Here, There and Everywhere: The Past, the Present and the Future of Local Storage in Cloud

Leping Yang, Yanbo Zhou†, Gong Zeng†, Li Zhang†, Saisai Zhang†, Ruilin Wu†, Chaoyang Sun†, Shiyi Luo†, Wenrui Li†, Keqiang Niu†, Xiaolu Zhang†, Junping Wu†, Jiaji Zhu†, Jiesheng Wu†, Mariusz Barczak*, Wayne Gao*, Ruiming Lu, Erci Xu and Guangtao Xue

Shanghai Jiao Tong University, †Alibaba Group, *Solidigm

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Alibaba

SOLIDIGM™

Outline

- **Background**
- **The Past**
- **The Present**
- **The Future**
- **Conclusion**

Local Storage Model in the Cloud

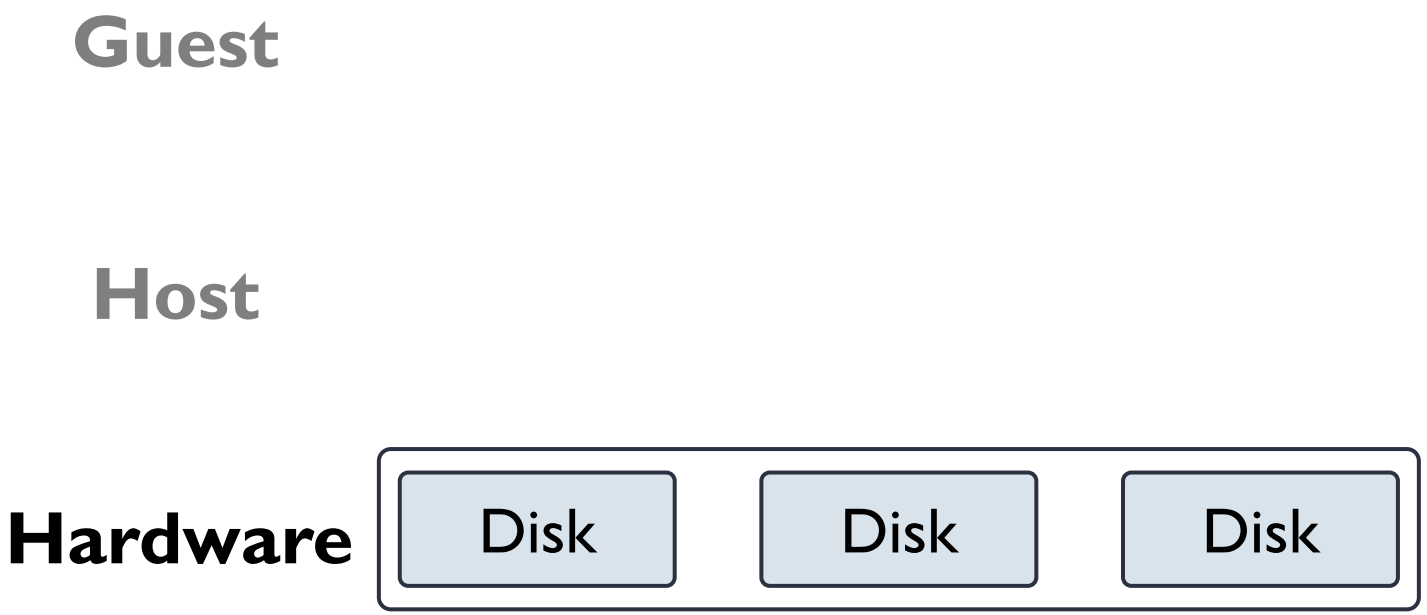
Guest

Host

Hardware

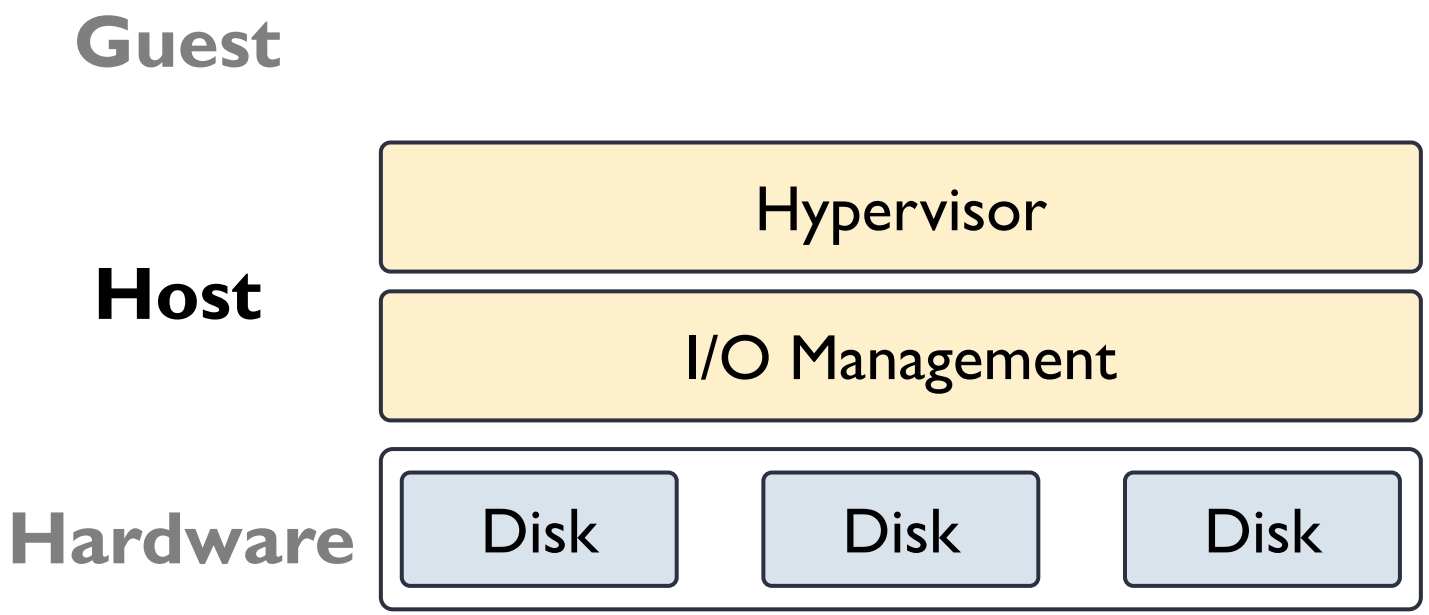
Local Storage Model in the Cloud

- Hardware: directly attached



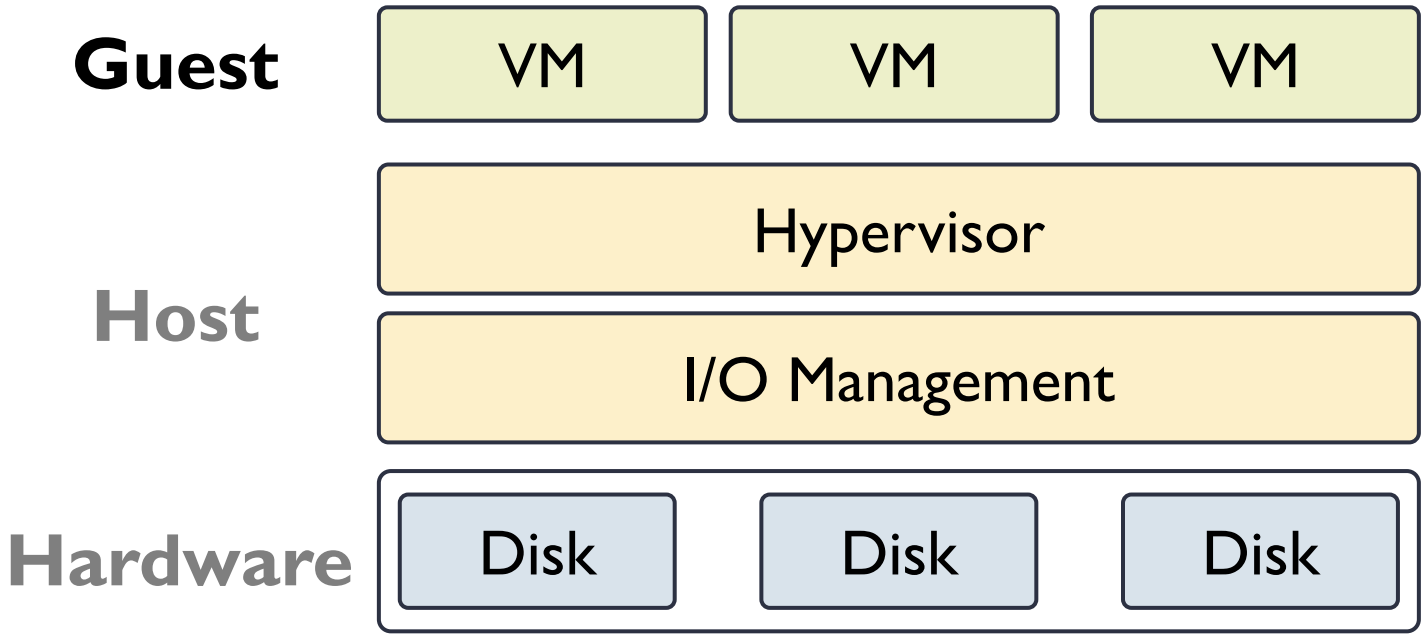
Local Storage Model in the Cloud

- Hardware: directly attached
- Host: manage and virtualize physical disks



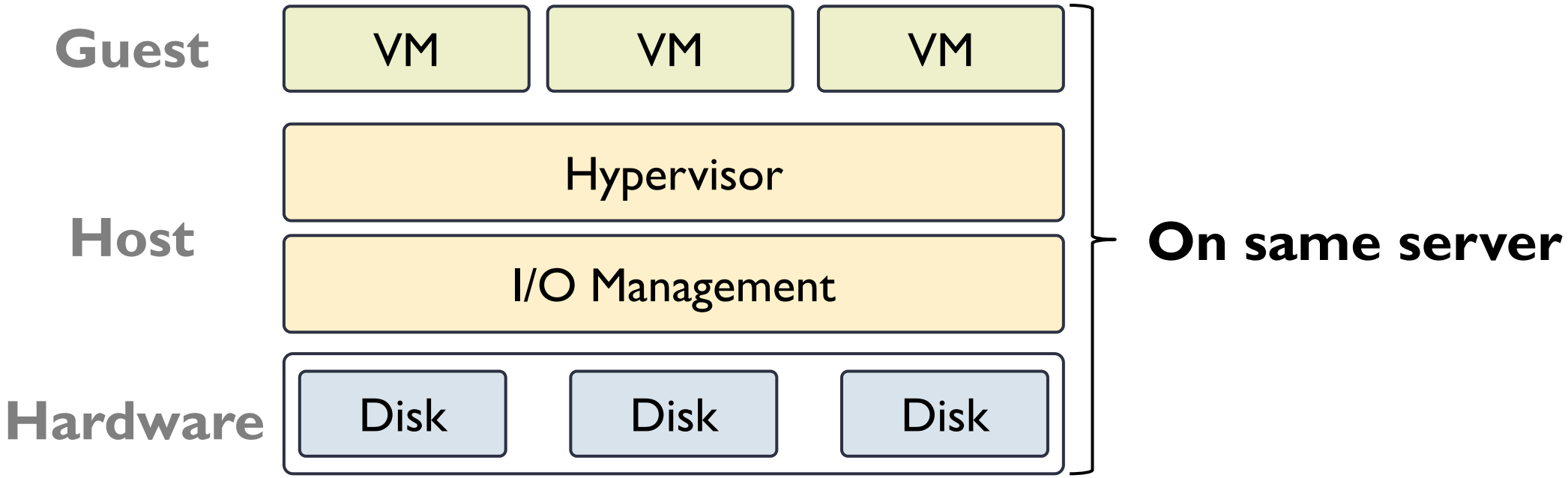
Local Storage Model in the Cloud

- Hardware: directly attached
- Host: manage and virtualize physical disks
- Guest: subscribe one/multiple virtual disks



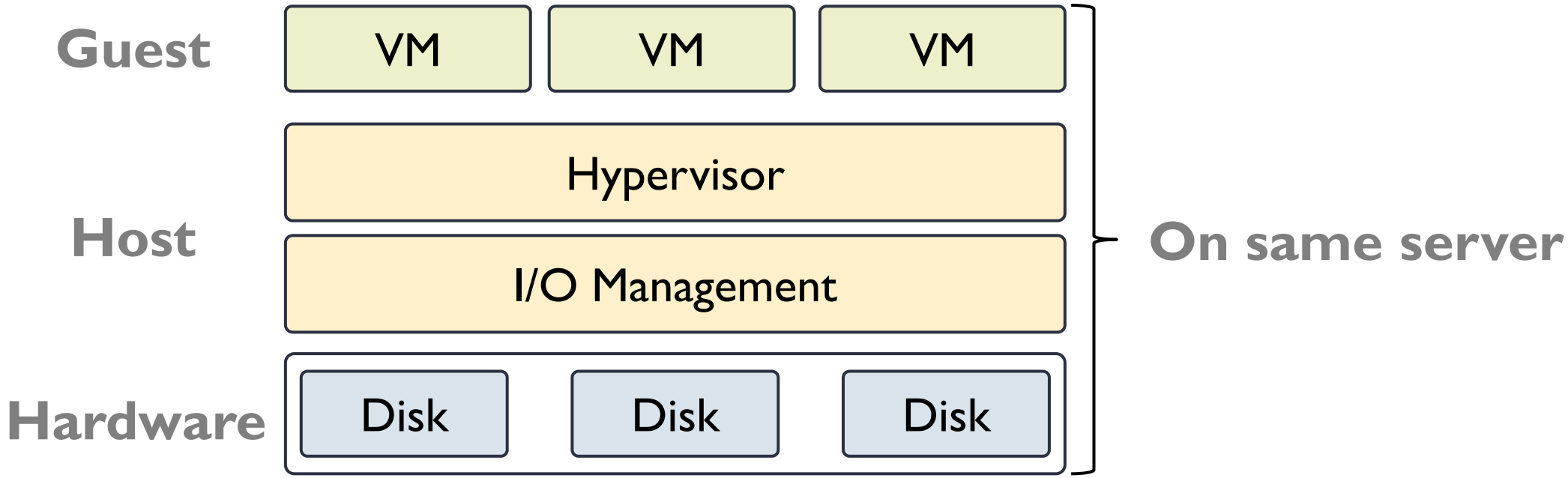
Local Storage Model in the Cloud

- Hardware: directly attached
- Host: manage and virtualize physical disks
- Guest: subscribe one/multiple virtual disks



Local Storage Model in the Cloud

- Hardware: directly attached
- Host: manage and virtualize physical disks
- Guest: subscribe one/multiple virtual disks



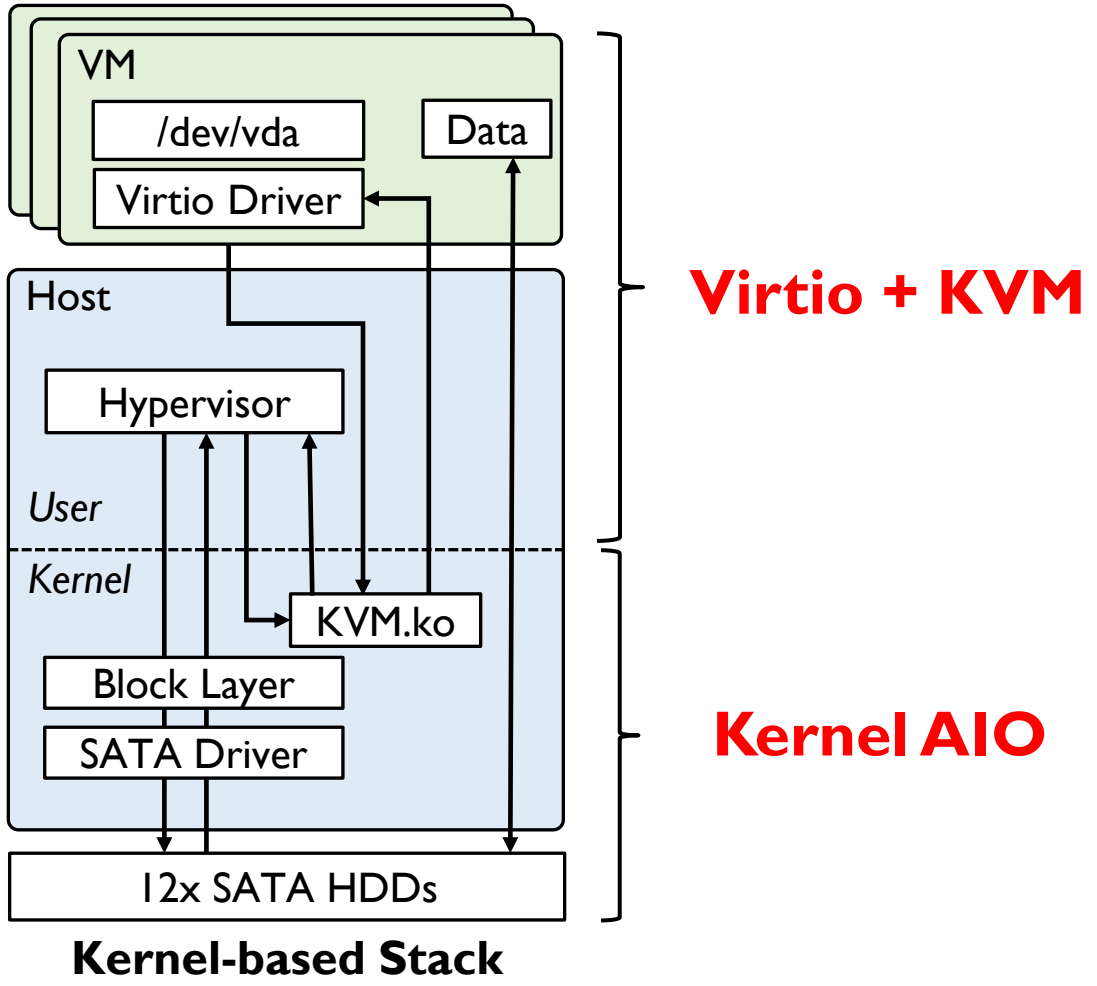
*We focus on the design and deployment of **NVMe SSD-based** local storage*

Outline

- Background
- **The Past**
- The Present
- The Future
- Conclusion

Espresso: From Kernel to SPDK

Kernel-based stack is outdated

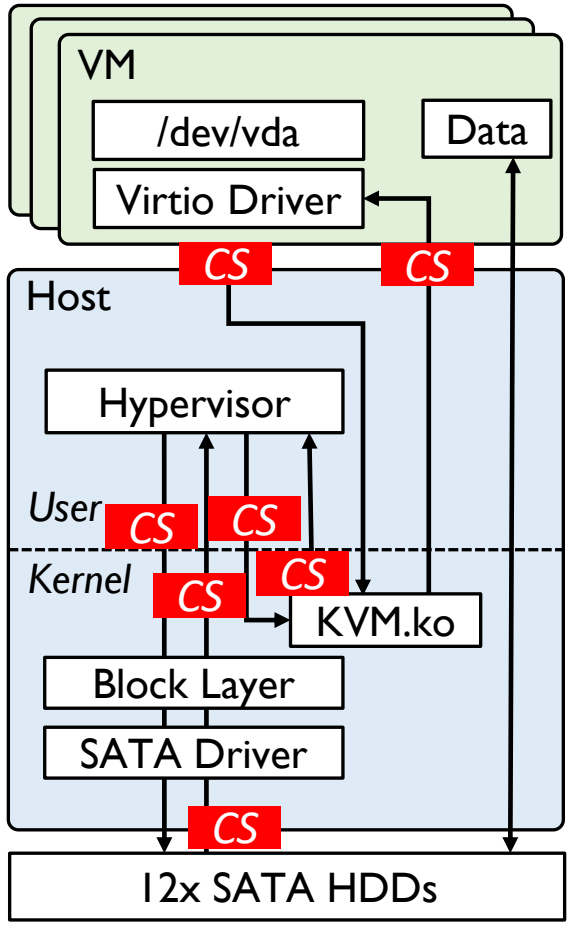


Kernel-based Stack

Espresso: From Kernel to SPDK

Kernel-based stack is outdated

- Context switches (CS)

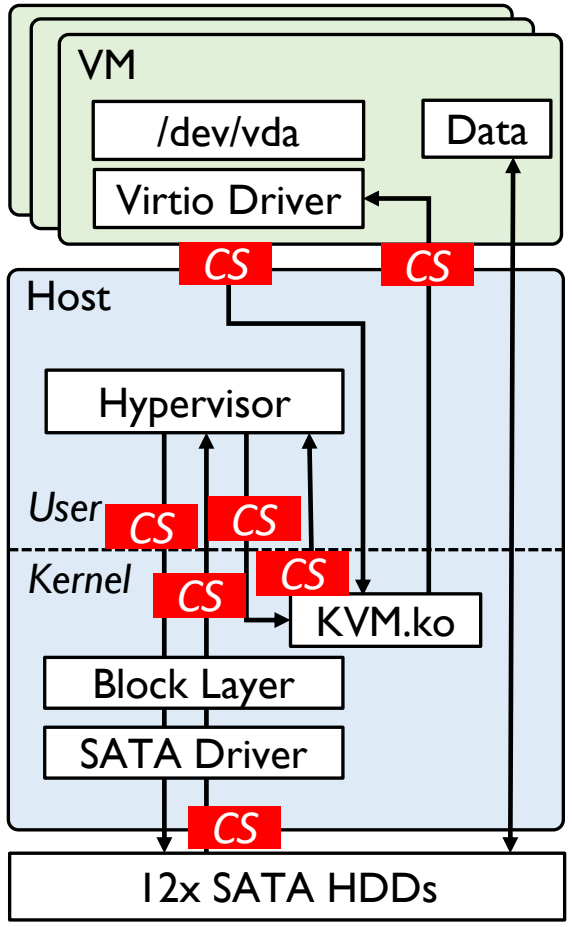


Kernel-based Stack

Espresso: From Kernel to SPDK

Kernel-based stack is outdated

- Context switches (CS)
- High latency & Low CPU-efficiency



Kernel-based Stack

Espresso: From Kernel to SPDK

Kernel-based stack is outdated

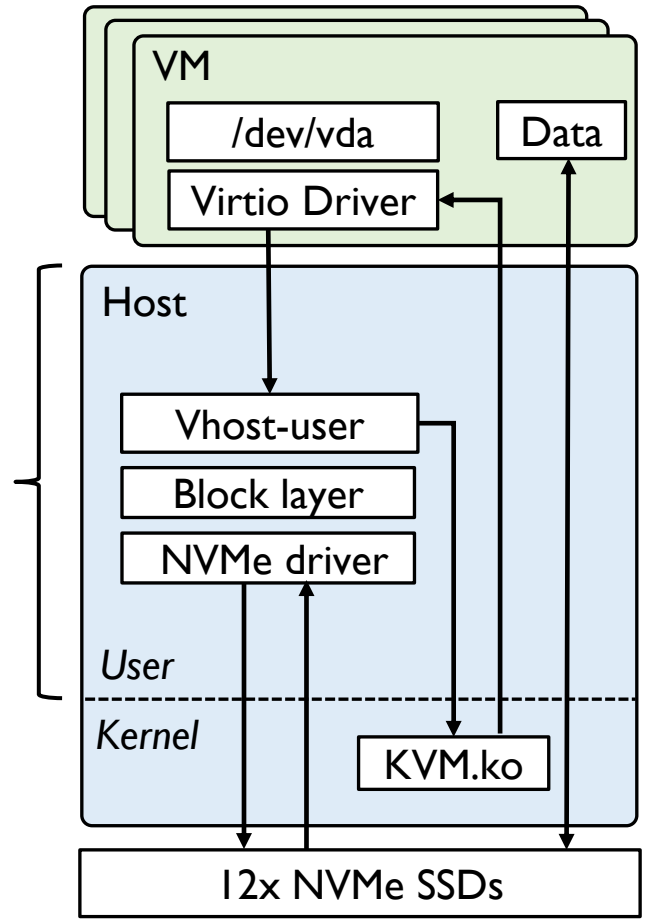
- Context switches (CS)
- High latency & Low CPU-efficiency

SPDK-based stack best fits

SPDK

Vhost-user

+ NVMe



SPDK-based Stack

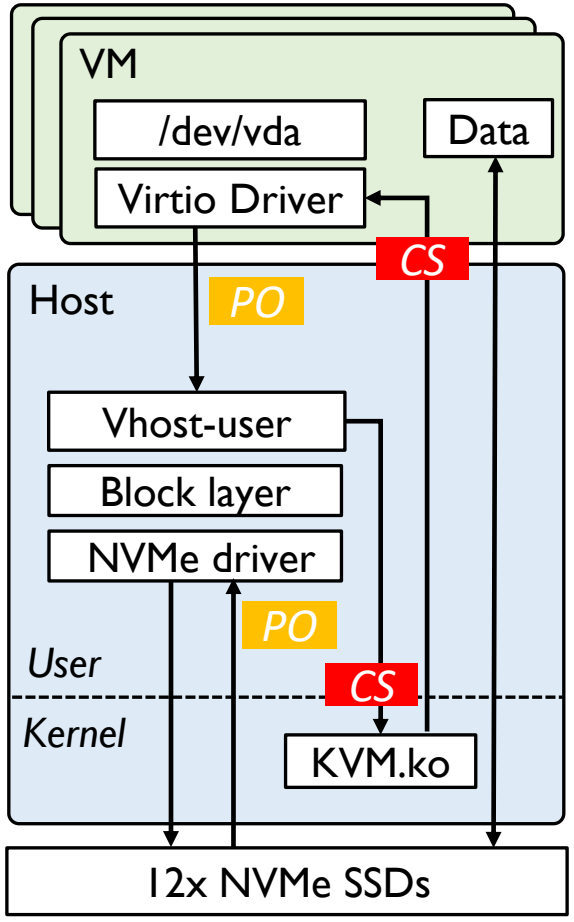
Espresso: From Kernel to SPDK

Kernel-based stack is outdated

- Context switches (CS)
- High latency & Low CPU-efficiency

SPDK-based stack best fits

- Userspace polling (PO)



SPDK-based Stack

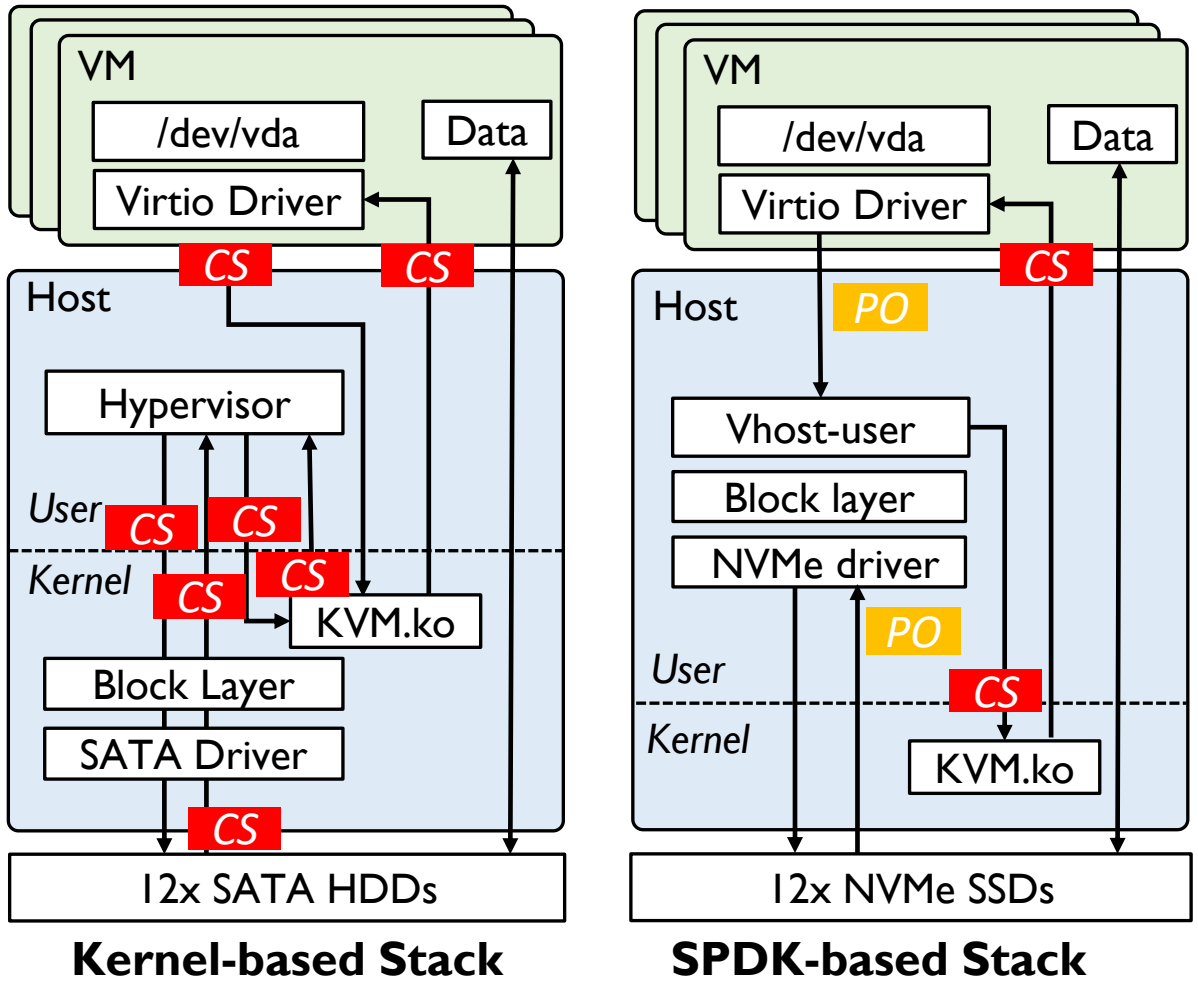
Espresso: From Kernel to SPDK

Kernel-based stack is outdated

- Context switches (CS)
- High latency & Low CPU-efficiency

SPDK-based stack best fits

- Userspace polling (PO)
- Low latency & High CPU-efficiency



Lessons from Espresso Deployment

SPDK shows limitations especially for future evolution

No bare-metal support

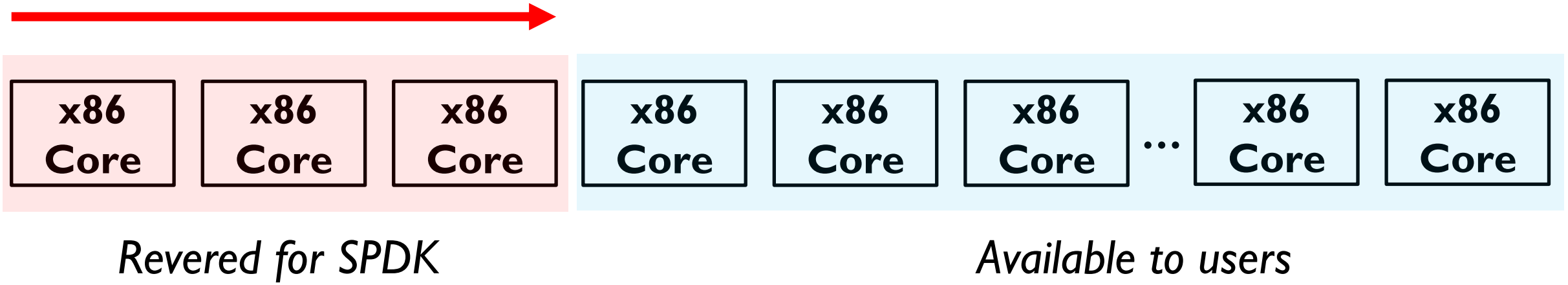
- Dedicating CPU cores for I/O

Lessons from Espresso Deployment

SPDK shows limitations especially for future evolution

No bare-metal support

- Dedicating CPU cores for I/O



Lessons from Espresso Deployment

SPDK shows limitations especially for future evolution

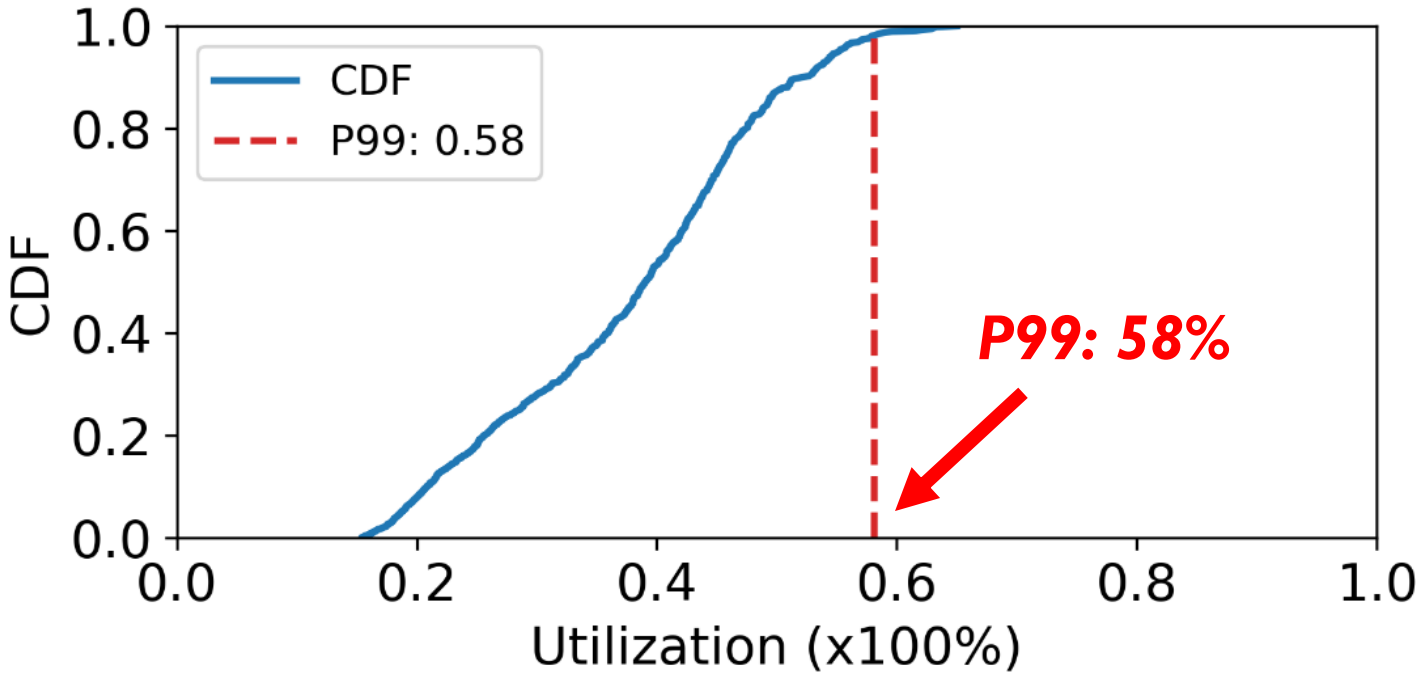
No bare-metal support

- Dedicating CPU cores for I/O

Not efficiently using CPU

- Busy polling

40%+ CPU cycles are wasted!



Effective CPU utilization

Lessons from Espresso Deployment

SPDK shows limitations especially for future evolution

No bare-metal support

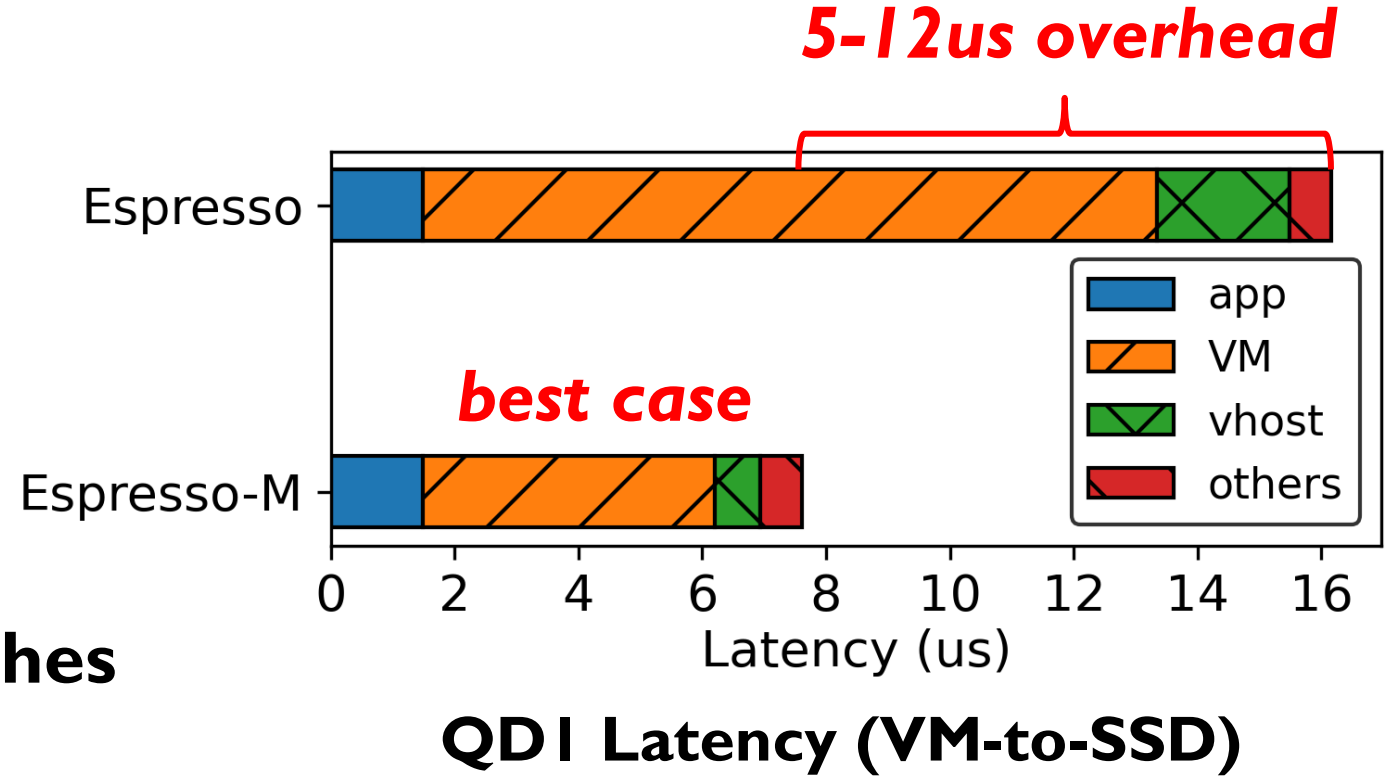
- Dedicating CPU cores for I/O

Not efficiently using CPU

- Busy polling

Not free from context switches

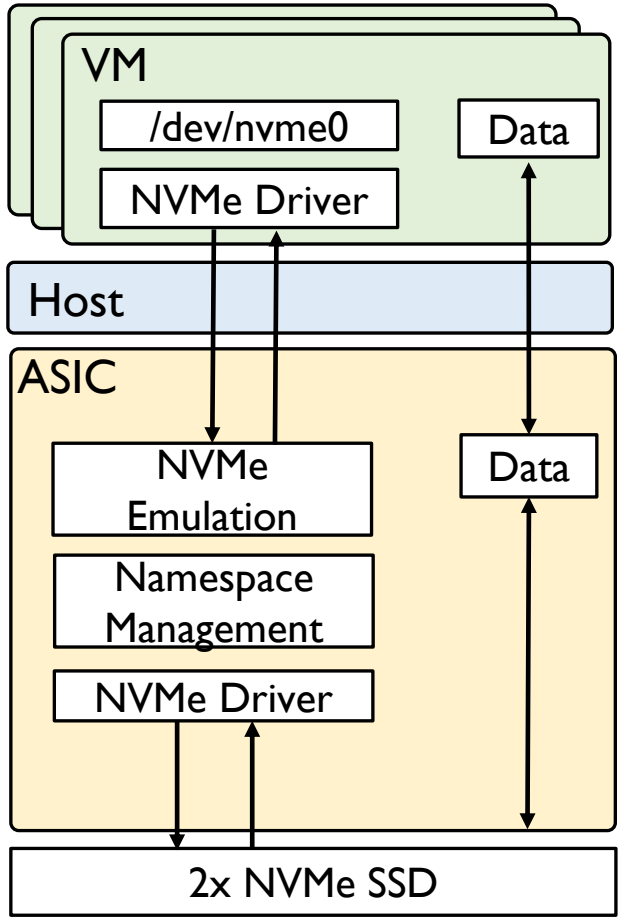
- Final step I/O to VMs



QD1 Latency (VM-to-SSD)

Doppio: From CPU to ASIC

ASIC-based hardware offloading

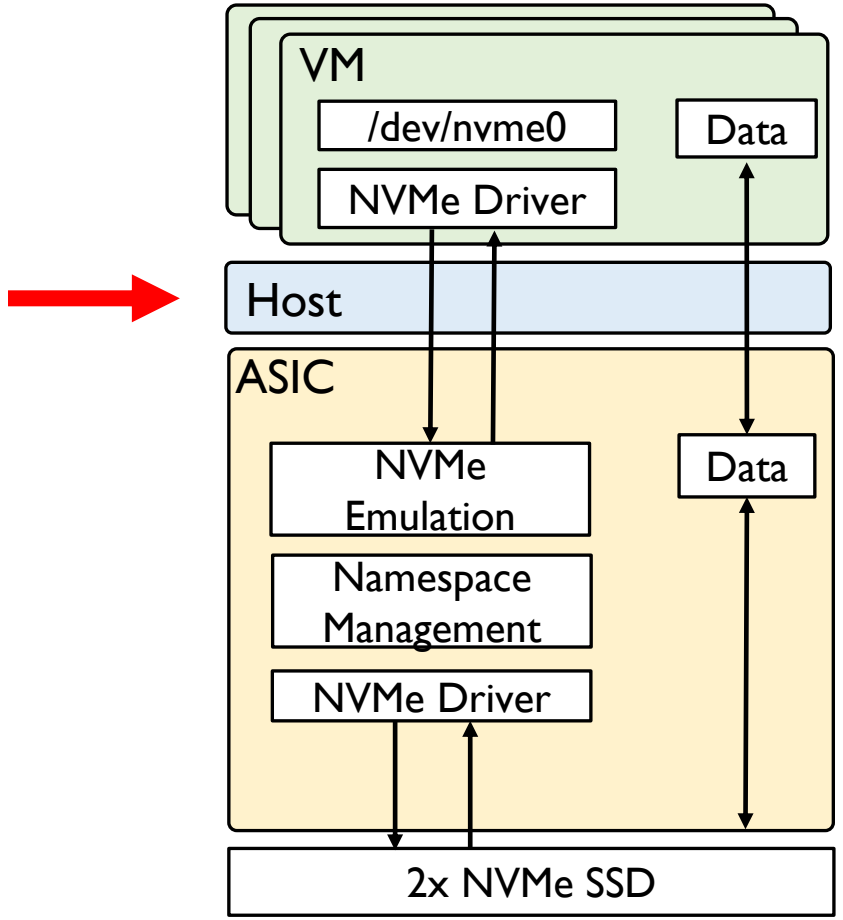


ASIC-based DPU

Doppio: From CPU to ASIC

ASIC-based hardware offloading

- Bare-metal support

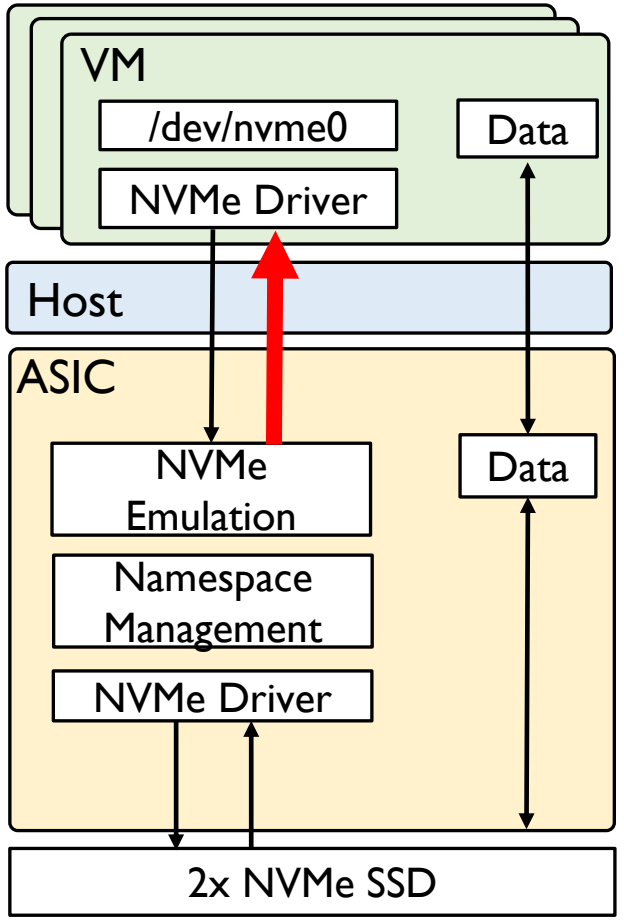


ASIC-based DPU

Doppio: From CPU to ASIC

ASIC-based hardware offloading

- Bare-metal support
- Free from host-side context switches

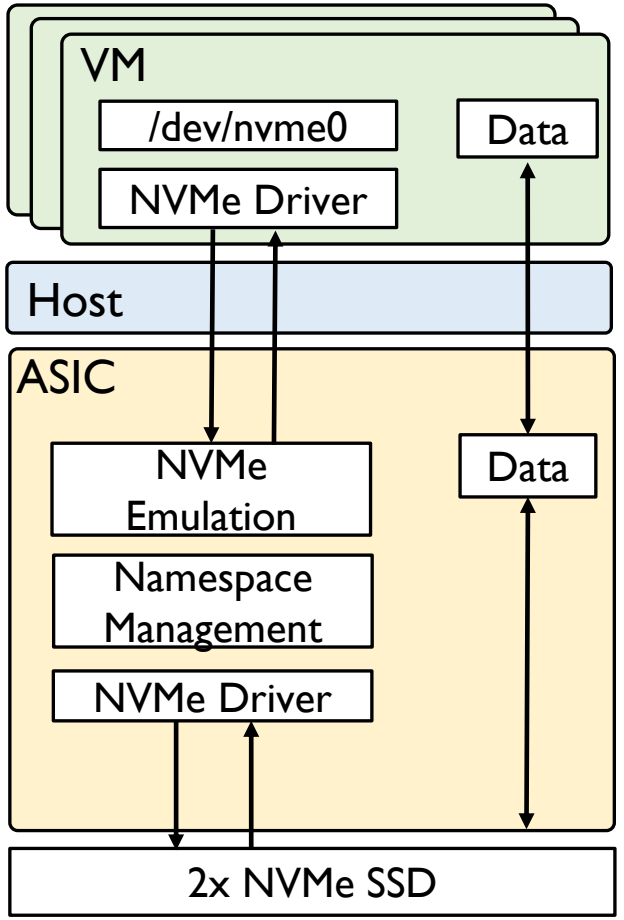


ASIC-based DPU

Doppio: From CPU to ASIC

ASIC-based hardware offloading

- Bare-metal support
- Free from host-side context switches
- Low cost and power consumption



ASIC-based DPU

Lessons from Doppio Deployment

ASIC-based offloading is not the cure-all

Hard to keep up with rapid SSD evolution

- ASIC takes longer time to manufacture

	2017		2020		2023	
	Read	Write	Read	Write	Read	Write
Vendor-1	490K	38K	920K	135K	1.1M	390K
Vendor-2	430K	40K	1M	180K	1.5M	250K

SSD performance (IOPS) evolution

Lessons from Doppio Deployment

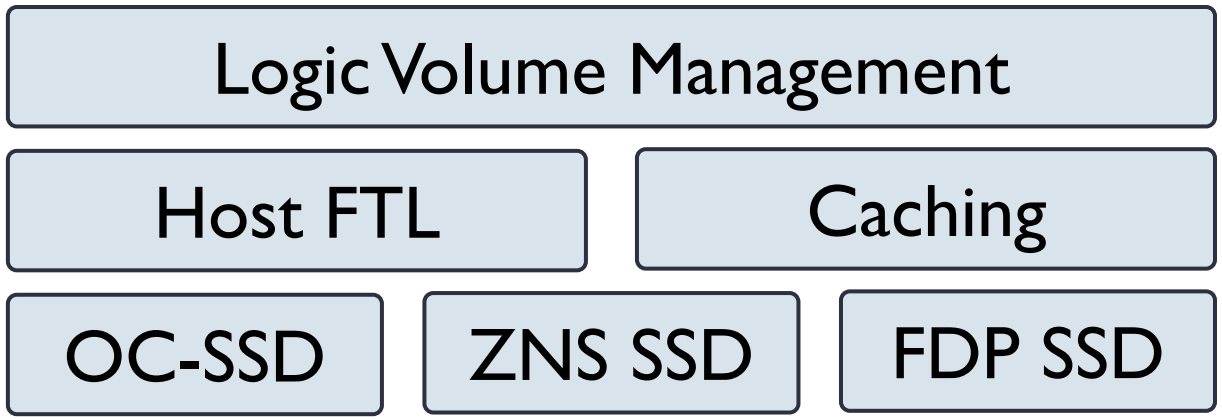
ASIC-based offloading is not the cure-all

Hard to keep up with rapid SSD evolution

- ASIC takes long time to manufacture

Inadequate for cloud features

- Hard for complex logics
- Logics are hard-wired



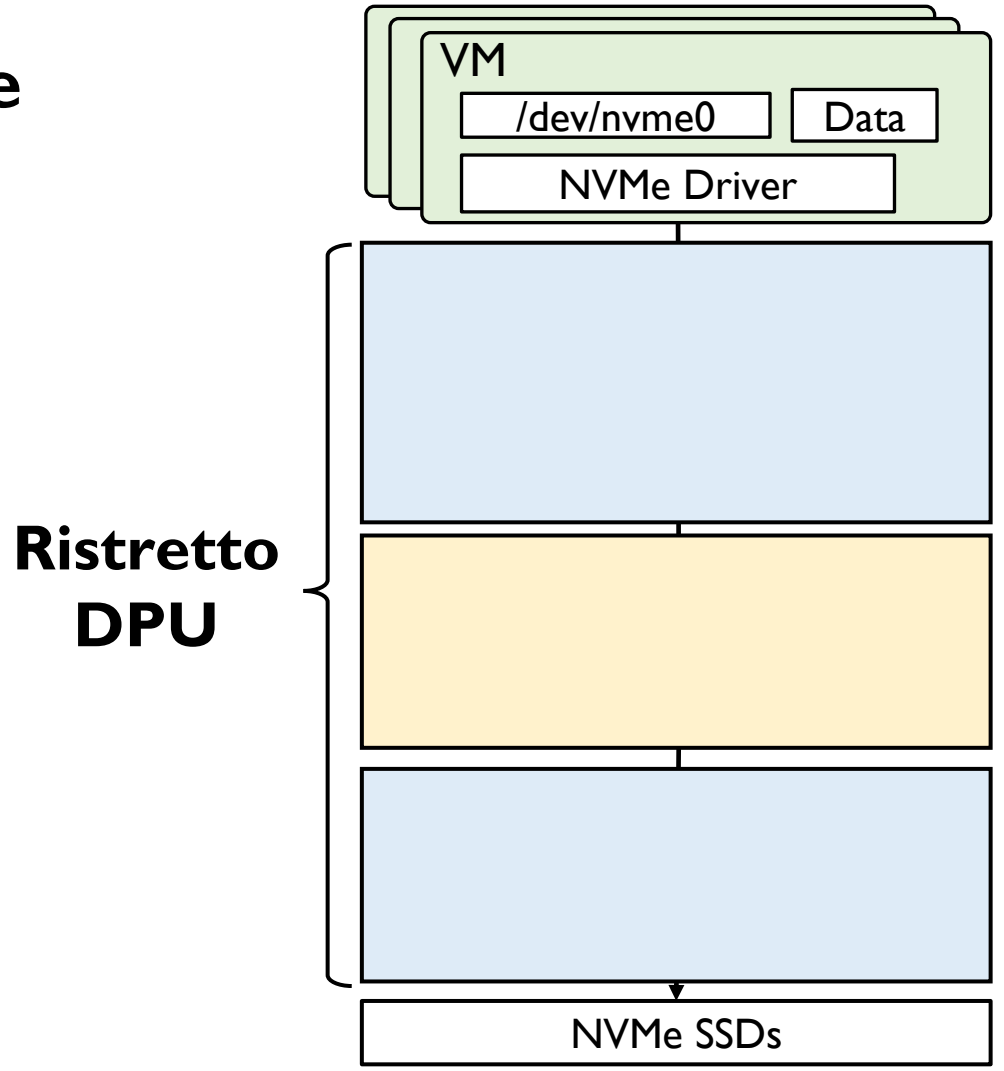
Cloud features

Outline

- Background
- The Past
- **The Present**
- The Future
- Conclusion

Ristretto: From ASIC to ASIC/SoC Co-design

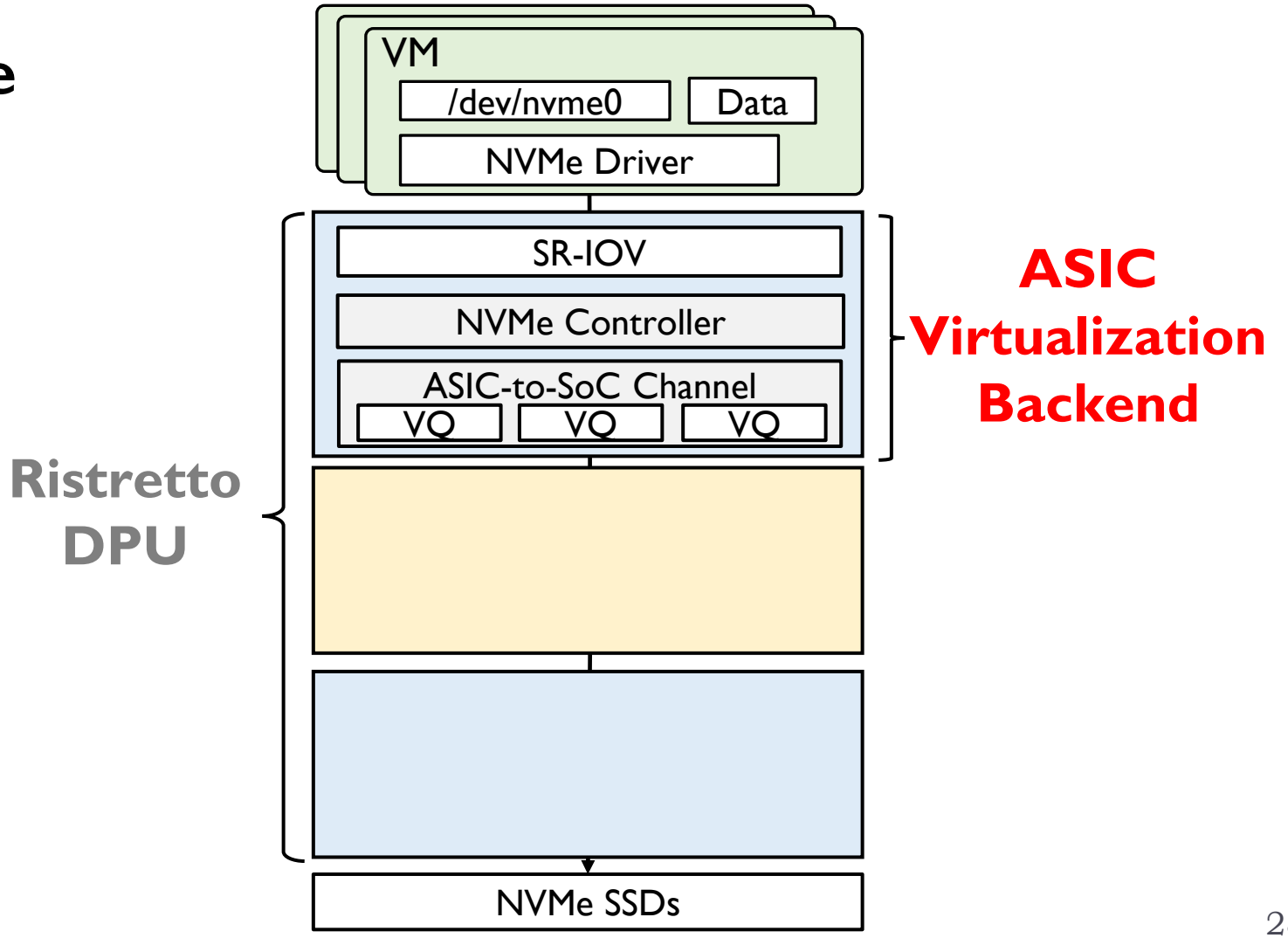
A three-layer architecture



Ristretto: From ASIC to ASIC/SoC Co-design

A three-layer architecture

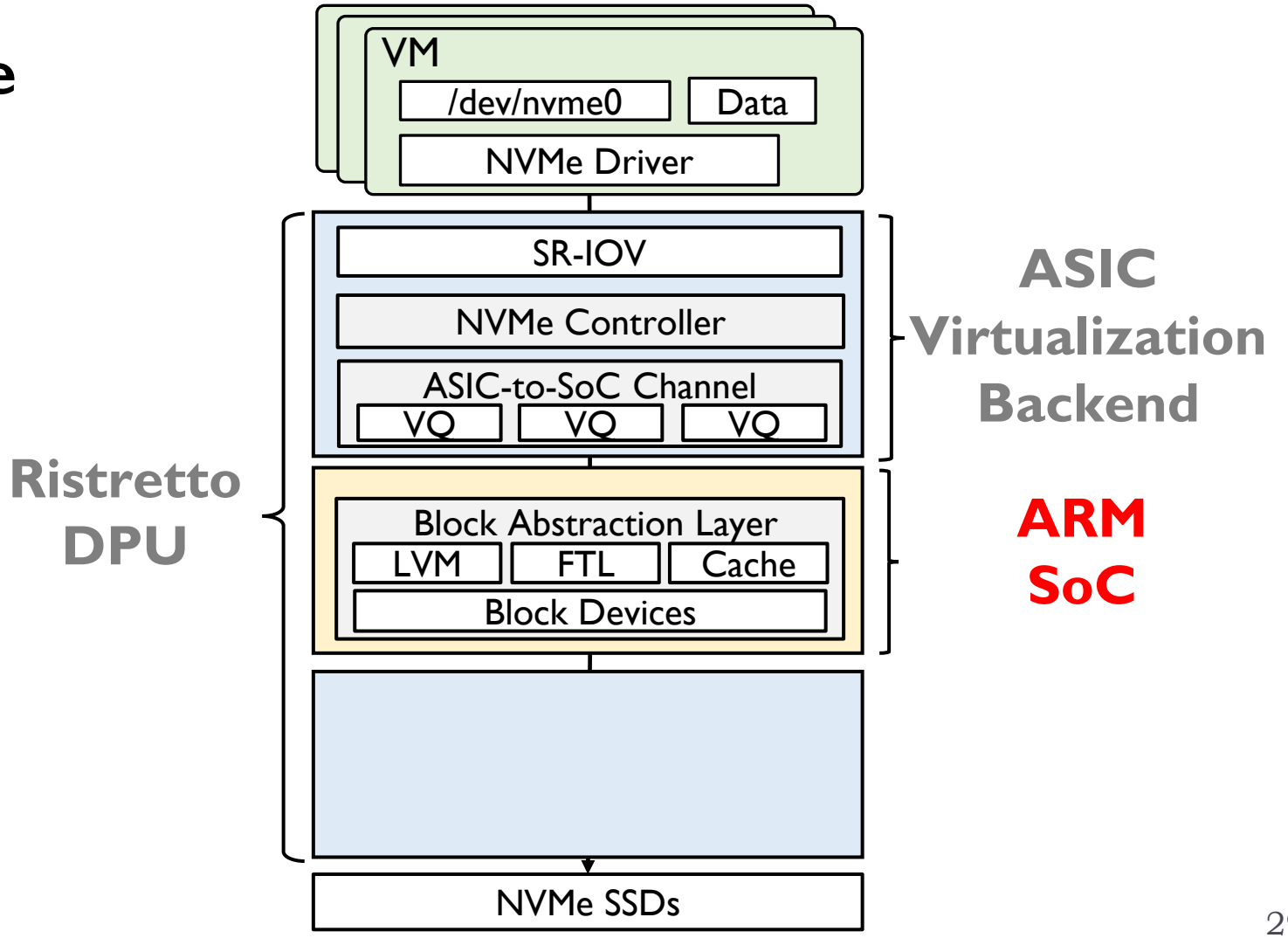
- ASIC virtualization backend



Ristretto: From ASIC to ASIC/SoC Co-design

A three-layer architecture

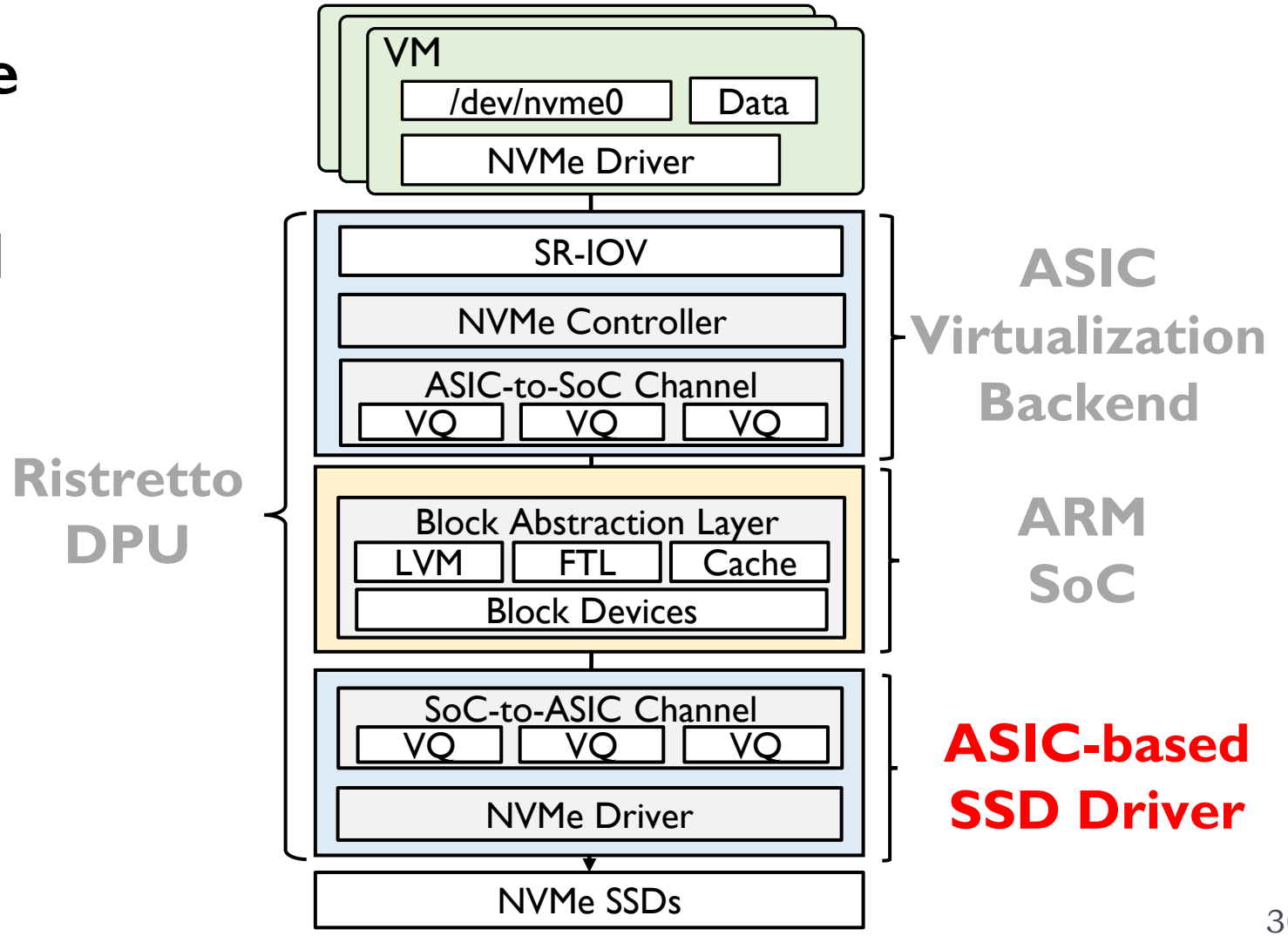
- ASIC virtualization backend
- ARM SoC



Ristretto: From ASIC to ASIC/SoC Co-design

A three-layer architecture

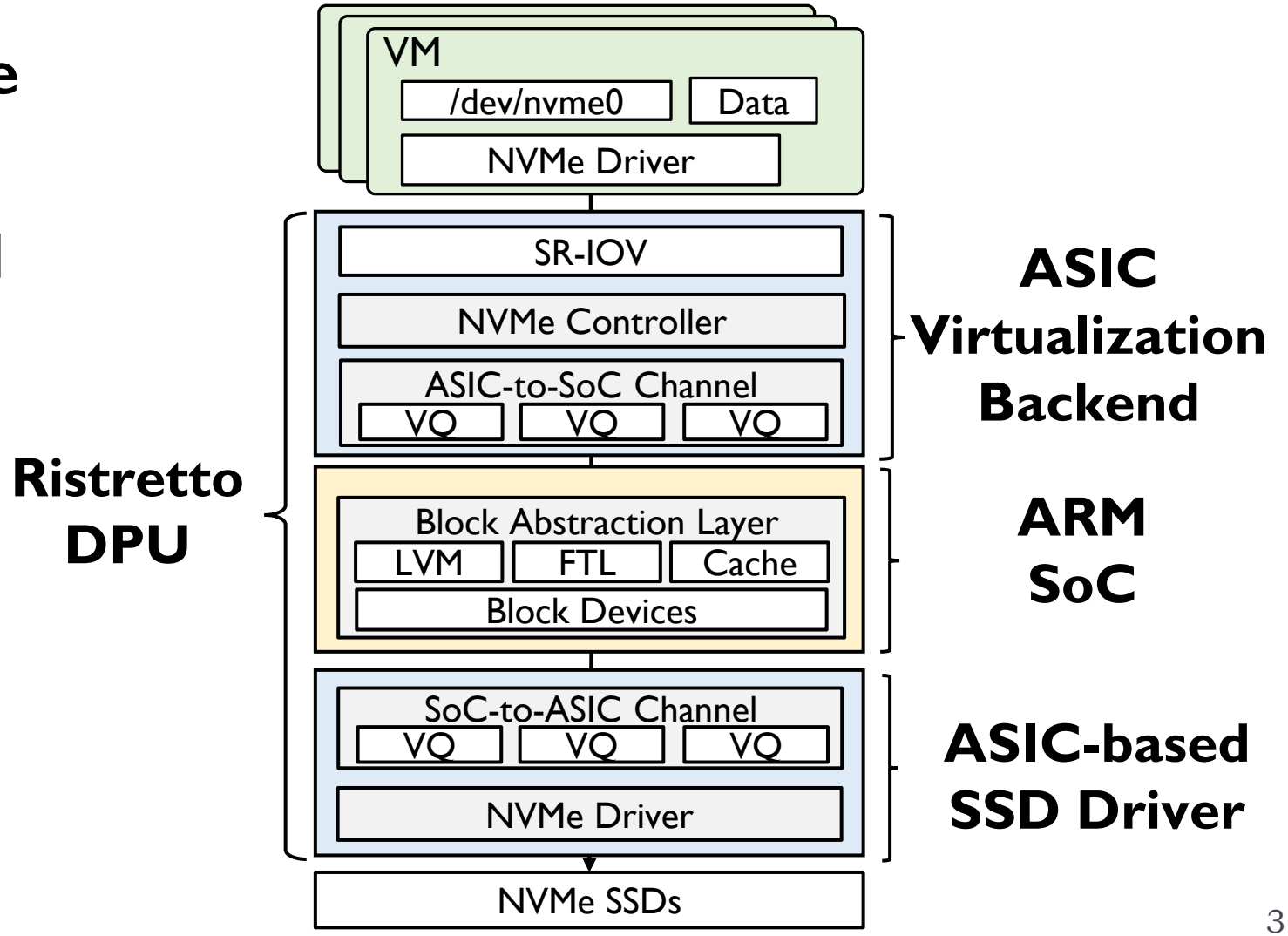
- ASIC virtualization backend
- ARM SoC
- ASIC-based SSD driver



Ristretto: From ASIC to ASIC/SoC Co-design

A three-layer architecture

- ASIC virtualization backend
- ARM SoC
- ASIC-based SSD driver



Outline

- Background
- The Past
- The Present
- **The Future**
- Conclusion

Elastic Block Storage (EBS)

Compute-Storage Co-located

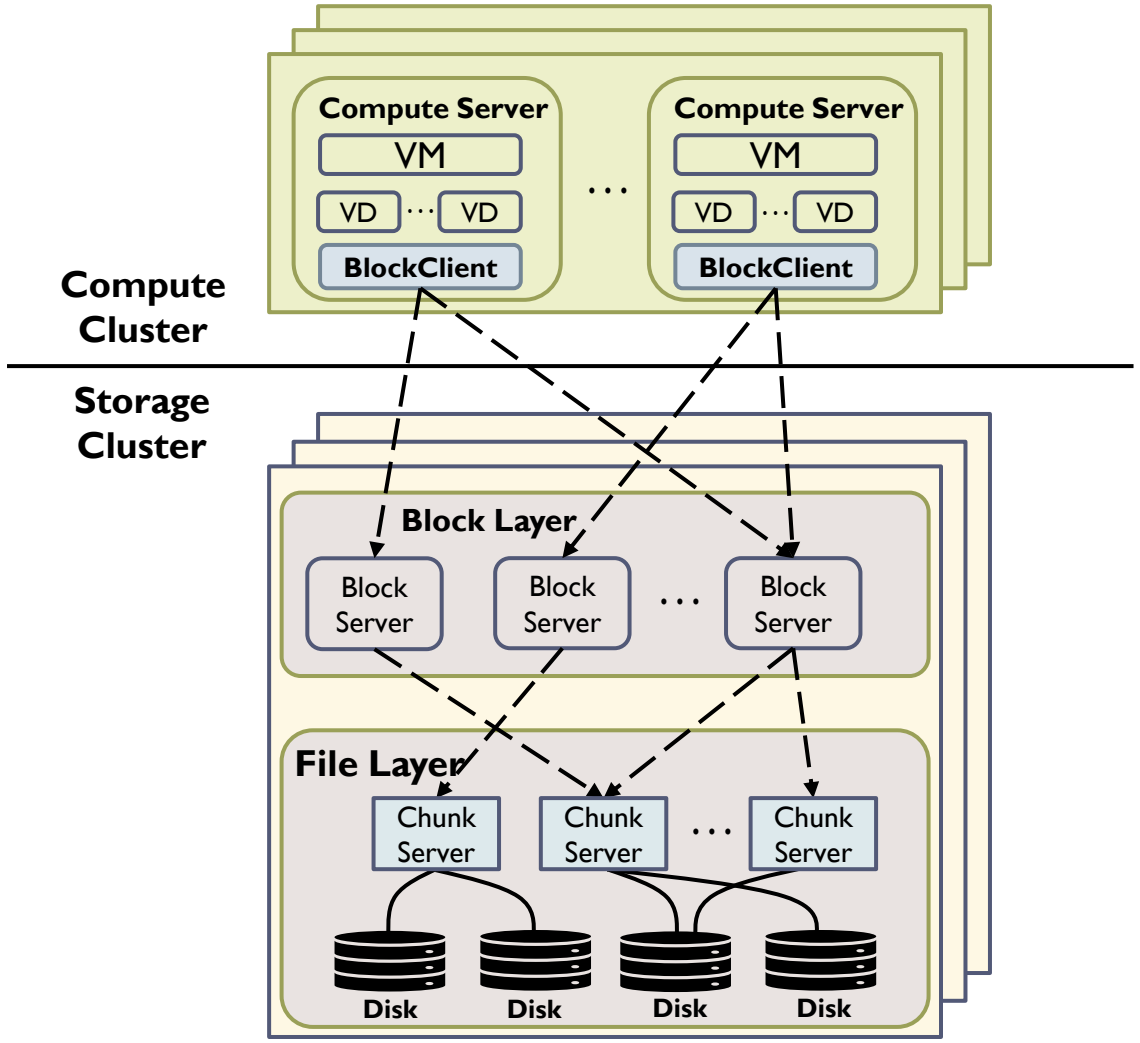
- Local Storage Limitations
  - Limited elasticity (unable to scale)
  - Limited to regions (certain AZs)
  - Limited availability (data loss)

V.S.

Compute-Storage Disaggregation

- EBS Advantages
  - Capacity scaling
  - Easy to access
  - Data redundancy

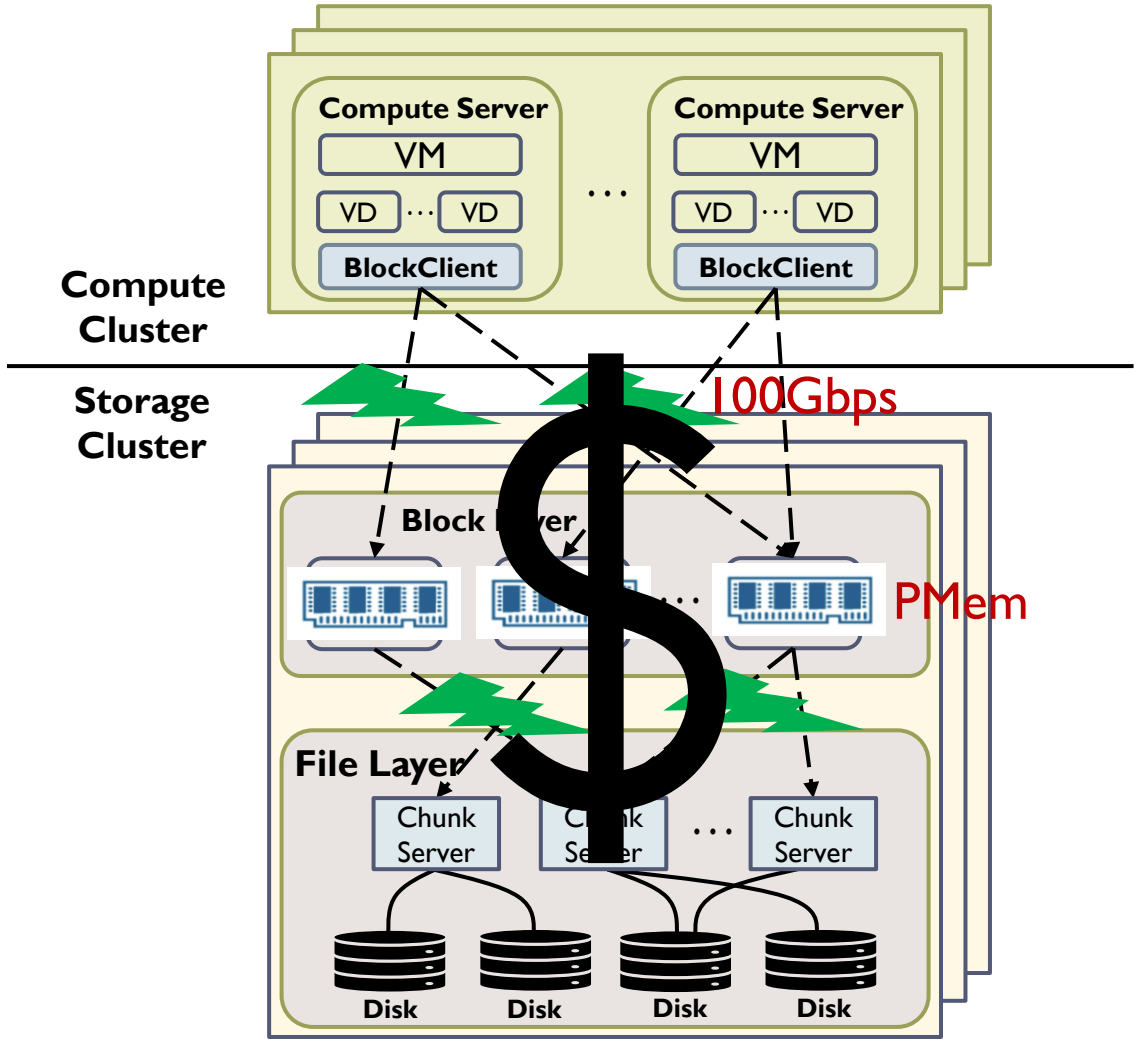
Elastic Block Storage (EBS)



Compute-Storage Disaggregation

- EBS Advantages
  - Capacity scaling (from 1GiB to 64TiB)
  - Easy to access (available in every AZ)
  - Data redundancy (three-replicas/EC)

EBSX ---- High-performance EBS



Compute-Storage Disaggregation

- EBSX as local storage
  - With all EBS advantages
  - Better storage media (PMem)
  - Low latency network (100Gbps)
  - Much higher price (~20x)

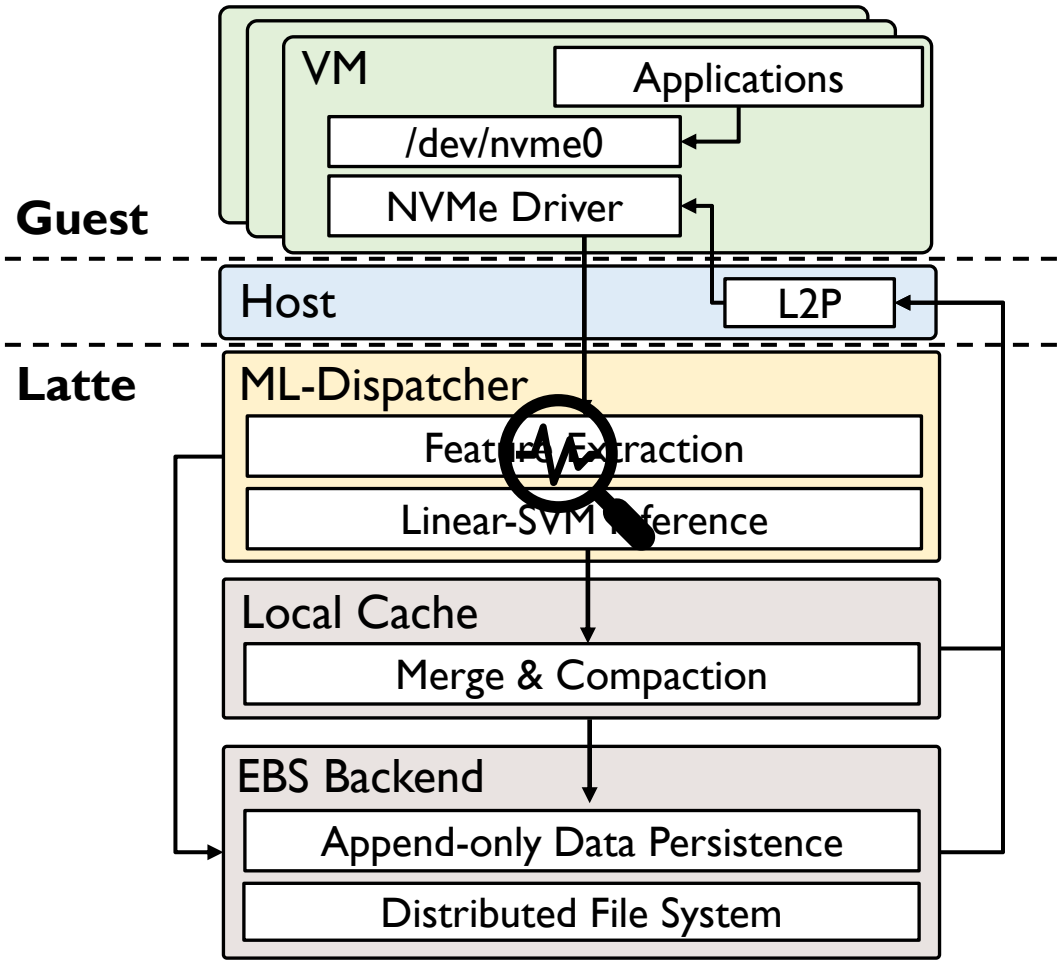
Latte: Local-Cloud Combined Storage

Local as Cache & EBS as Backend

- Append-only write
- Merge I/O in local cache
- Elastic backend capacity
- Strong reliability

ML-based dispatcher for write

- Learn workload pattern
- Write-to/-bypass cache
- Detect I/O tail-latency



(a) Write Path

Latte: Local-Cloud Combined Storage

Local as Cache & EBS as Backend

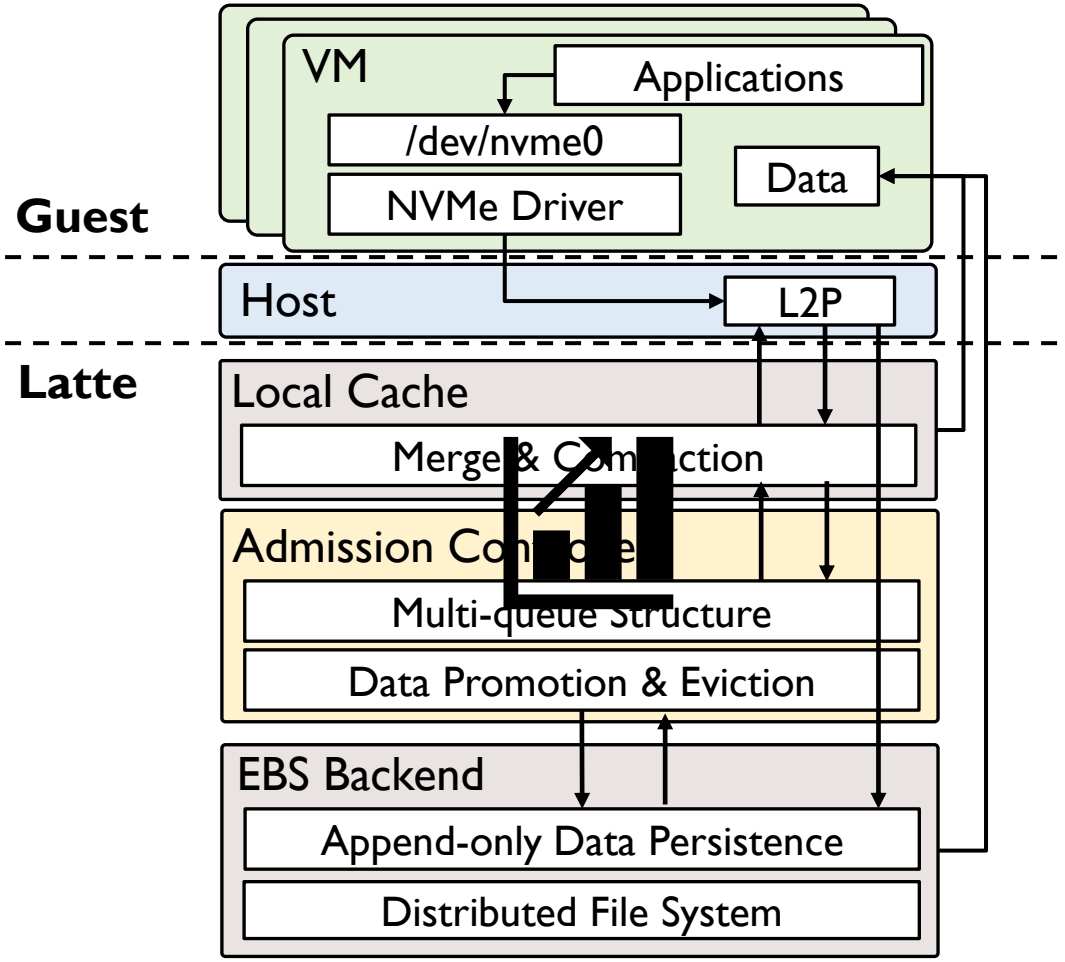
- Low local latency and elasticity of EBS

ML-based dispatcher for write

- Decide I/O path for lower tail-latency

Admission control for read

- Track data access frequency
- Promote hot data & Evict cold data
- Improve cache hit rate and space utilization



(b) Read Path

Experimental Setup

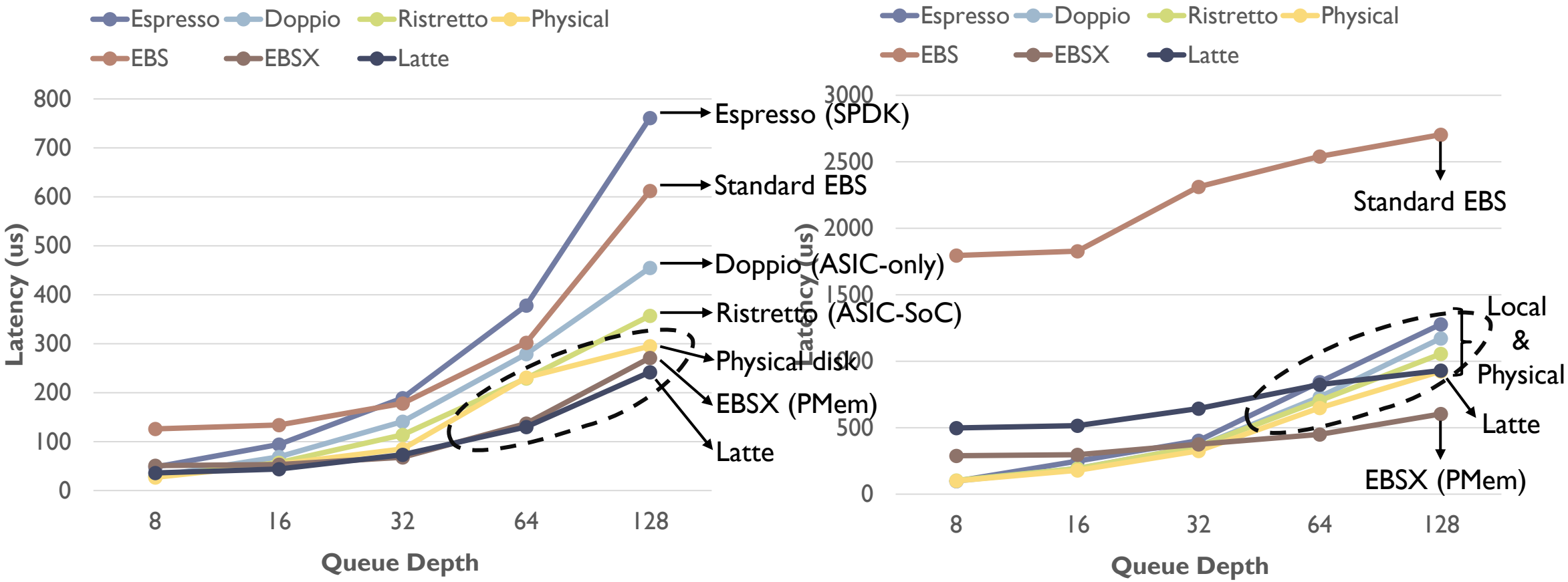
Hardware

- Local storage disks:
  - Espresso, Doppio and Ristretto
- Physical disks:
  - Enterprise Gen4 NVMe SSD
- Cloud disks:
  - Standard EBS, EBSX
- Latte:
  - Ristretto cache + Standard EBS backend

Software

- VMs:
  - 8 VDs backed by 8 PCIe Gen4 local disks
  - 128 vCPU cores; 8 cores for SPDK polling
  - Each mounts one EBS/EBSX cloud disk
- Customized SPDK FTL
  - Mount Latte as an SPDK Block Device in VM
- FIO
  - Micro-benchmarks and trace replay

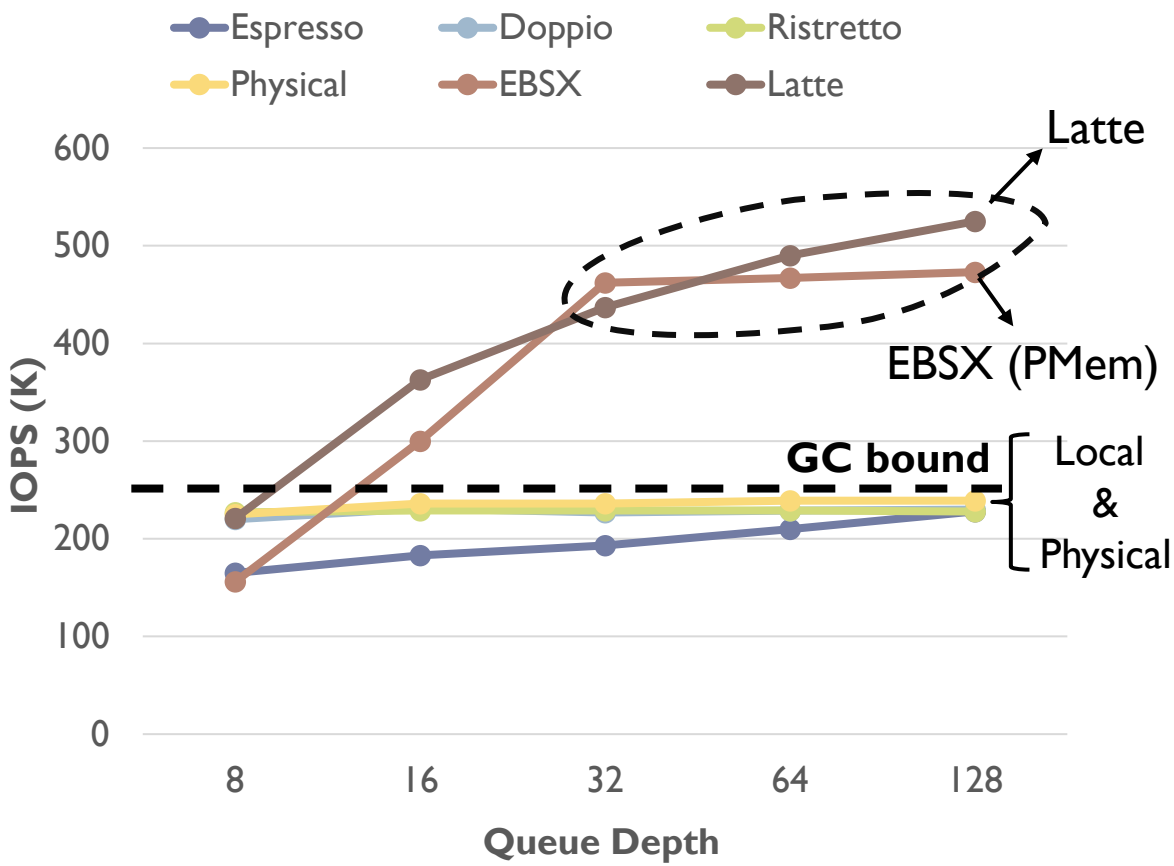
Micro-benchmark



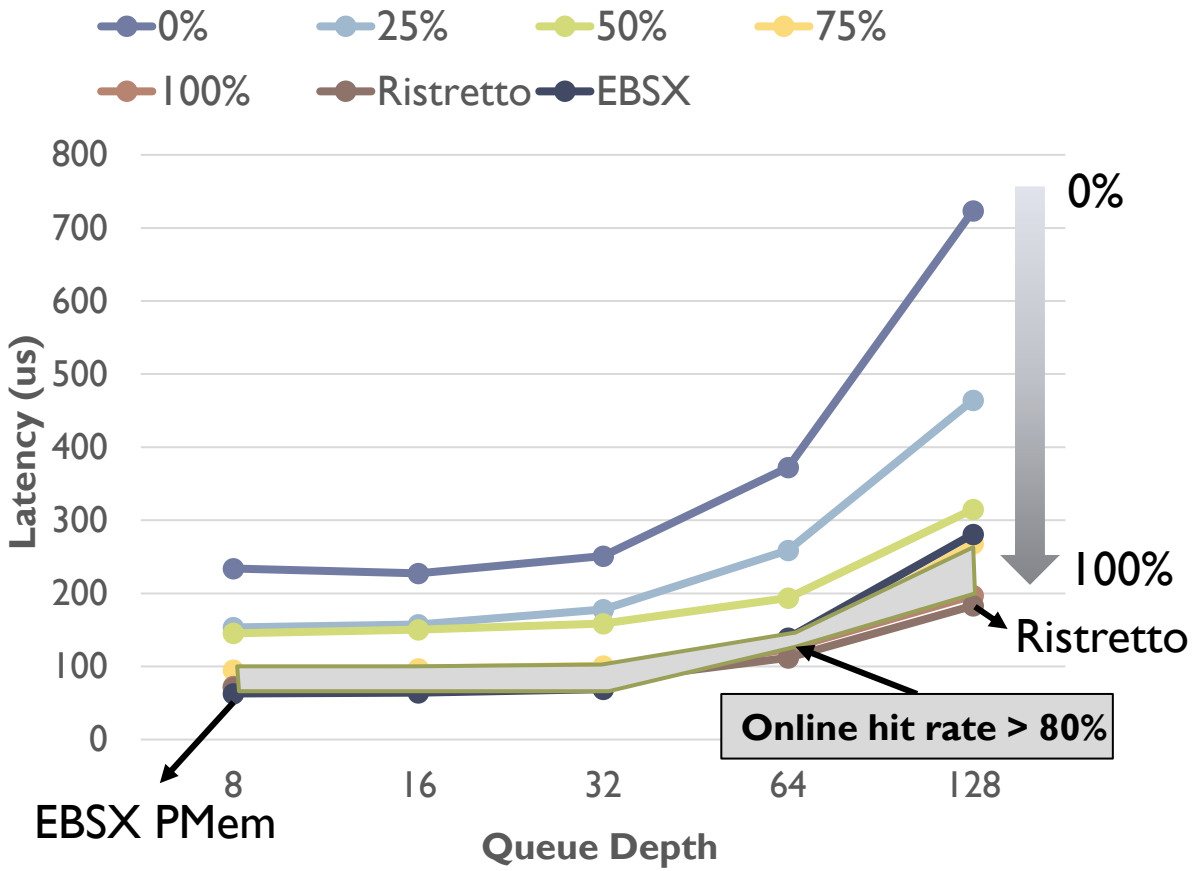
❑ **Avg. Latency**

❑ **P99.9 Latency**

Micro-benchmark

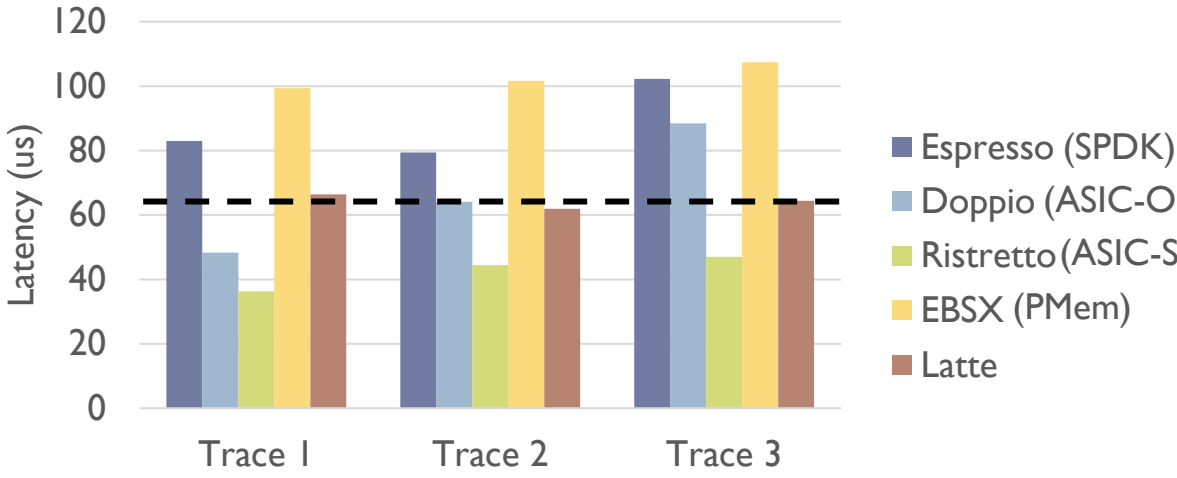


❑ **Write IOPS w/ GC**

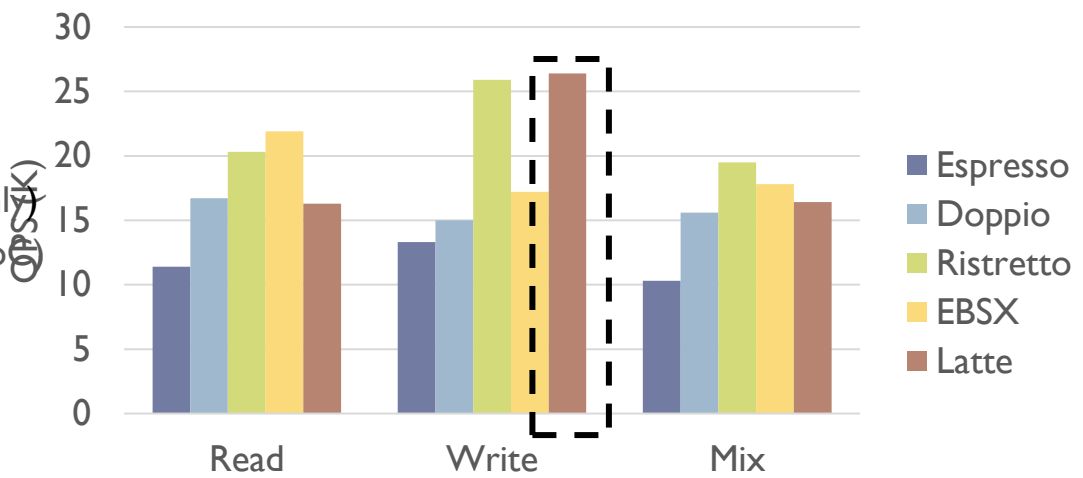


❑ **Read Latencies of various hit rates**

Macro-benchmark



- **Online Traces Replay Latency**



- **MySQL Sysbench Test**

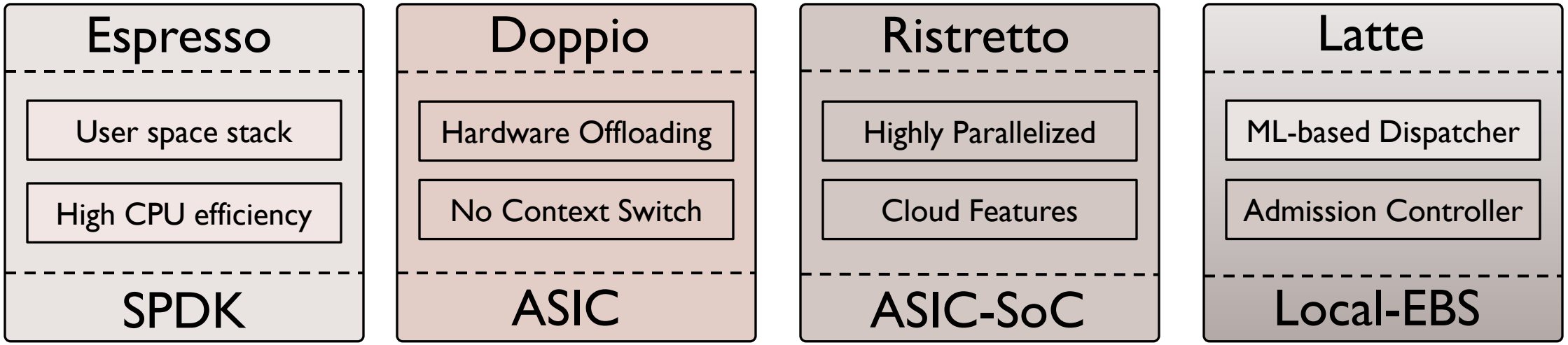
CapEx

Type	Max. IOPS	Max. Throughput	Unit Price
Ristretto	550K	6.7GB/s	1
EBSX	450K	6.0GB/s	~ 20$
Latte	750K	8.9GB/s	2.1 ~ 4.0

- **Local storage** is cheap with limitations
- **EBSX** has much larger CapEx than **local**
- **Latte** reduces unit price with higher perf

Conclusion

- **Evolution of Local Storage Stack**
  - SPDK for Espresso
  - ASIC-only for Doppio
  - ASIC-SoC Design for Ristretto
- **Latte: Future of local storage**
  - Local-cloud hybrid storage stack
  - ML-based dispatcher for write path
  - Admission controller for read path



Thank You!

Q & A

More details are in the paper