Single-sided messaging for accelerators: A directional talk

Garret Swart, Oracle

CERN openlab Workshop. 23 January 2020

Safe Harbor

The following is intended to outline our general product direction. It is intended for information purposes only, and may 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, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

Overview

- Single-sided messaging provides hardware-managed communications between network endpoints
  - Embodied in technologies like: SPDK, ibverbs, RDMA, NVMe/RDMA, RDMA/RoCE, CXL
- Direct support for these technologies inside accelerators can allow for efficient, flexible, interoperable communication between heterogenous CPUs, GPUs, FPGAs, and ASICs
- Oracle Exadata, a storage accelerator, is a commercial proof point using one-sided RDMA to access a Persistent Memory storage tier

RDMA Read/Write Pattern

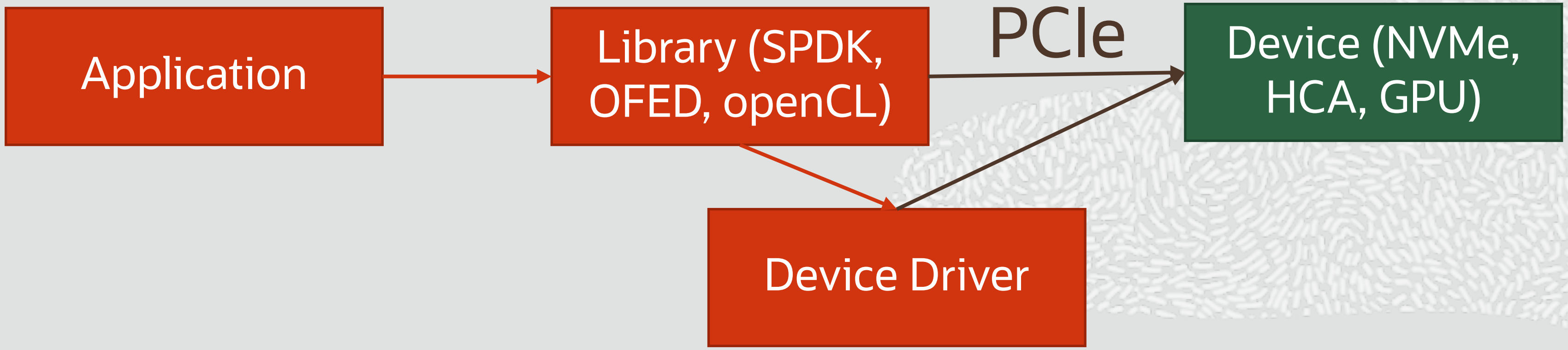
- Reads and Writes are directed to a memory region registered by an EndPoint
- Data is transferred and the response is sent when the transfer is complete
  - Transfer complete does not ensure Written data is visible, must follow Write with Read to ensure visibility
- RDMA used to require InfiniBand, but is now available on Ethernet: RDMA over Converged Ethernet (RoCE)

NVMe Request-Response Pattern

- Each request is self-describing and in a standard format including: Request ID, Namespace ID, operation, arguments
- Client maps Namespace ID to the Endpoint and formats the request
- Endpoint accepts requests and validates the Namespace/Target ID, operation and optional capability
- Each request results in a response, tied to the request ID, containing a return code, and a response payload
- NVMe requests can be transported over PCIe or RDMA/RoCE

~~OS Drivers~~

- OS Device Drivers provide a uniform SW access mechanism to a class of devices and allow multiple applications to share a device
- IB verb and SPDK libraries disintermediate the OS Driver for individual message and I/O interactions.
- OS is used to map device control registers and queues so they can be shared between the application and device
- The hardware devices provide a shareable abstraction, reducing the need for an OS device driver on each interaction



Application

Library (SPDK, OFED, openCL)

PCIe

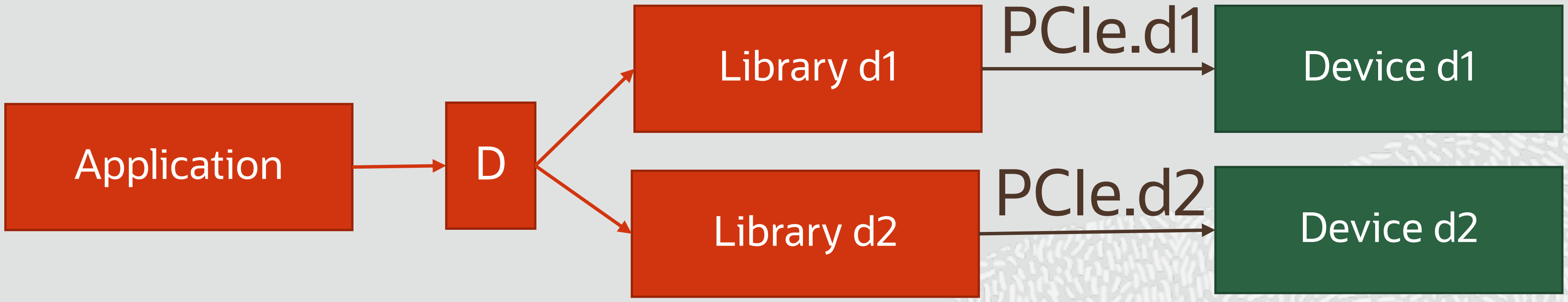
Device (NVMe, HCA, GPU)

Device Driver

Controller-Host Interfaces

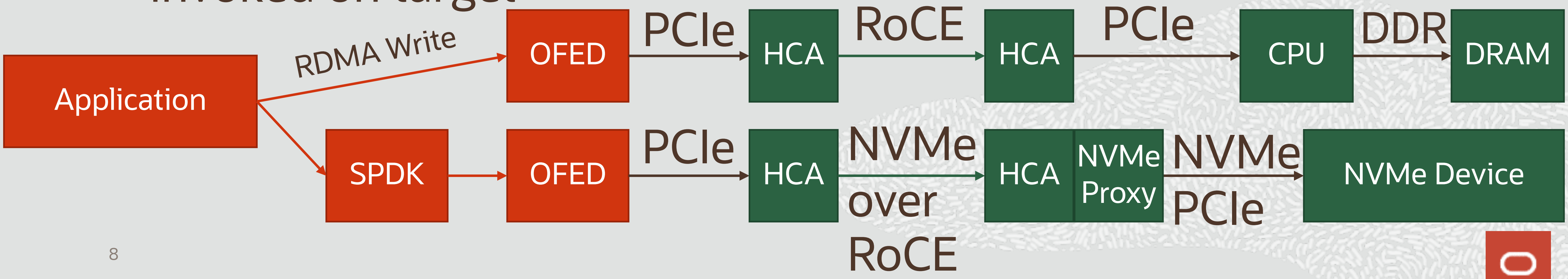
API standards are no longer enough…

- A standard Controller-Host Interface (like NVMe) allows a single library to be used for many similar devices
- Proprietary Controller-Host Interface and provide their own library variant that communicates with their device (openCL, ibverb)



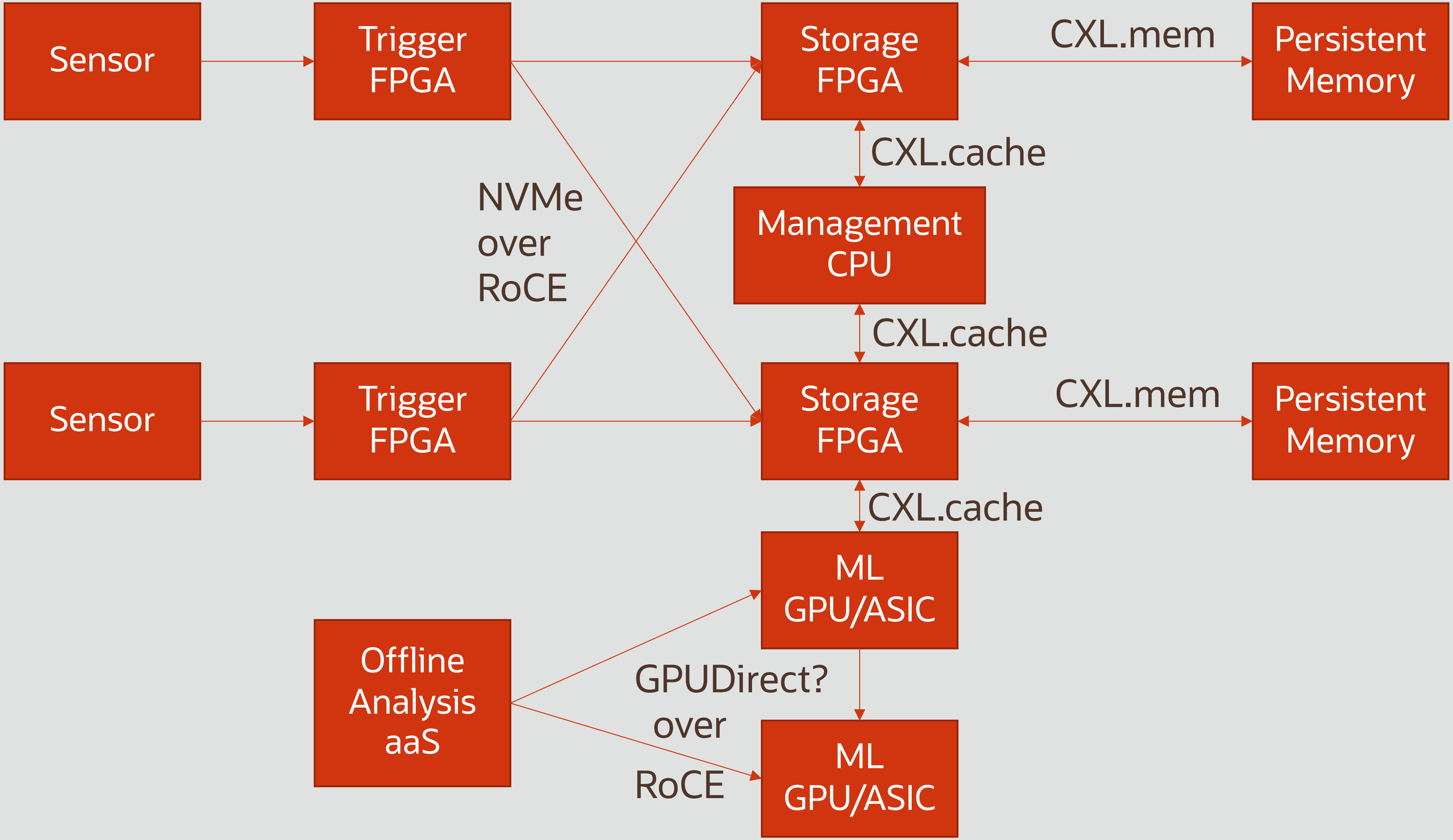
Proxied Transport: RDMA Write and NVMe over RDMA

- Accessing a remote target requires a remote proxy to access the remote device
- The proxy can be implemented in the library but the server can be more efficiently implemented inside the HCA, which can be integrated with the target
- Call these operations “single-sided” as no software is invoked on target

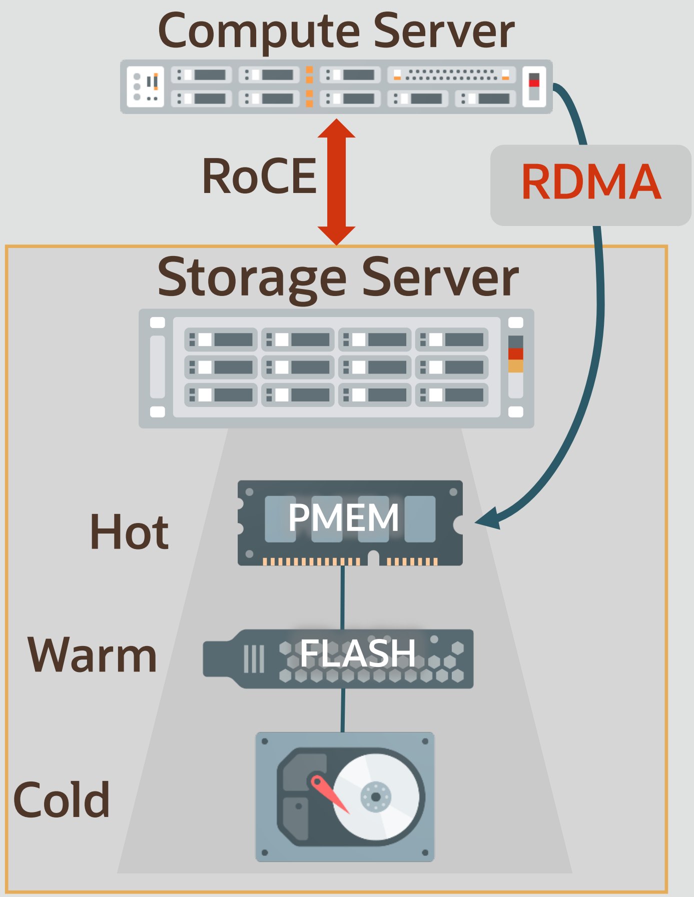


Accelerating Computations: Smart Storage and Smart Memory

- RDMA and NVMe provide simple read and write operations
- But we want to avoid data movement and issue higher level requests
  - ML/Analytics: Execute computation, return or forward results
  - Ingest/Access: Updates on a data structure
- Extend NVMe with Namespace-defined operations (ADTs) and capability-based security
- Allow clients and servers to run in HCA, CPU, Uncore, FPGA, GPU, ML Accelerator, Storage ASIC
- Leave complexity in the CPU, heavy lifting in Accelerator
  - E.g., access policy done in CPU, enforcement in Accelerator


Sensor
Trigger FPGA
Storage FPGA
CXL.mem
Persistent Memory
CXL.cache
NVMe over RoCE
Management CPU
CXL.cache
Sensor
Trigger FPGA
Storage FPGA
CXL.mem
Persistent Memory
CXL.cache
ML GPU/ASIC
Offline Analysis aaS
GPUDirect? over RoCE
ML GPU/ASIC

Exadata X8M *Persistent Memory Data Accelerator* adds Persistent Memory Storage Tier



- Exadata Storage Servers transparently add Persistent Memory Accelerator in front of flash memory

  World's First and Only Shared Persistent Memory Optimized for Database

- Database uses RDMA instead of I/O to read remote PMEM
  - Bypasses network and I/O software, interrupts, context switches
  - 10x better latency
  - 2.5x higher I/Os per second
- PMEM automatically tiered in front of flash and disk
  - Caching only hottest data increases effective capacity 10X
- PMEM RDMA also used to accelerate log writes up to 8x

The Technology Significance of Exadata

The 12+ years of Exadata evolution speak to our mindset and capabilities:

- Extreme performance and availability for *steady-state critical production workloads* – infrastructure grade
- Built with enterprise-grade *COTS components* (e.g., RoCE)
- Ever-increasing capabilities within the *same rack footprint*