



NVMe Based Reconfigurable Compression Engine

David Sloan

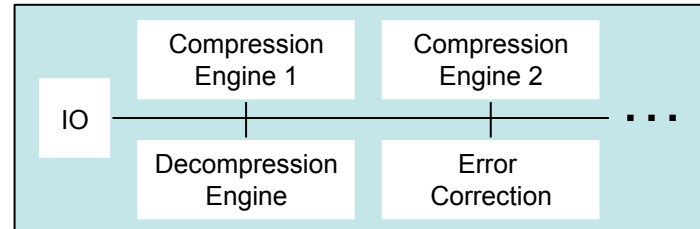
Flash Memory Summit 2018
Santa Clara, CA



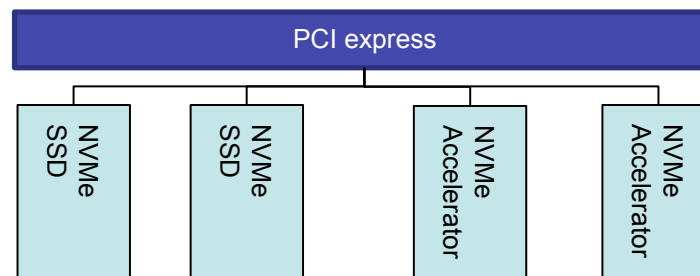


Overview

- Compression Engine for Reconfigurable Platform



- Why NVMe?





Flash Memory Summit

Compression Engine Targets

- **Standard compression output**
 - Must be decompressible without proprietary SW
- **Ditch the CPU**
 - Reduced CPU usage and higher data rates
- **Efficient design**
 - FPGA must be able to fit other modules



Compression: Output standard

- Deflate
 - Used in Zip, Gzip, and zlib
 - Zlib headers used
 - Static Huffman only
 - Raw data blocks are not used



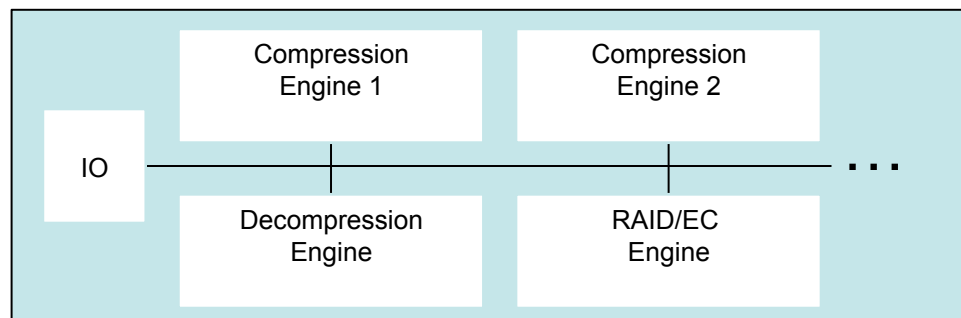
Compression: No CPU?

- Application specific HW can significantly outperform the CPU for well defined tasks
 - Protein Corpus: CPU* 58 MB/s, HW: 677 MB/s
- CPU is only used to manage data transfers to/from compression device and manage NVMe transfers

*CPU results from zlib level 1 on an Intel i5-6500@3.2 GHz

Compression: Shared FPGA

- Modular design allows for simplification of meeting specific needs
- Many compression cores or many different cores





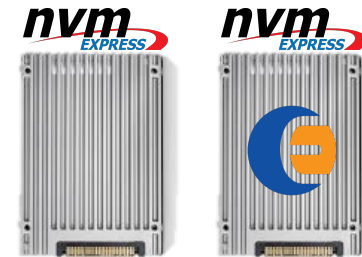
Compression: Large compressible files

Engine	calgary.1G		cal4k.1G	
	Compression Ratio	Throughput	Compression Ratio	Throughput
ZLIB-1 on CPU [2]	2.62	81 MB/s	29.56	340 MB/s
QAT-8955 [3]	2.60	1.46 GB/s	7.30	2.85 GB/s
Eideticom-H [2,4]	2.22	2.04 GB/s	35.81	2.97 GB/s
Eideticom-F [2,4]	2.12	2.19 GB/s	27.93	3.14 GB/s

1. Intel, "Programming Intel QuickAssist Technology Hardware Accelerators for Optimal Performance", April 2015, URL: https://01.org/sites/default/files/page/332125_002_0.pdf.
2. Tests were performed on a single core of an Intel i5-6500 @3.2GHz machine running Ubuntu 16.04.
3. Intel QuickAssist 8955 with six compression cores on it's ASIC chipset. All of the compression cores were used for this test [1].
4. FPGA test were performed on a NoLoad with three compression cores. The -H option provides higher compression while the -F option provides higher data throughput (~ same area)



NVMe



- High speed, CPU efficient standard
- In-box drivers
- Allows for use of peer to peer HW in PCIe
 - Drastically reduces system memory usage
 - More memory BW free for CPU compute tasks



Flash Memory Summit

NVMe: CPU load

- CPU compute tasks are completely offloaded to the NVMe Accelerator
- CPU utilization is determined primarily by transfer size
 - 32 kB → ~5 GB/s / physical CPU core*
 - 64 kB → ~10 GB/s / physical CPU core*

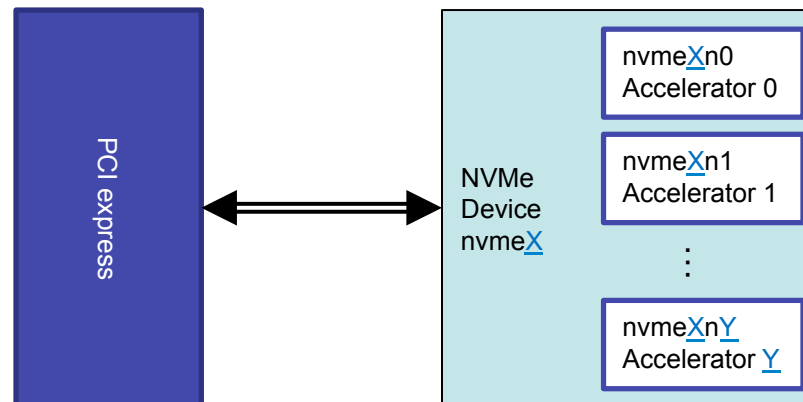
*CPU Intel i7-7800X@3.5 GHz

Flash Memory Summit 2018
Santa Clara, CA



NVMe Accelerator

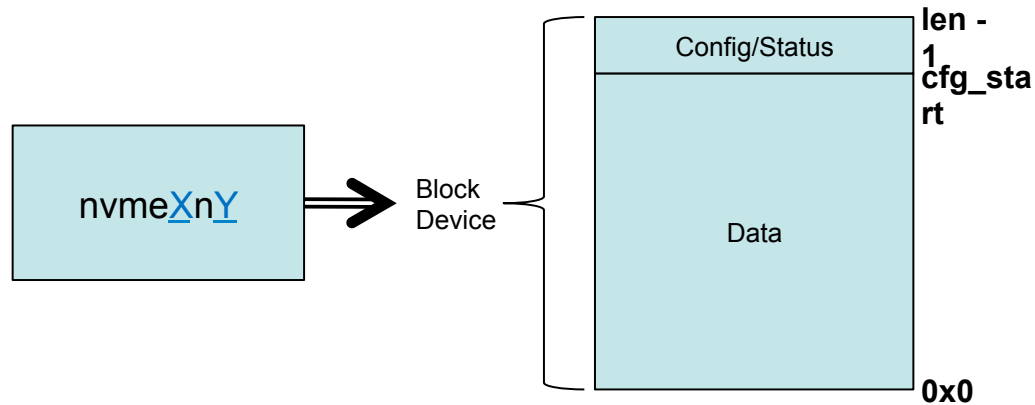
- Accelerator cores exposed as NVMe namespaces





NVMe Accelerator

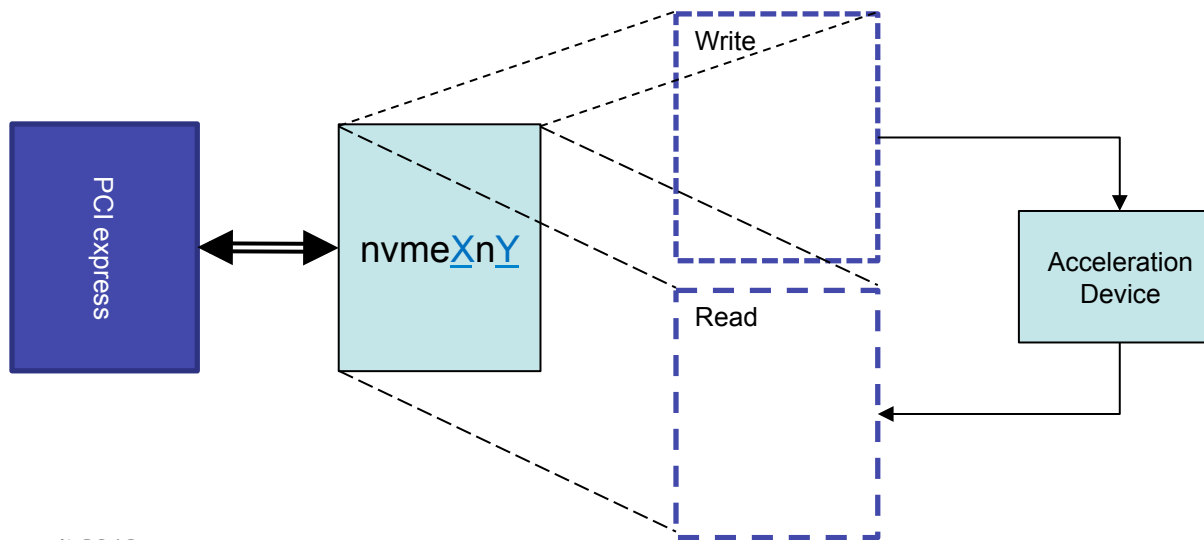
- Data and config operations are memory mapped to the block device





NVMe Accelerator

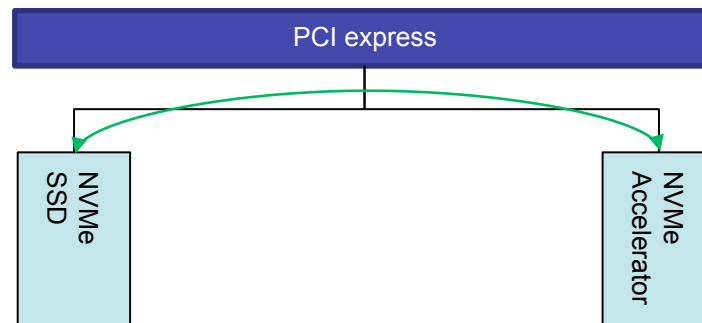
- Reads map to device output
- Writes map to device input





NVMe: Peer to Peer

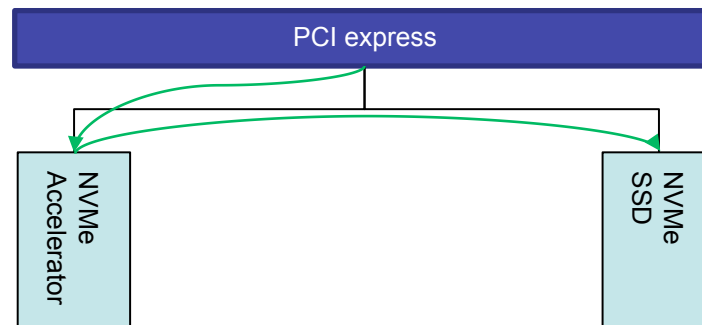
- PCIe allows for direct communication between NVMe devices
- Further reduces CPU overhead for offloaded acceleration





NVMe: Peer to Peer - CPU to Drive

- Jobs from CPU can be processed then stored directly on NVMe SSD without touching system memory





NVMe: Peer to Peer - Demo

- NVMe SSD input and output
- NoLoad NVMe Accelerator
- Both P2P and Standard Operation

