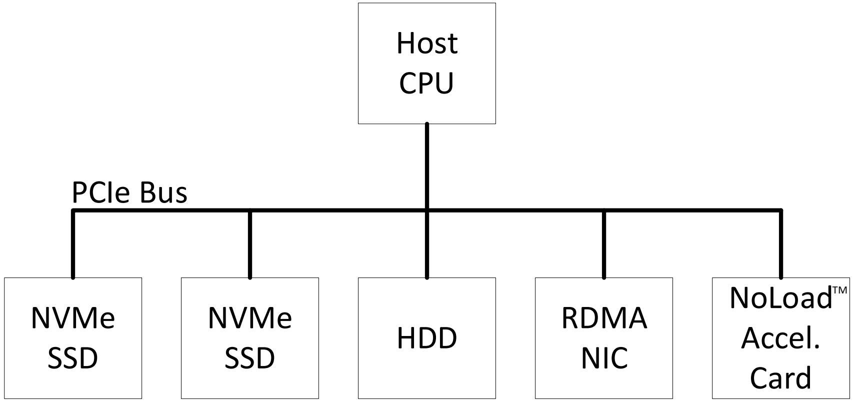
SDC 18
September 24-27, 2018
Santa Clara, CA
www.storagedeveloper.org

FPGA Accelerator Disaggregation Using NVMe-over-Fabrics

Sean Gibb, VP Software
Stephen Bates, CTO
EIDETICOM

Acceleration

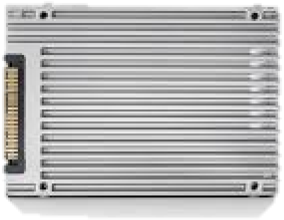


- Storage I/O Bandwidth rapidly increasing
- Storage workloads taxing on host CPU
- FPGAs provide a compelling solution for storage workloads
- NoLoad = NVMe Offload

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

NoLoad Accelerator

NoLoad Bitfiles



U.2 FPGA Card

COTS PCIe FPGA Card

Cloud Servers i.e. Amazon F1

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

Why NVMe?

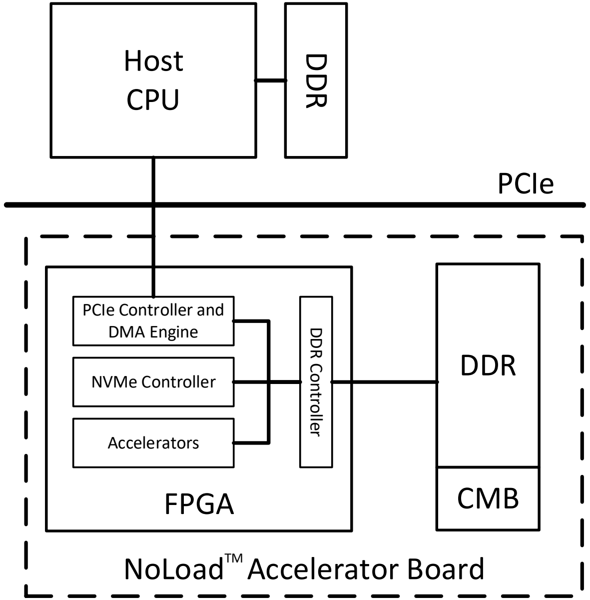
- Accelerators require:
  - Low latency
  - High throughput
  - Low CPU overhead
  - Multicore awareness
  - QoS awareness

- Accelerators require:
  - 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?”

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

Accelerator and Controller Architecture



- Host CPU communicates with accelerators via NVMe controller using standard NVMe commands
- NVMe controller pushes and pulls commands and data via DMA engine
- NVMe controller is in-house developed soft controller on a RISC-V
- Board has external DDR for accelerators that require large data storage
- Controller supports command queue and data CMB (using portion of DDR)
- Developed an accelerator wrapper to handle details of NVMe

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

NVMe for Accelerators

- Presents as NVMe 1.3 device with multiple namespaces
  - One namespace per accelerator
- Accelerators map to namespaces and are discovered using identify namespace command
  - Vendor specific fields provide accelerator specific information
- Configuration using in-situ data path configuration or vendor specific command
- Input data and in-situ configuration are transferred using NVMe Writes to the namespace associated with the accelerator
- Output data and in-situ status are transferred using NVMe Reads to the namespace associated with the accelerator

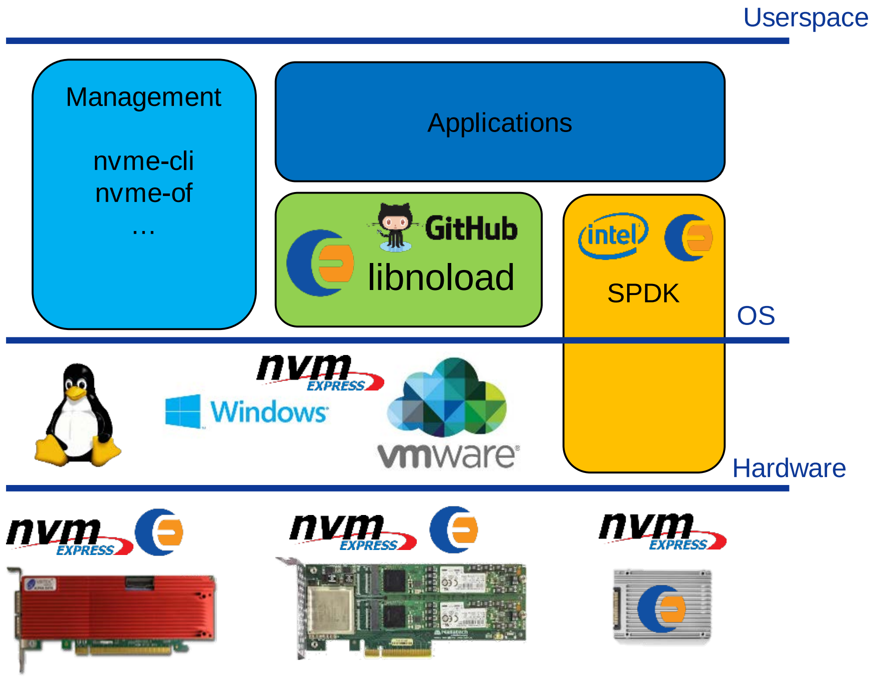
2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

NVMe for Accelerators

- In-house NVMe controller supports advanced features including queue and data CMB, SGL and NVMe-oF
- Also support peer-to-peer (P2P) operation
- No customized drivers required – all inbox drivers!
- Leverage industry-standard NVMe test tools
  - FIO and nvme-cli
  - Assist with deployment and benchmarking
- Take advantage of rich NVMe ecosystem
  - Can leverage servers and storage systems developed for NVMe

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

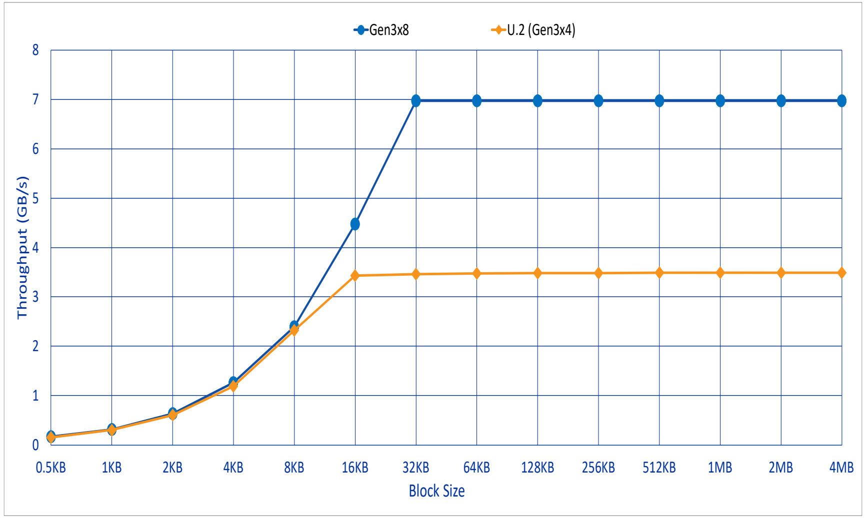
libnoload



- Developed user API to assist with common tasks associated with acceleration over NVMe
- Provides C and C++ libraries
- Handles discovering NoLoad adapters and enumerating accelerators on the adapters
- Provides support to lock/unlock accelerators
- Provides thin wrappers over system calls for writing data to and reading results back from the accelerators
- Handles seamless integration with our accelerator interface IP
- API is BSD licensed and available via our public github

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

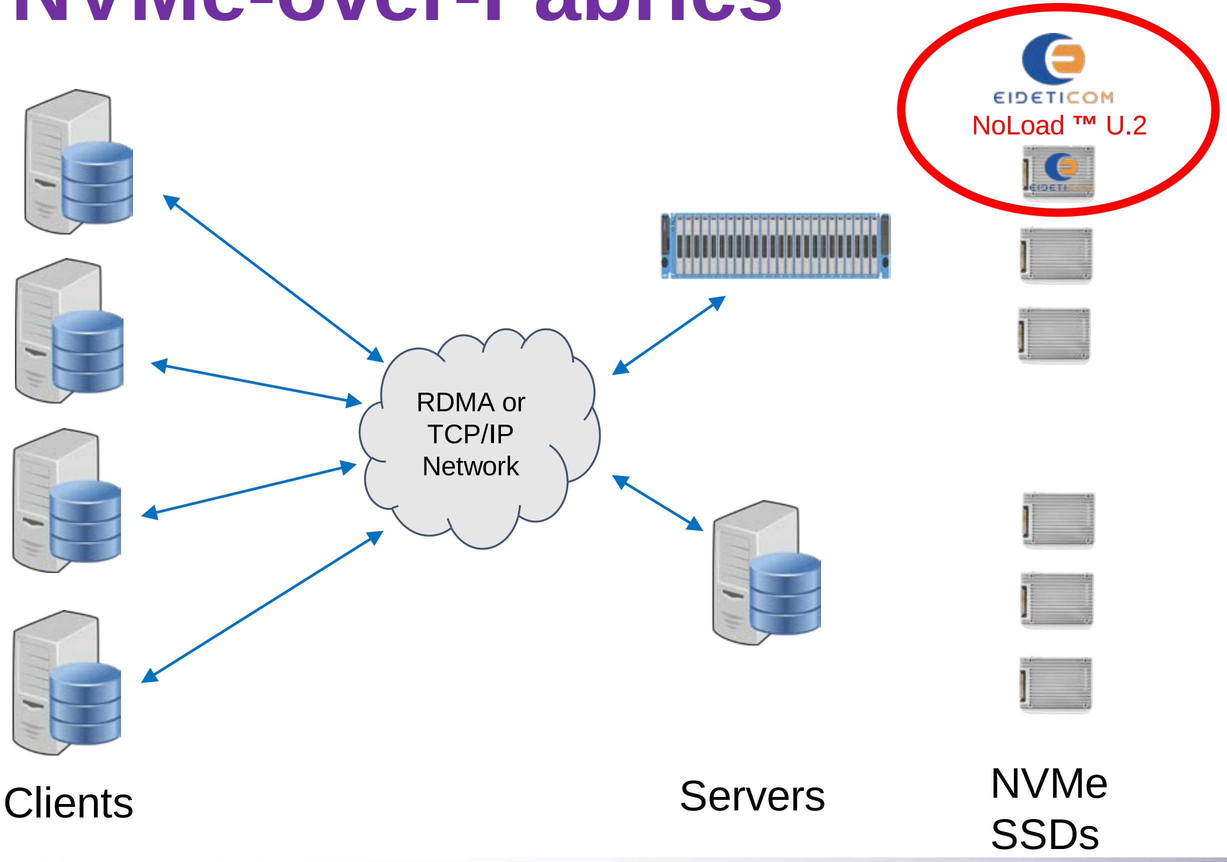
Controller Performance



- Results for single RISC-V core controller implementation
- Saturating bus for ≥32 kB block transfer for Gen3x8 (COTS FPGA card)
- Saturating bus for ≥16 kB block transfer for Gen3x4 (U.2 form factor)
- Focus to date has been on accelerators with ≥16 kB block sizes (i.e. EC, compression)
- Working on multicore RISC-V system that drastically improves small block performance

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

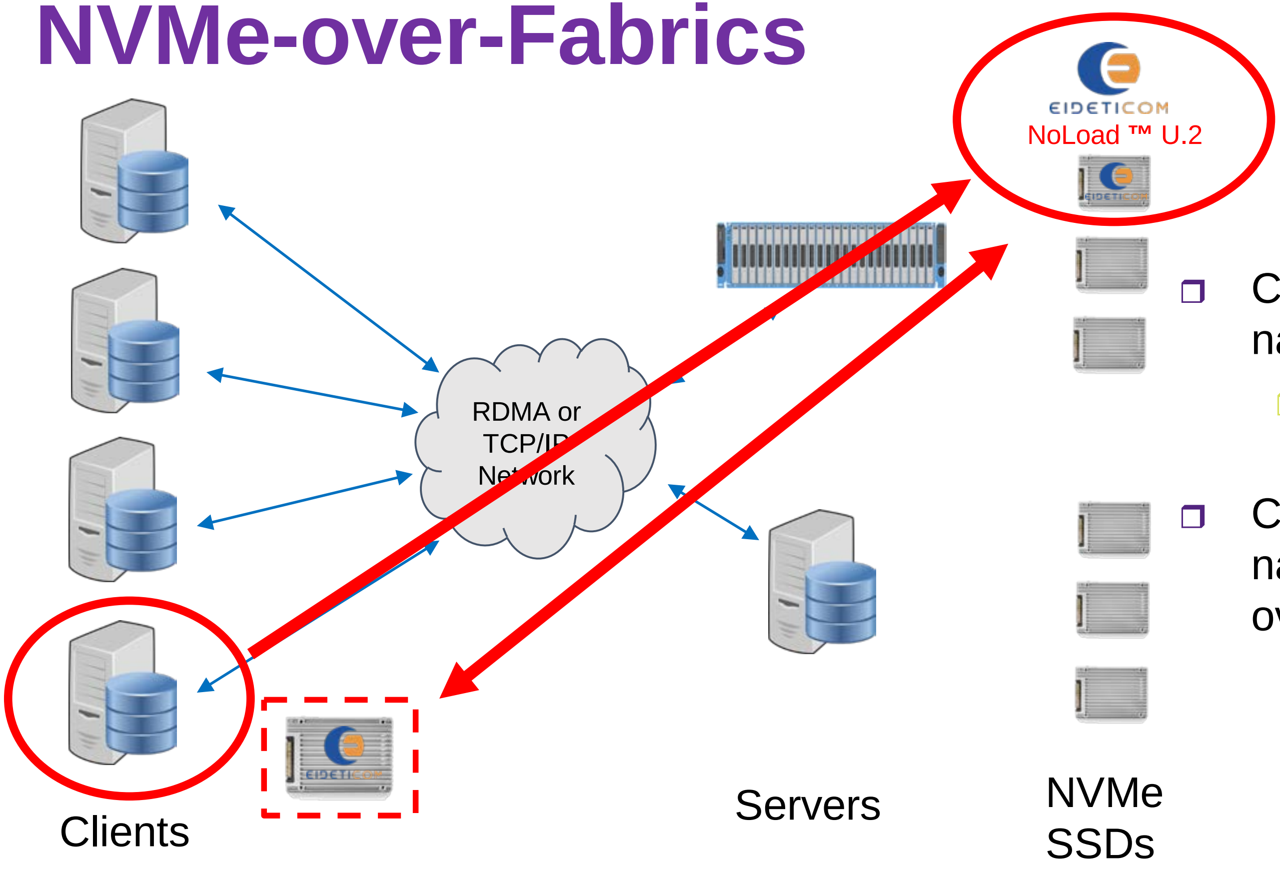
NVMe-over-Fabrics



- NVMe-over-Fabrics (NVMe-oF) allows namespaces to be shared across networks
- Expose NVMe namespaces to client machines using inbox drivers
- NoLoad is a standard namespace:
  - Can share it in the same way as any other NVMe device

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

NVMe-over-Fabrics



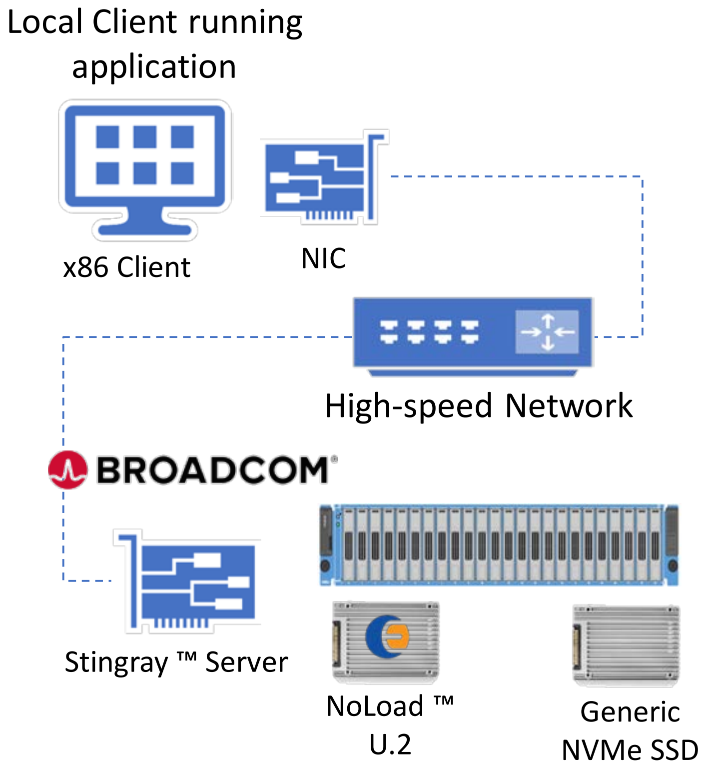
- Clients request to borrow namespace(s) from server
  - Recall that accelerators map to namespaces
- Client given access to the namespace (aka accelerator) over the connection

NVMe-over-Fabrics

```
amaier@dionysus:~$ sudo nvme list
Node             SN                   Model                                    Namespace Usage                      Format           FW Rev
---------------- -------------------- ---------------------------------------- --------- -------------------------- ---------------- --------
/dev/nvme1n1     2018-03-05-0001      Eideticom NoLoad Accelerator Alpha       1           0.00   B /   2.15  GB    512   B +  0 B   1.7.2719
/dev/nvme1n2     2018-03-05-0001      Eideticom NoLoad Accelerator Alpha       2           0.00   B /   2.10  MB    512   B +  0 B   1.7.2719
/dev/nvme1n3     2018-03-05-0001      Eideticom NoLoad Accelerator Alpha       3           0.00   B /   8.59  GB    512   B +  0 B   1.7.2719
/dev/nvme1n4     2018-03-05-0001      Eideticom NoLoad Accelerator Alpha       4           0.00   B /   8.59  GB    512   B +  0 B   1.7.2719
/dev/nvme1n5     2018-03-05-0001      Eideticom NoLoad Accelerator Alpha       5           0.00   B /   8.59  GB    512   B +  0 B   1.7.2719
/dev/nvme1n6     2018-03-05-0001      Eideticom NoLoad Accelerator Alpha       6           0.00   B /   8.59  GB    512   B +  0 B   1.7.2719
/dev/nvme2n1     289d9c51523fb07c     Linux                                    1         750.16  GB / 750.16  GB    512   B +  0 B   4.14.49-
/dev/nvme2n2     289d9c51523fb07c     Linux                                    2          97.00  GB /  97.00  GB    512   B +  0 B   4.14.49-
```

- Clients see newly acquired namespaces as local NVMe block devices
- Normal NVMe operations can be executed as if resources local attached to client machine

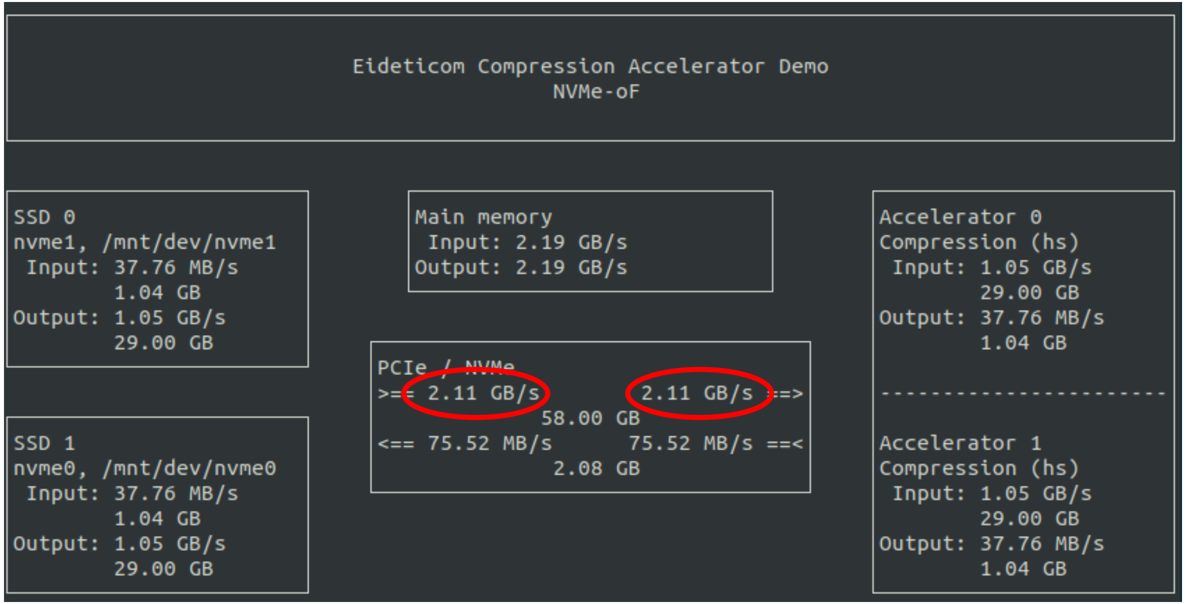
Case Study: Compression-over-Fabrics



- Demonstrate GZIP compression-over-fabrics
  - Both RoCE and TCP/IP networking
- Both NoLoad and generic NVMe SSD located on remote server
- U.2 accelerator form factor (Gen 3x4)
- Local client running the application is unaware that it is using an over-Fabrics acceleration device
  - User space code is exactly the same direct attach, over-Fabrics, or peer-to-peer

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.
EIDETICOM

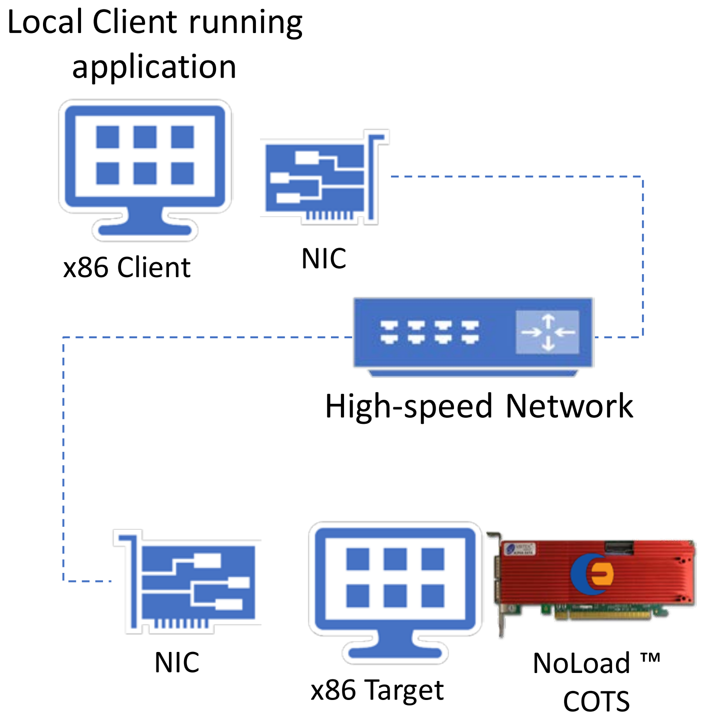
Case Study: Compression-oF Results



- Compression over RoCE or TCP/IP attains same throughput as direct attach
- Fabrics latency is hidden by having multiple compression operations in flight
- More about impact on target machine to follow

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

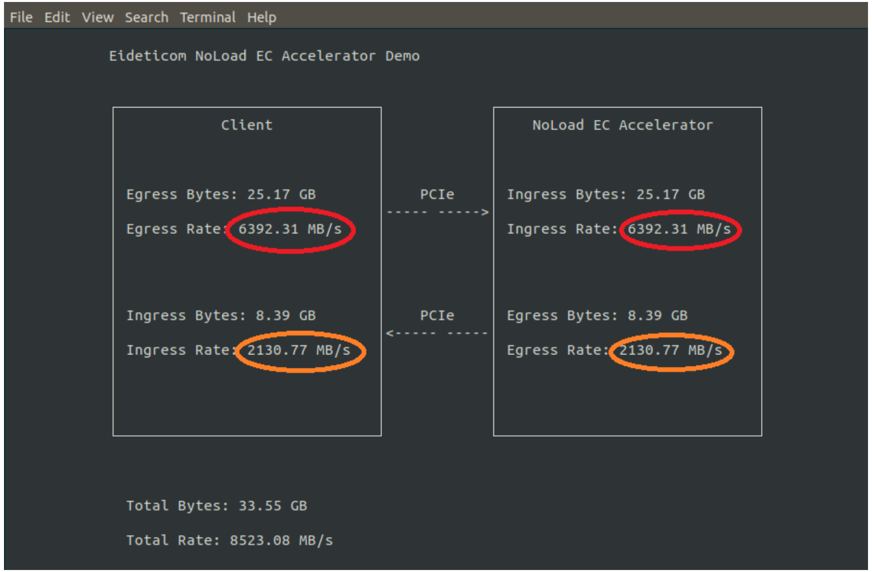
Case Study: EC-over-Fabrics



- ISA-L compatible Reed-Solomon based EC over RoCE and TCP/IP
- Supports up to 32+4 disk groups with block sizes ranging from 16kB to 128kB
- Gen3x16 accelerator form factor
- Both NoLoad and generic NVMe SSD located on remote server
- User space code is exactly the same direct attach, over-Fabrics, or peer-to-peer

SDC 18
2018 Storage Developer Conference. © Eideticom. All Rights Reserved.
EIDETICOM

Case Study: EC-oF Results



- Client software is not aware that it is performing EC over fabrics connection
- Results have a small latency penalty vs direct attach results

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

Mitigating Target Impact (CPU)

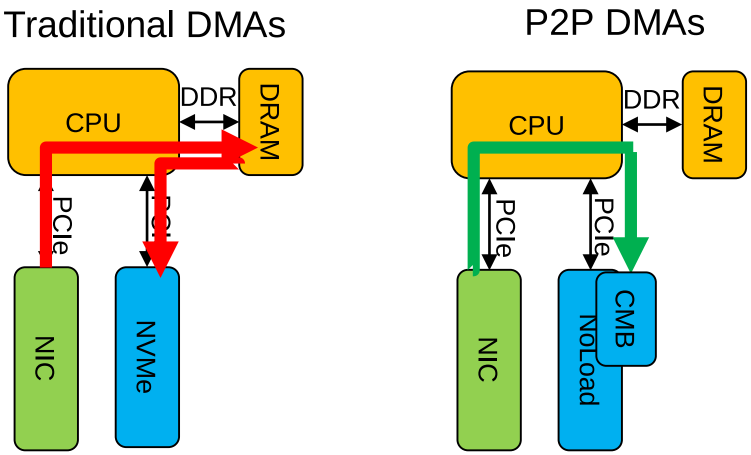
- What is the impact on resources in target machine?
  - All transfers flow into and out of DDR on CPU
  - Target CPU has to process all commands
  - How to mitigate?
- NVMe-oF Offload allows the NIC to directly connect to NVMe devices
  - Using Mellanox ConnectX-5 can offload the NVMe work from the target CPU

Operation	Latency (read/write) us	CPU Utilization	CPU Memory Bandwidth	CPU PCIe Bandwidth	NVMe Bandwidth	Ethernet Bandwidth
Vanilla NVMe-oF	188/227	1.00	1.00	1.00	1.00	1.00
ConnectX-5 Offload	128/138	0.02	2.40	1.03	1.00	1.00

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

Mitigating Target Impact (Memory)

- NVMe CMB (Controller Memory Buffer) is a PCIe BAR that can be used for Submission and Completion Queues, PRPs, SGLs, and data
- PCIe drivers can register memory or request access to memory for DMA
- P2P framework called p2pmem is being proposed for Linux kernel
- P2P DMA allows us to bypass CPU DRAM



2018 Storage Developer Conference. © Eideticom. All Rights Reserved.

Processor Offload

- NVMe Offload + p2pmem = Big Savings

Operation	Latency (read/write) us	CPU Utilization	CPU Memory Bandwidth	CPU PCIe Bandwidth	NVMe Bandwidth	Ethernet Bandwidth
Vanilla NVMe-oF	188/227	1.00	1.00	1.00	1.00	1.00
ConnectX-5 Offload	128/138	0.02	2.40	1.03	1.00	1.00
Eideticom NoLoad p2pmem	167/212	0.55	0.09	0.01	1.00	1.00
CX5 Offload + Eideticom NoLoad p2pmem	142/154	0.02	0.02	0.04	1.00	1.00

2018 Storage Developer Conference. © Eideticom. All Rights Reserved.