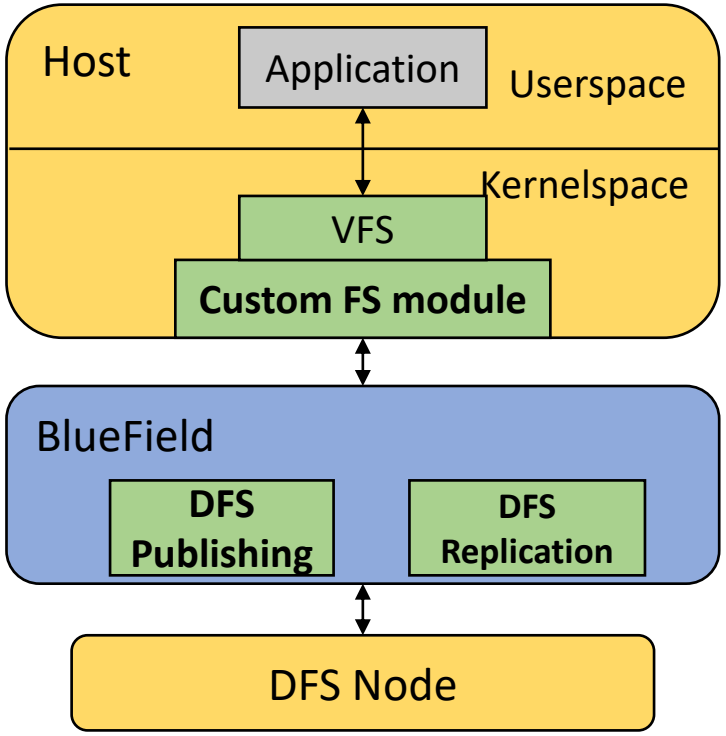


READ IOPS QD128@16KIB

2021 OFA Virtual Workshop: *How to efficiently provide software-defined storage using SmartNICs*
Jonas Pfefferle, Nikolas Ioannou, Jose Castanos, Bernard Metzler
IBM Research Zurich

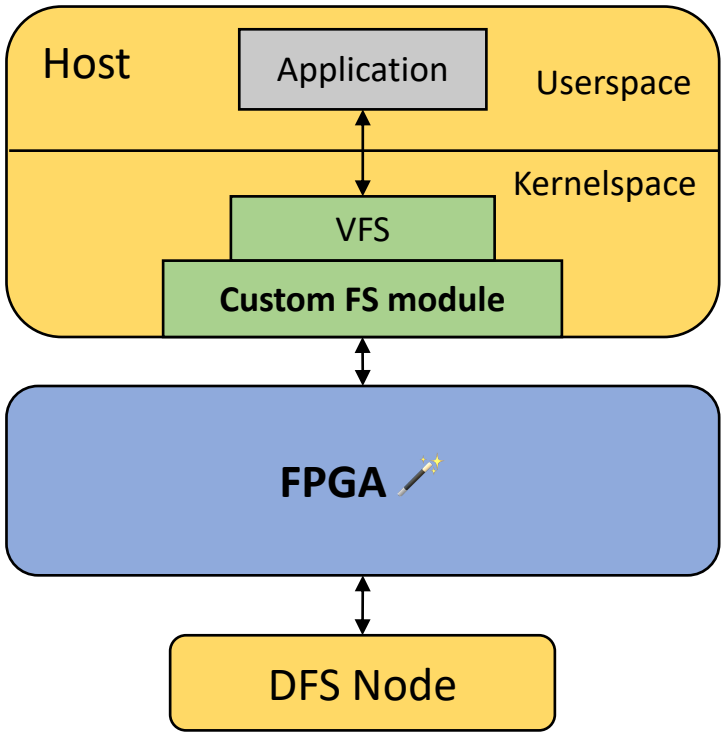
Related DPU File System research

LineFS



[Kim, Jongyul, et al. "LineFS: Efficient SmartNIC offload of a distributed file system with pipeline parallelism." *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles*. 2021.]

Fisc (Alibaba)



[Li, Qiang, et al. "Fisc: a large-scale cloud-native-oriented file system." *21st USENIX Conference on File and Storage Technologies (FAST 23)*. 2023.]

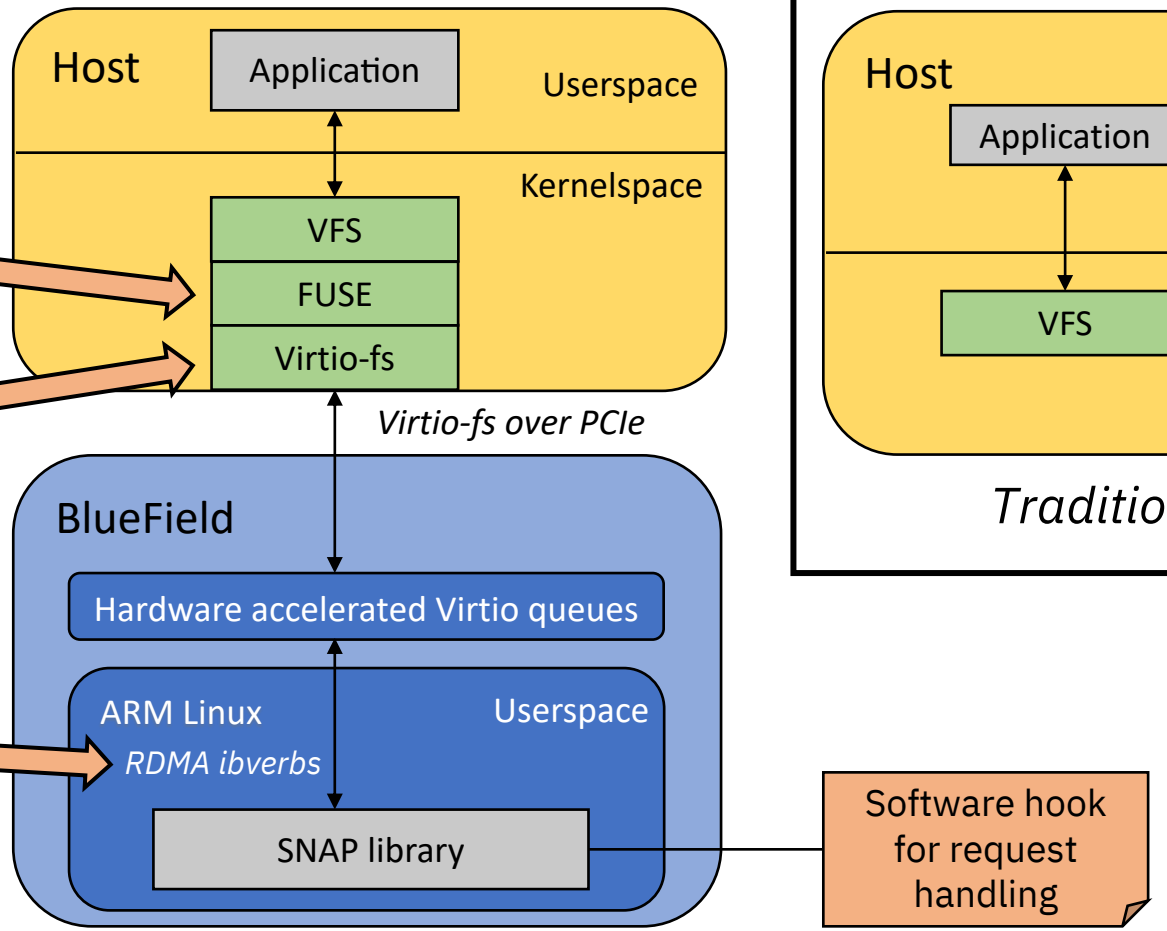
DPFS does full FS offload on CPU-based DPUs without custom kernel modules

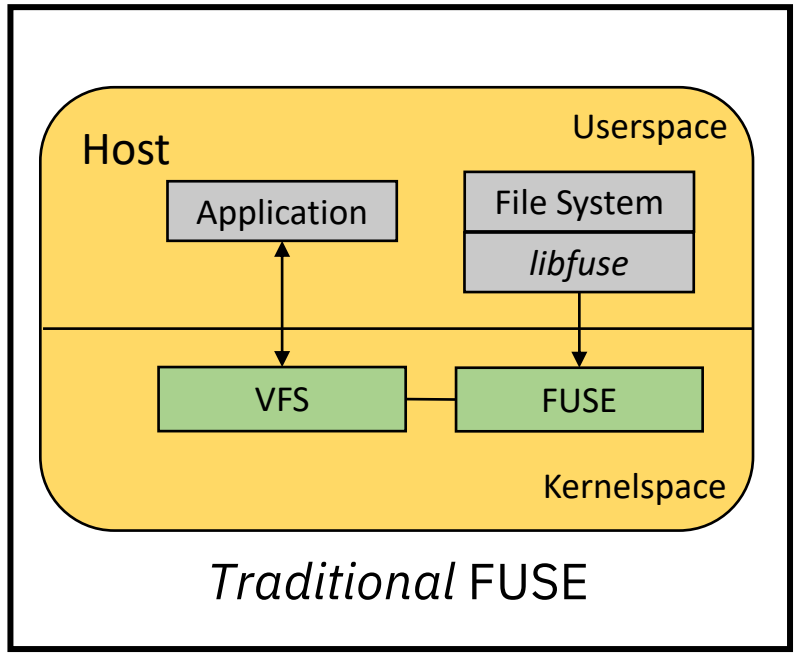
Virtio-fs on the Nvidia BlueField-2*

Linux VFS encoded into FUSE ABI

Sends FUSE over PCIe

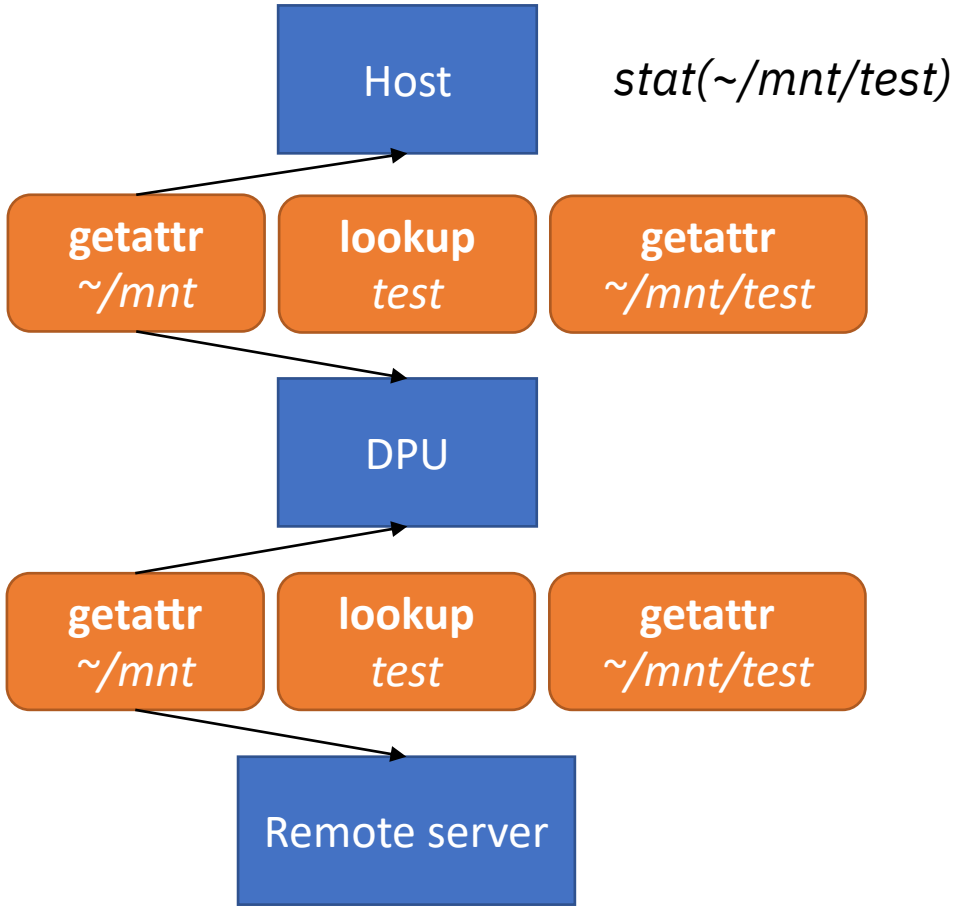
Firmware exposes virtio-fs via RDMA interface



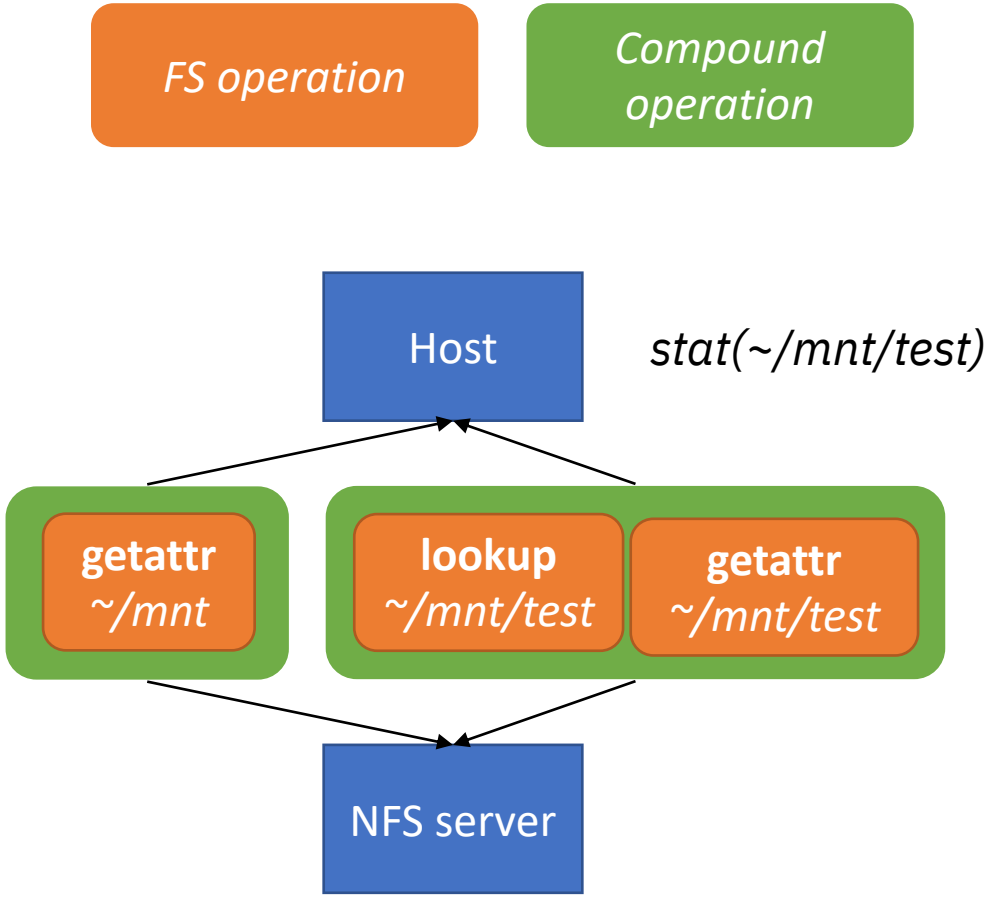


**Currently only available with the limited technical preview program of Nvidia BlueField.*

Metadata performance dissection



Single *getattr* from host: **87 usec**
Full stat (*getattr*, *lookup*, *getattr*): **273 usec**



Full stat: **212 usec**

Scoring breakdown (all subcategories for the options)

Option 1: Traditional DFS

Efficiency			Management			Security	
Performance	Overhead	Multi-tenancy	Support all cloud clients	Client transparency	Operator control	Attack surface	Network isolation

Option 2: NFS Gateway

Efficiency			Management			Security	
Performance	Overhead	Multi-tenancy	Support all cloud clients	Client transparency	Operator control	Attack surface	Network isolation

Option 3: File System on top of Remote Block

Efficiency			Management			Security	
Performance	Overhead	Multi-tenancy	Support all cloud clients	Client transparency	Operator control	Attack surface	Network isolation

Performance summary

DPFS with the BlueField-2 performance:

- The DPU incurs a base 40μs latency overhead
- On par in simple R/W workloads
- DPFS-NFS worse in larger block size workloads than Host NFS
  - Because of framework limitations and puny Arm cores
- Lower latency than host NFS with specialization in the file system
- Bottleneck on metadata operation performance
  - Because of FUSE lack of compounding
- Smaller performance gap between host and DPU-virtualized than with block storage
- **4.4x** savings in host CPU cycles/op compared to NFS

Future is looking bright with next generation DPUs like BlueField-3!