Accelerating RocksDB with Eideticom’s NoLoad® NVMe-based Computational Storage Processor

Stephen Bates, CTO
Eideticom
B.C.S
(Before Computational Storage)
Let There Be Light!

Computational Storage Standardization

Birds of a Feather Meeting - FMS2018
What are we trying to achieve?

**Offload certain tasks from the host CPU**
- It can be more efficiently (pj/bit) to perform certain tasks via an accelerator/offload engine.
- Our biggest competitor is not other Computational Storage companies, it is a Xeon CPU.

**Reduce data movement**
- Reduce DMA traffic by moving offloads closer (on) the SSDs/HDDs/NV-DIMMs etc
- pj/bit
- May allow for less DRAM on host CPU (less capacity and/or less bandwidth (channels)).
- Reduces “DMA contagion”, the impact of large amounts of DMA traffic on application QoS
- Tied to frameworks like p2p_dma (see SDC2018 talk)

**Achieve the above in a vendor-neutral, standards-based way that we can write open-source software too**
- Leverage existing standards and ecosystems where possible.

Make Computational Storage “Consumable by Idiots”.
Plug in a CSx and your fans wind down...
None of this is new. Shoulders of Giants.
NoLoad® Computational Storage Processor (CSP)

Eideticom’s NoLoad® CSP
- Purpose built for acceleration of storage and compute intensive workloads

NVMe Computational Accelerators
- Compression, Encryption, Erasure Coding, Deduplication, Data Analytics, AI and ML

Consumable Computation Offload
- NVMe compliant, standards-based interface
- Leverages existing NVMe eco-system

Disaggregation
- Disaggregating compute and storage into independently scalable resources
**Why NVMe?**

- **Accelerators Require**
  - Low Latency
  - High Throughput
  - Low CPU Overhead
  - Multicore Awareness
  - QoS Awareness

- **NVMe Provides**
  - Low Latency
  - High Throughput
  - Low CPU Overhead
  - Multicore Awareness
  - QoS Awareness

- Why develop and maintain a driver when NVMe capabilities align so well with accelerator needs and you can have world-class driver writers working on your driver?
- **Real question is “Why not NVMe?”**
## NVMe for Computational Storage

<table>
<thead>
<tr>
<th>Present</th>
<th>Present as NVMe compliant device with multiple namespaces (one per accelerator).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td>Accelerators map to namespaces and are discovered using identify namespace command (vendor specific fields provide accelerator specific information)</td>
</tr>
<tr>
<td>Configure</td>
<td>Configure and initialize accelerator via in-situ data path configuration or offline configuration</td>
</tr>
<tr>
<td>Input</td>
<td>Input data and configuration transferred using NVMe Writes by writing to the namespace associated with accelerator</td>
</tr>
<tr>
<td>Output</td>
<td>Output data and status are retrieved using NVMe Reads by reading from the namespace associated with accelerator</td>
</tr>
</tbody>
</table>

- Works over PCIe or Fabrics.
- Management via NVMe-MI – customers love this!
- Security via NVMe features, TCG and OPAL.
- Long term what Eideticom does today morphs to a vendor-neutral open standard
NVMe for Computational Storage

Discovery

$ sudo nvme eid list

<table>
<thead>
<tr>
<th>Node</th>
<th>SN</th>
<th>Model</th>
<th>Namespace</th>
<th>Type</th>
<th>Format or SubType</th>
<th>FW Rev</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/nvme0n1</td>
<td>nvme1</td>
<td>Vendor</td>
<td>1</td>
<td>Conventional LBA</td>
<td>512 B + 0 B</td>
<td>1.0</td>
</tr>
<tr>
<td>/dev/nvme0cs2</td>
<td>nvme1</td>
<td>Vendor</td>
<td>2</td>
<td>Computation</td>
<td>libz Compression</td>
<td>1.0</td>
</tr>
</tbody>
</table>

NVMe already have good mechanisms for discovery for both PCIe and fabric attached namespaces. We can leverage that to discover which NVMe-based Computational Storage Devices are available to a host and what the Computational Storage Services provided by those Csxes do.

DISCLAIMER: The above nvme-cli output is just an illustrative example for discussion purposes.
NoLoad® CSP – Hardware Platforms

NoLoad® CSP U.2
- Standard U.2 NVMe form-factor: Utilizing SFF-8639 connector
- BittWare 250-U2

NoLoad® CSP Alveo
- Standard GPU form-factor: x16 PCIe
- Deployed on Xilinx Alveo U200, 250 or U280

NoLoad® CSP E1.S EDSFF
- Standard E1.S NVMe form-factor
- BittWare 250-E1.S Hardware
NoLoad® CSP - Software

- Management
  - nvme-cli
  - nvme-of
  - etc

- Applications
  - libnoload
  - SPDK

- User space
  - Both kernel and User space frameworks supported

- OS
  - No changes to OS
  - Use In-box NVMe drivers

- Hardware
  - NoLoad® CSP Hardware
A.C.S
(After Computational Storage)
End Solutions – Application Acceleration: RocksDB

**Bottom Line**
- 6x more transactions per sec
- 2.5x more efficient
- 4x reduced NAND costs
- Improved QoS

**Details**
- Eideticom’s NoLoad CSP
- Xilinx Alveo U280 (HBM)
- Dell R7425 PowerEdge server
- RocksDB
- Linux Operating System
- 2 NoLoad instances with compression offload
NoLoad® RocksDB NVMe Solution

- RocksDB integration required modifications to RocksDB source code to leverage services provided by NoLoad®.
- This consisted of under 100 lines of new code to tie RocksDB to NoLoad® via libnoload.
- Tasks like compression can now be offloaded to the CSP.
- It is also possible to use p2pdma to reduce data movement by DMAing data from the NoLoad CSP directly to the NVMe SSDs.
- Today a computational task requires two NVMe IO Commands. As we standardize this could drop to one command!
End Solutions – NVMe-oF

**Bottom Line**
- NoLoad Accelerators located in a remote server can be accessed by any client with a RDMA or TCP/IP connection.

**Details**
- Disaggregation of NoLoad Accelerators using NVMe-oF.
- NoLoad Accelerators identify as NVMe Namespaces, which can be accessed/shared using NVMe-oF.
- NoLoad Accelerators (compression, EC) shared across RDMA & TCP/IP.
- Broadcom BCM58800 SmartNIC.
Conclusions

• The large consumers of accelerators (hyperscalers) want vendor-agnostic, consumable interfaces, software and management stacks. NVMe gives them all that.
• Eideticom’s NoLoad CSP is the world’s first UNH certified NVMe-based CSP.
• Using NoLoad we can accelerate customers applications and filesystems.

This is all getting standardized so customers can enjoy a scalable, vendor agnostic framework for acceleration and computational storage!