KALRAY

STORAGE DEVELOPER CONFERENCE SDC 21
BY Developers FOR Developers

SPDK Implementation on a Manycore / Many node System

Jean-François Marie
Chief Solution Architect

Rémy Gauguey
Senior Software Architect

www.kalrayinc.com

KALRAY

ABSTRACT

ABSTRACT

DPU on Manycore and SPDK are a Great Combination

As you know, the Storage Performance Development Kit (SPDK) provides a set of tools and libraries for writing high performance, scalable, user-mode storage applications.

Kalray's MPPA® manycore architecture proposes a unique 80-cores system. A manycore processor is characterized by an apparent grouping from a software point of view of cores and their portion of the memory hierarchy into computing units. This grouping can delimit the scope of cache consistency and inter-core synchronization operations, include explicitly addressed local working memories (as opposed to caches), or even specific data movement engines and other accelerators. Computing units interact and access external memories and processor I/O through a communication device that can take the form of a network-on-chip (NoC).

The advantage of the manycore architecture is that a processor can scale to massive parallelism by replicating the computing units and extending the network on chip, whereas for a multi-core processor the replication applies to the core level.
For storage purposes, the internal processor clusters are configured with one dedicated cluster as a control and management plane, and the remaining four clusters as four independent data planes.
We have implemented SPDK so that it provides a unique scalable platform that can deliver high performances on an 80-core system.

This presentation will explain how we have ported SPDK on our processor core, and what unique pieces of technologies have been developed in order to coordinate with the processor internals.
We will also explain how the platform can scale.

KALRAY

THE PRESENTERS

Jean-François Marie
Chief Solution Architect

Jean-François has more than 30 years of experience in the high-tech industry. He started his career dealing with real time systems, before joining Sun Microsystems as a data center architect, then EMC² and finally NetApp in 2006, where he had various roles in a 13 -year career. He held various roles, from Chief Technologist and Product Marketing Director for EMEA, to French Expert team manager to handle new technology introduction. He also managed global and regional accounts, alliances and partners.

Jean-François was also an active SNIA member for 10 years and French SNIA President for 2 years. He has a Masters degree in Electronics, specialized in micro-processor design and embedded systems.

On a personal note, he has been a Basket Ball player, a coach and head coach for 25 years.

Rémy Gauguey
Senior Software Architect

Rémy Gauguey is a Senior Software Architect at Kalray, for the Data Center Business Unit. He has more than 25 years of experience in the high-tech industry, with strong expertise in SoCs, RTOS and high-performance packet processing.

He develops advanced architectures for composable infrastructure, leveraging the MPPA® manycore technology from Kalray.

Rémy has been previously developing his expertise at Conexant, Mindspeed Technologies and the CEA labs. He holds several patents in the fields of software architecture and packet processing.

Agenda

1.Why Manycore architecture/processor?

- **The need for change**
- **The need for an open platform**

2. Overview on an SPDK technical implementation

- Why SPDK?
- SPDK optimization for a manycore
- Use Cases

KALRAY

A GROWING NEED TO ANALYSE DATA ON THE FLY

Twitter
500m of Tweets
every day
6000 every second

9x Data created
In the last 2 years
versus Entire History of the humanity!

IoT
6Bu IoT devices
connections by **2026**

Facebook
100m of video hours
seen every days

Data are Exploding
AI, 5G, IoT

Telecom
5Bu Mobile Phone
Users by **2020**

Youtube
5b of videos
every day

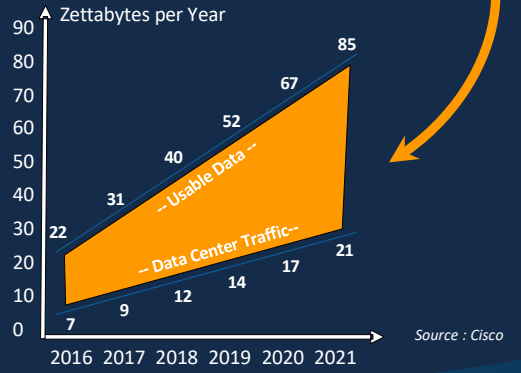
Smart Factory
5 petabytes of video data/day
1 Smart Factory

Autonomous Vehicle
4 terabytes
of data/day
1 Autonomous vehicle

Smart City
200 petabytes of data/day
1 million-people
smart city

ONLY 25% of ”usable” data
will reach a **data center**

75% are Ephemeral
And need to be analyzed
locally in real time



KALRAY

A NEW GENERATION OF COMPUTING PROCESSOR IS REQUIRED

Intelligent Systems Need to Analyze a Huge Flow of Data in Real-Time

COMPUTERS
APPLICATION PROCESSING

SMARTPHONES
CONNECTIVITY & MOBILITY

INTELLIGENT SYSTEMS
ANALYSIS & DECISION-MAKING

1970+ **1990+** **2020+**

A Disruption Requiring
a New Generation of Processors & IP

Multiple emerging end-markets

INTELLIGENT SYSTEMS & EDGE COMPUTING OPPORTUNITIES

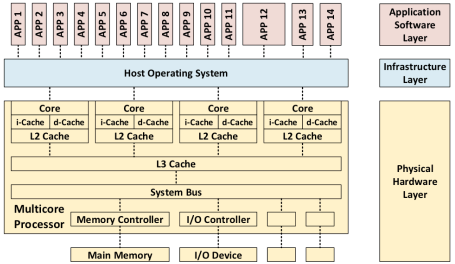
- **Data centers**
- **Autonomous driving**
- Aerospace & Defense
- Industrial 4.0 & Robotics
- 5G / Telecom
- Other Edge Computing Applications

KALRAY

MULTICORE & MANYCORE PROCESSORS

The Key Architectural Differences

Homogeneous Multicore Processor



Multiple CPU cores sharing a cache-coherent memory hierarchy

- Scalability by replicating CPU cores
- Standard programming models

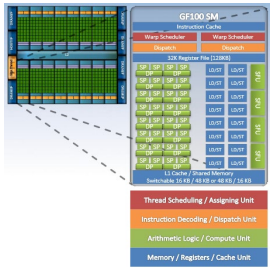
Energy efficiency issues

- Global cache coherence scaling

Time-predictability issues

- No scratch-pad or local memories

GPGPU Manycore Processor



Multiple Streaming Multiprocessors

- Restricted programming models

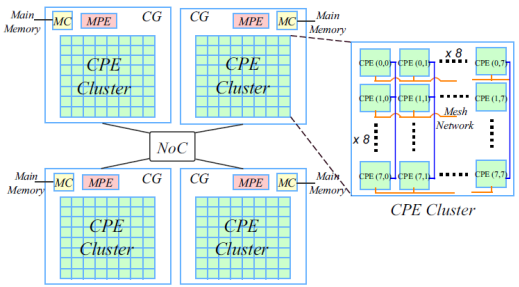
Performance issues of 'thread divergence'

- Branch divergence slow down the execution
- Memory divergence: non-coalesced accesses

Time-predictability issues

- Dynamic allocation of thread blocks
- Dynamic scheduling of warps

CPU-Based Manycore Processor



Multiple "Compute Units" connected by a network-on-chip (NoC)

- Scalability by replicating Compute Units
- Standard multicore programming inside a Compute Unit

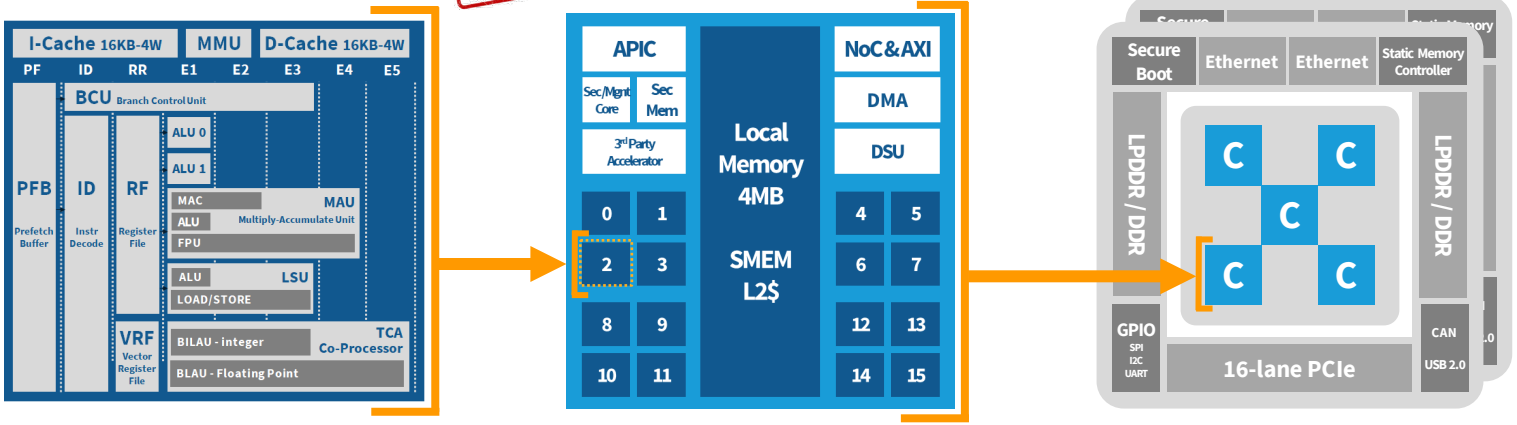
Compute Unit

- Group of cores + DMA
- Scratch-pad memory (SPM)
- Local cache coherency

MANYCORE ARCHITECTURE EXAMPLE

The I/O Processor for Next Gen Intelligent Systems

PATENTED



MANYCORE CORE

- VLIW 64-bit core
- 6-issue VLIW architecture
- MMU + I&D cache (16KB+16KB)
- 16-bit/32-bit/64-bit IEEE 754-2008 FPU
- Vision/CNN Co-processor (TCA)

CLUSTER / COHERENT GROUP OF CORES

Architecture

- 16 cores
- 1 safety/security dedicated core
- 600 to 1200 MHz

Memory

- L1 cache coherency (configurable)
- 4MB configurable memory (L2 cache)
- 256 bits / bandwitdh up to 614GB/s)

MULTI CLUSTER ARCHITECTURE

5 Clusters: 80 cores + 80 co-processors

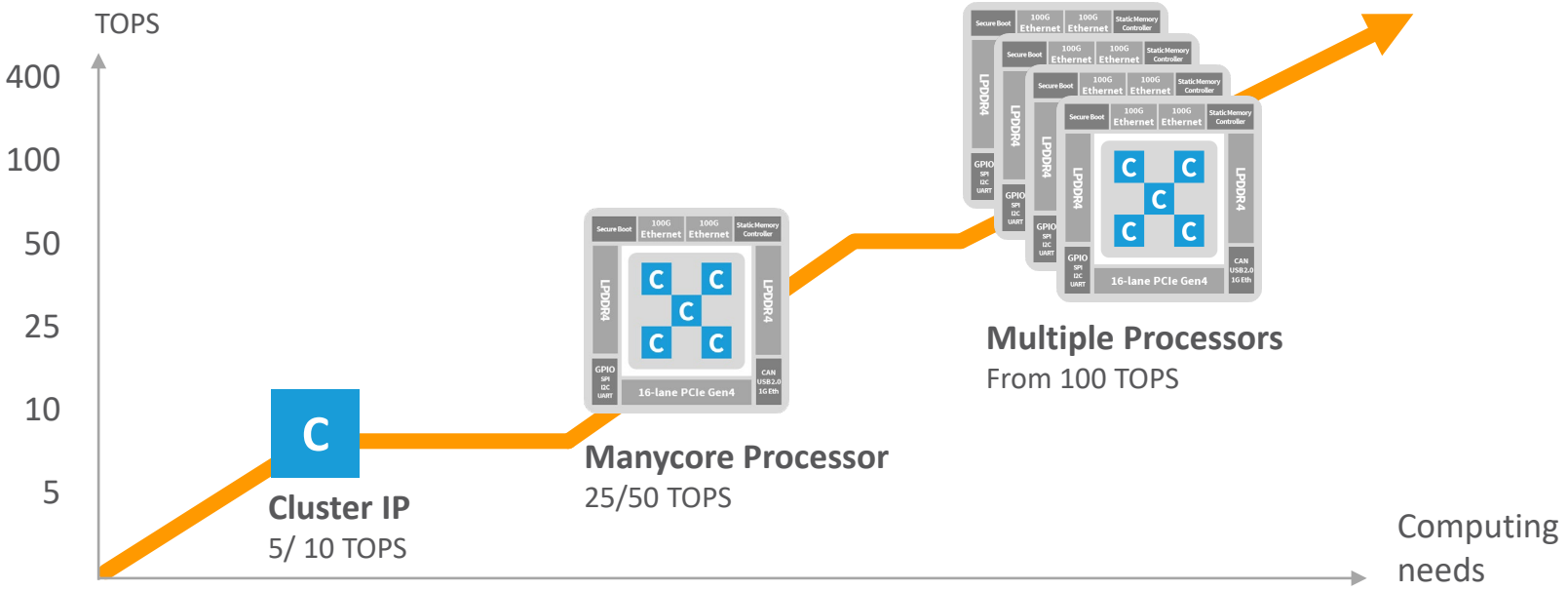
- Load Balancer / Packet Parser
- 2x100Gbps Ethernet
- PCIGen4
- DDR4 - 3200

AXI Bus + NoC Bus

- L2 refill in DDR and direct access to DDR from clusters
- DMA-based highly efficient data connection

KALRAY

MANYCORES ARE SCALABLE BY ESSENCE
From IP to Multi-die, Ensure Software Scalability



DPU POSITIONING

Overcome Weaknesses of Traditional Solutions

High Computing Efficiency
(Performance of a high-end DSP/GPU)

Power Efficiency
(Sub 30W chip typ.)

Easily Programmable
(C/C++/Open CL; Linux; POSIX; RTOS)

Concurrent Execution of Dozens of Heterogeneous Critical Tasks

Low Latency / Deterministic / Real-Time
(High speed I/O – RDMA type of architecture)

Performance Scalability
(within die; die-to-die; many dice)

DPU	FPGA	GPU	x86
✓	✓	✓	
✓		X	X
✓	X		✓
✓	✓	X	X
✓	✓	X	X
✓			

AN OPEN AND STANDARDS BASED FRAMEWORK
The Core of a Storage OS

Highly Efficient

Use multi-OS capability of MPPA®, for maximum efficiency

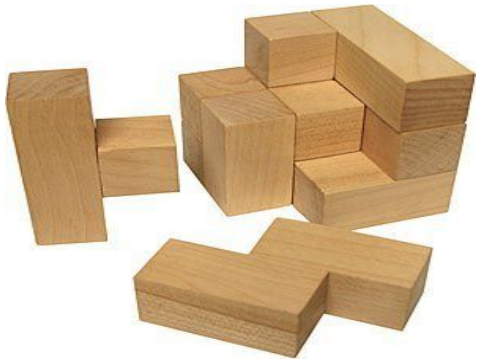
- Linux: Control and Management plane (typical : 1 Cluster - 16 cores)
- Cluster OS (light OS): Data plane (typical: 1 to 4 Clusters – 16 to 64 cores)

AccessCore® SDK
Open Software & Tools

Easy Software Development and Migration

Standard APIs & Tool Chain

- Linux: **SPDK**, nvmet, virtio, ibverbs …
- Cluster OS: verbs API, sockets, **SPDK nvme, SPDK BDEV**, …
- Yocto Build, GCC, GDB, LLVM



Modular & Scalable

Provided with optimized baseline software modules

- Network functions: OpenDataPlane / Network stack
- Storage functions: **SPDK BDEV layer**; NVMe-oF + NVMe

KALRAY

SMART STORAGE CONTROLER

Builds Next Generation High Perf, Secure, Scale Out, NVMe-oF Storage

HIGH AVAILABILITY

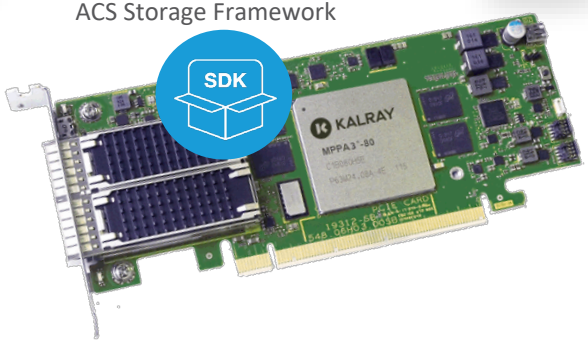
DATA PROTECTION

SECURITY

DATA MANAGEMENT

STORAGE EFFICIENCY

FULLY PROGRAMMABLE
ACS Storage Framework



HIGH-SPEED I/O INTERFACES
NVMe-oF TCP & RoCE
(2x100GbE)

ZERO HOST CODE IMPACT
x86 transparent integration
via NVMe emulation
Or no x86

HIGH-PERF SSDs I/F
(RC or P2P PCIe Gen4)

LOW CONSUMPTION
30W Typ.

Agenda

1.Why Manycore architecture/processor?

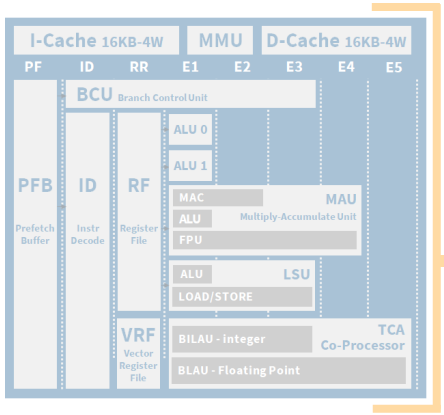
- The need for change
- The need for an open platform

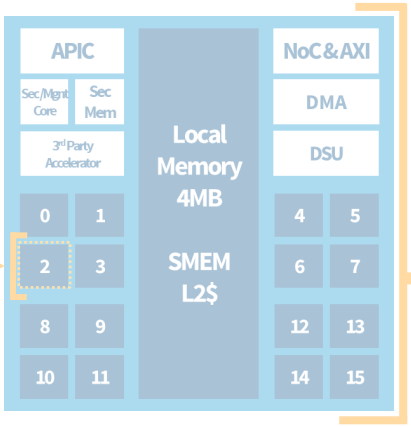
2. Overview on an SPDK technical implementation

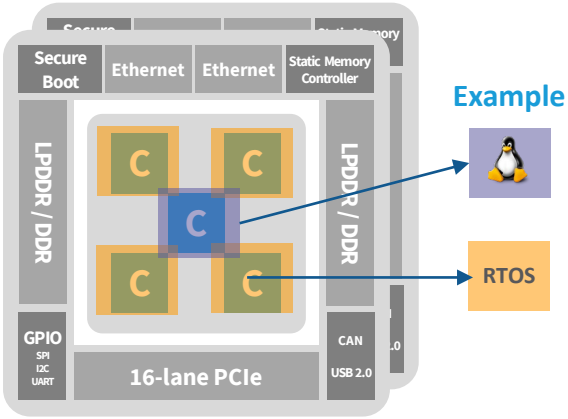
- **Why SPDK?**
- **SPDK optimization for a manycore**
- **Use Cases**

KALRAY

COOLIDGE™ ARCHITECTURE
A Multi-OS / Multi-System Architecture







3RD GENERATION KALRAY CORE

- VLIW 64-bit core
- 6-issue VLIW architecture
- MMU + I&D cache (16KB+16KB)
- 16-bit/32-bit/64-bit IEEE 754-2008 FPU
- Vision/CNN Co-processor (TCA)

CLUSTER

Architecture

- 16 cores
- 1 safety/security dedicated core
- 600 to 1200 MHz

Memory

- L1 cache coherency (configurable)
- 4MB configurable memory (L2 cache)
- 256 bits / bandwidth up to 614GB/s)

MULTI CLUSTER ARCHITECTURE

5 Clusters: 80 cores + 80 co-processors

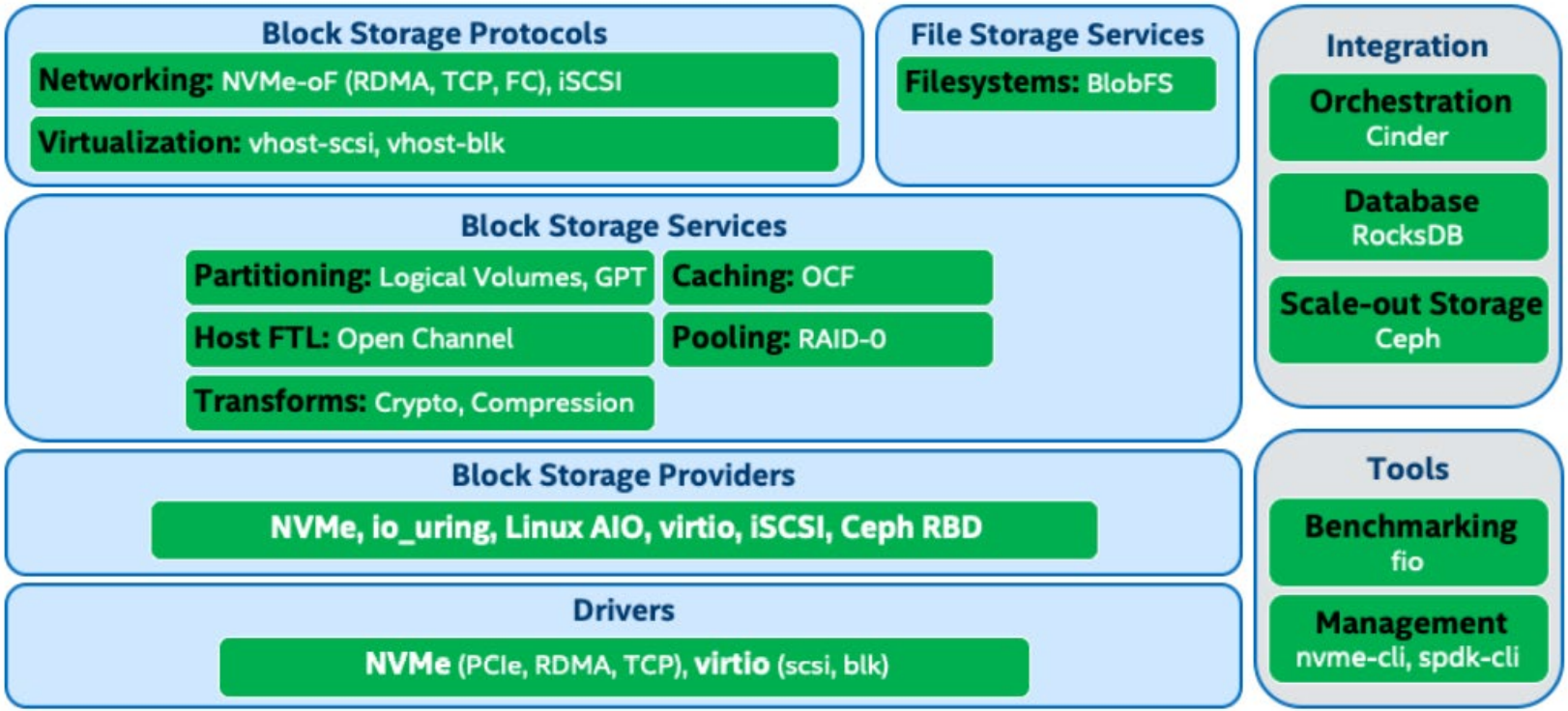
- Load Balancer / Packet Parser
- 2x100Gbps Ethernet
- PCIGen4
- DDR4 - 3200

AXI Bus + NoC Bus

- L2 refill in DDR and direct access to DDR from clusters
- DMA-based highly efficient data connection

ACCESSCORE® FOR STORAGE & NETWORKING

SPDK Layer Details



SPDK
Software Architecture Overview

Performance via Concurrency

- Instead of context switching, dedicate core(s) to specific tasks
- Avoid interrupt handler overhead and latency by polling
- Instead of locks, pass messages

Threading model

- Model: Many connections per thread with polled asynchronous I/O
- Low memory overhead,
  No interrupts,
  No context switching

Open Source Software

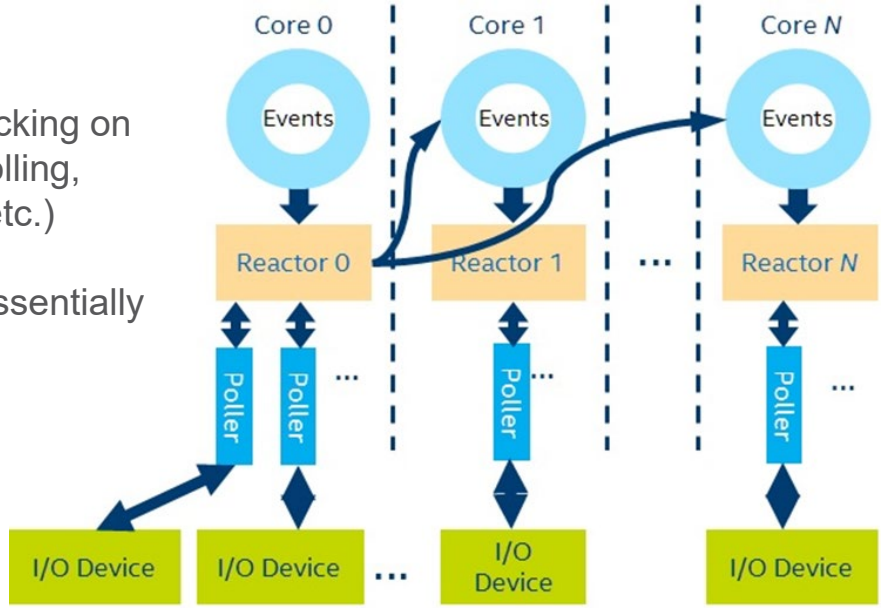
- New features continuoulsy added (ex ZNS)
- Can develop features on x86 then port to MPPA®
- Integration with Openstack Cinder

Scalable & efficient

- User space, lockless, polled mode components
- Customizable via BDEV layer (provide ‘storage services’ like compression, dedup, RAID)

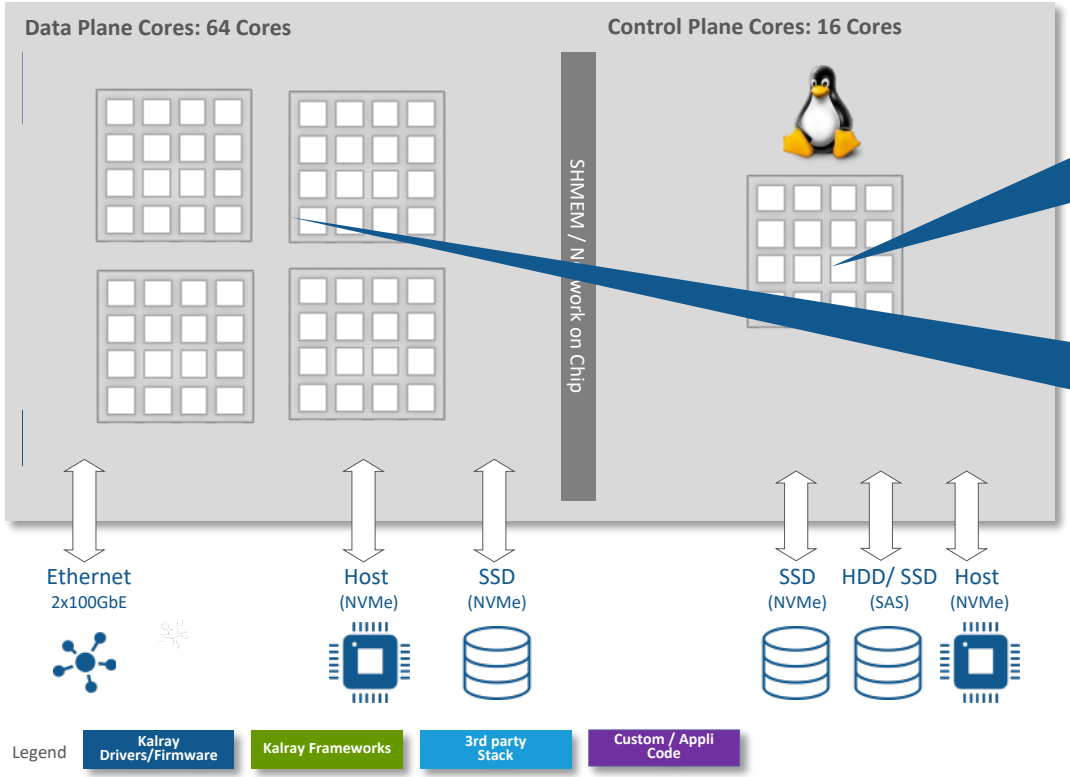
SPDK
Software Architecture Overview

- **Reactor**: Scheduler pinned to cores
- **Poller**: “Task” running on a reactor checking on asynchronous events (ex RDMA CQ polling, NVMe CQ polling, TCP socket polling etc.)
- **Event**: Cross thread communication, essentially used for control plane
- **I/O channel**: Abstract h/w I/O queues



Storage Performance Development Kit (SPDK) Application Event Framework (Reactor, Event, Poller)

ACS4.x ARCHITECTURE
A Fully Flexible Software Environment

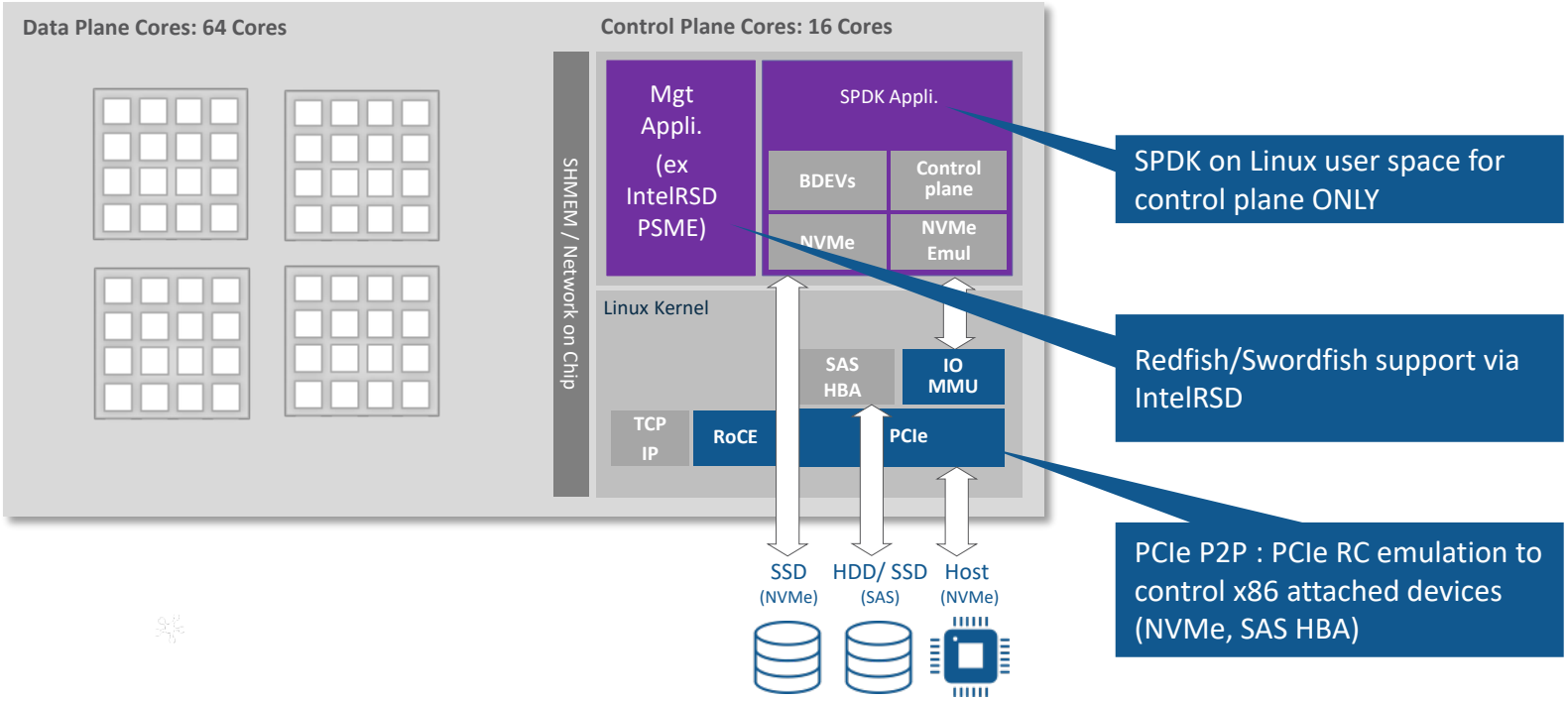


Linux on control plane cluster (16 cores)
- Management (RestAPI)
- Control plane (SPDK)

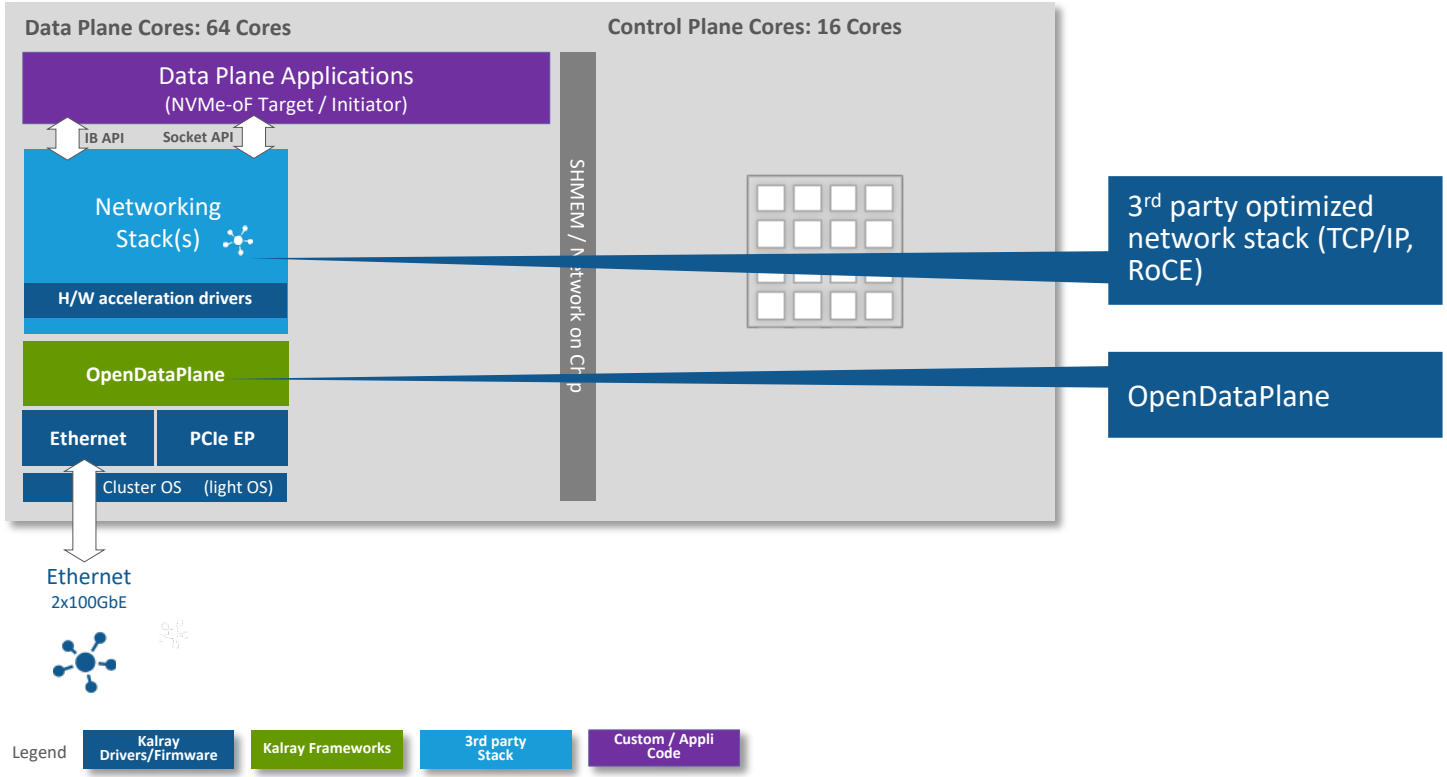
ClusterOS on 4 data plane clusters (64 cores)
- Network stack (TCP/RoCE)
- SPDK on CoS (4 instances)

ACS4.x ARCHITECTURE

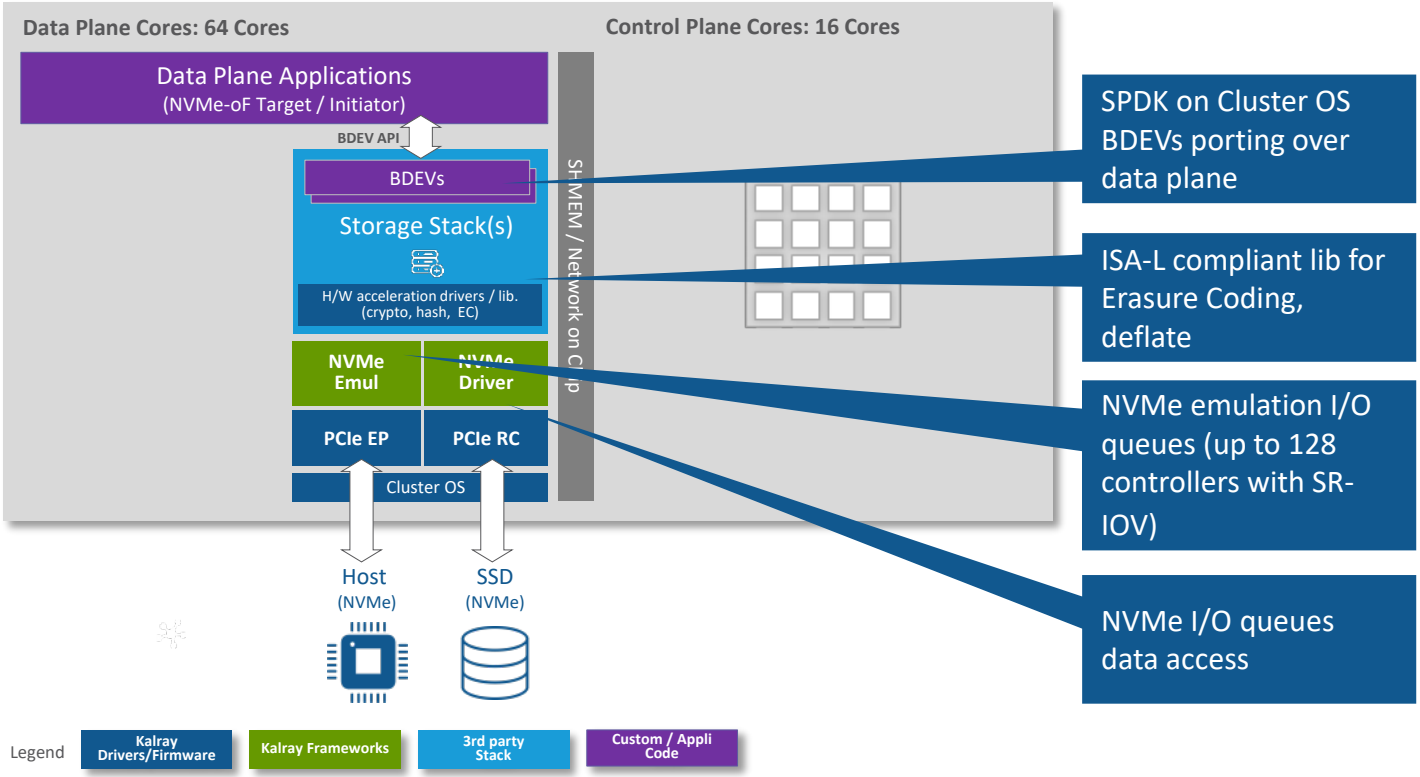
A Fully Flexible Software Environment



ACS4.x ARCHITECTURE
A Fully Flexible Software Environment

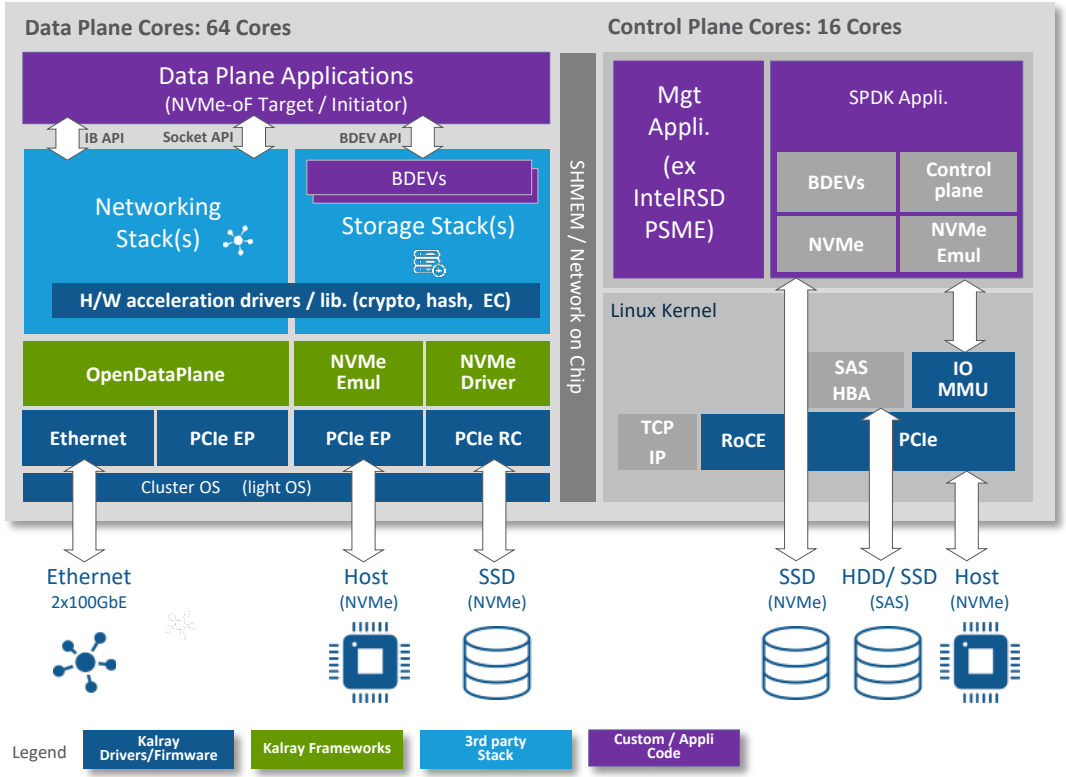


ACS4.x ARCHITECTURE
A Fully Flexible Software Environment



KALRAY

ACS4.x ARCHITECTURE
A Fully Flexible Software Environment



- A complete & modular software framework
- Based on an optimized SPDK for both data plane **AND** control plane
- Open to partners

KALRAY

SPDK ON MPPA®

An Optimized Manycore Implementation

A scalable and multi-instances SPDK implementation

- 5 SPDK instances on 5 MPPA® ‘clusters’
  - 1x on Linux mainly for control plane, but also for non accelerated protocols (ex iSCSI)
  - 4x on light for data plane (I/O queues only)
- A single SPDK seen from external management

A modular Block Device Abstraction Layer

- BDEVs can be fed from network (TCP/RoCE), PCIe (NVMe emulation), or even inter-cluster communication path

SPDK on a proprietary lightweight OS

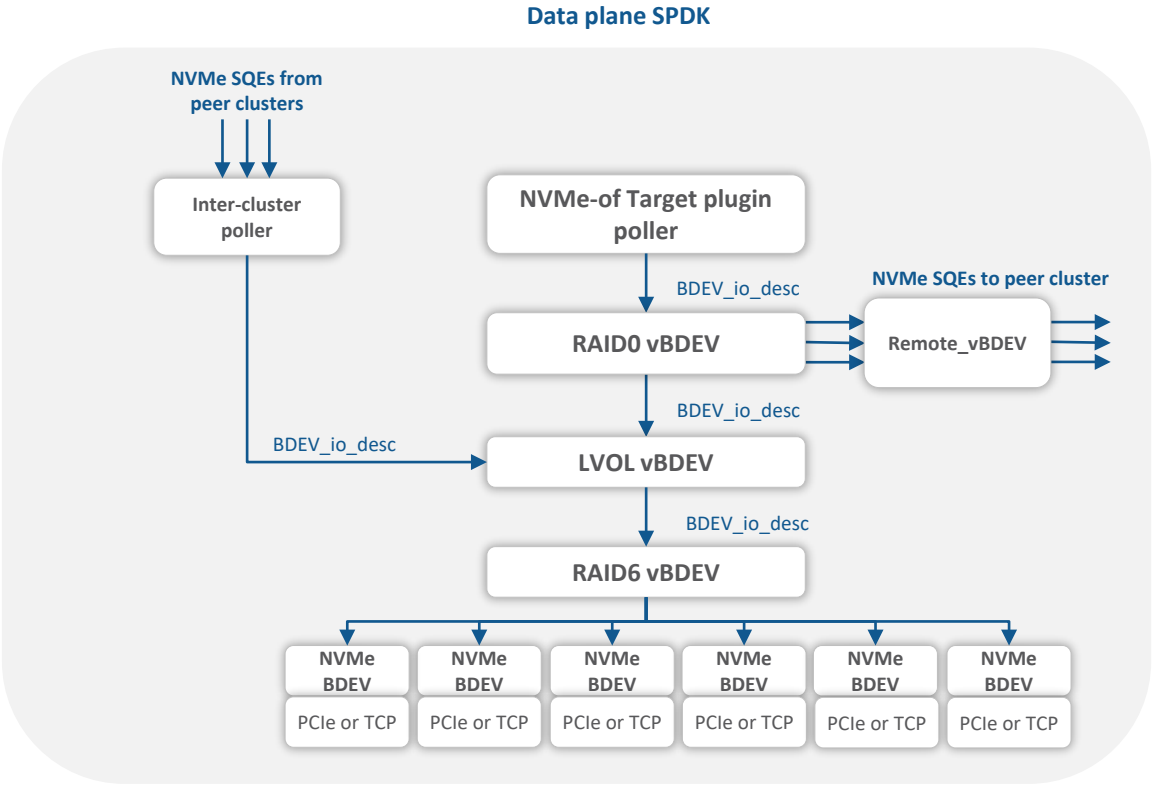
- New ‘Event Abstraction Layer’: DPDK EAL replaced by ODP EAL
- Cache optimized implementation
  - Pipelined implementation with some core dedicated to networking (TCP / RoCE) and others to BDEVs

An efficient inter-cluster I/O communication

- A “Share Nothing” approach between SPDK instances
  - Create parallelism by stripping I/Os
- Zero-copy I/O exchange between SPDK instances
- Zero-copy I/O exchange with Linux SPDK for non NVMe storage (ex iSCSI)

SPDK ON MPPA®

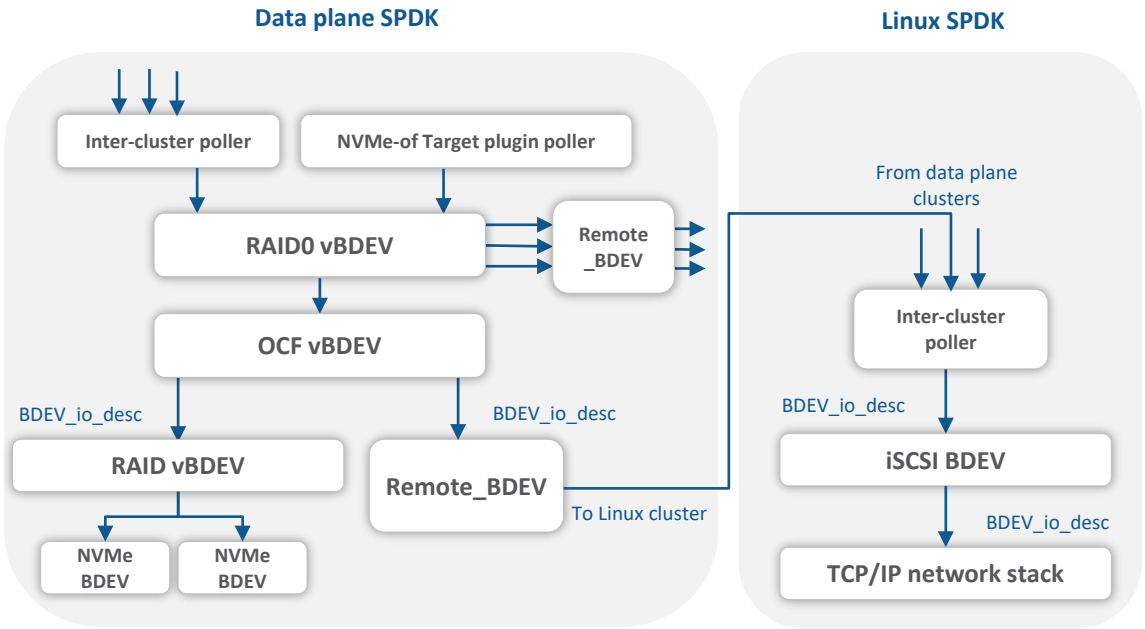
Use Case: Distributed Logical Volume



- NVMe I/Os stripped to several dataplane clusters for a ‘share nothing’ approach between SPDK instances
- Work seamlessly with PCIe or network (TCP/RoCE) target interface
- Zero copy on data blocks located in shared memory area
- Scalability with multi-MPPA® architecture
- Multi SPDK management hidden by Linux SPDK instance
- Up to 1MIOPS @4K

KALRAY

SPDK ON MPPA®
Use Case: iSCSI HDD Caching

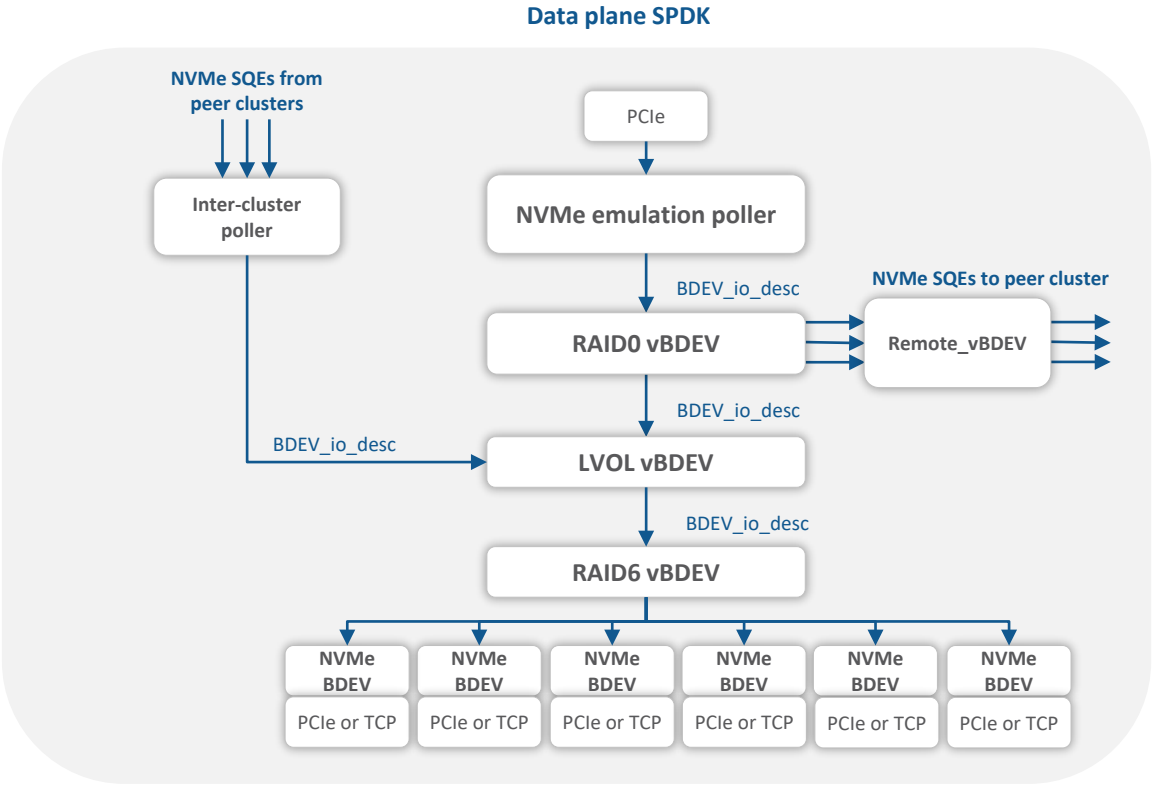


- NVMe-oF exposed iSCSI volumes cached by NVMe SSDs
- Open CAS Framework (OCF) on data plane SPDK
- NVMe I/Os stripped to several dataplane clusters to ensure independent OCF layers (no metadata sharing)
- Zero copy on data blocks located in shared memory area between data plane and Linux
- iSCSI initiator on Linux

KALRAY

SPDK ON MPPA®

Use Case: NVMe to NVMe-oF Initiator



- NVMe-oF connectivity for Bare Metal Cloud environment or OS without NVMe-oF support (ex: Windows)
- Expose several NVMe logical volumes via PCIe SR-IOV (NVMe emulation)
- Local PCIe backend devices (PCIe Peer to Peer)
- Remote (NVMe-oF TCP/RoCE) backend devices
- Up to 2MIOPS @4K (local backend devices) or 1MIOPS @4K (remote NVMe-TCP devices)

KALRAY

KALRAY

STORAGE DEVELOPER CONFERENCE
SDC 21
BY Developers FOR Developers

THANK YOU

www.kalrayinc.com

DISCLAIMER

Kalray makes no guarantee about the accuracy of the information contained in this document. It is intended for information purposes only, and shall not be incorporated into any contract. It is not a commitment to deliver any material, code or functionality, and should not be relied upon in making purchasing decisions. The development, release and timing of any features or functionality described for Kalray products remains at the sole discretion of Kalray.

Trademarks and logos used in this document are the properties of their respective owners.