Real-World Performance Advantages of NVDIMM and NVMe: A Case Study with OpenZFS

Nick Principe
iXsystems

Twitter: @nickprincipe
Github: @powernap
Email: nap@ixsystems.com
Agenda

- Flash and Persistent Memory Devices
  - Which aspect of performance matters most?
  - Device Performance Survey
- Real-world example: OpenZFS SLOG Device
  - Intro to OpenZFS: What is a SLOG?
  - SLOG Type Performance Survey
Which Aspect of Performance Matters Most?

- Comparing flash and persistent memory devices to each other
  - Synchronous write latency is a key differentiator
- But don’t forget maximum MiB/s!
- Mostly controller or interconnect limited
Storage Interconnect Performance
Why We Care About NVMe

Unidirectional Peak Interconnect Performance

Dual-ported NVMe = x2 lanes per-controller
Storage Interconnect Performance
Why We Care About NVDIMM

Unidirectional Peak Interconnect Performance

<table>
<thead>
<tr>
<th>Interconnect Type</th>
<th>MB/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>SATA-1</td>
<td>150</td>
</tr>
<tr>
<td>SATA-2</td>
<td>300</td>
</tr>
<tr>
<td>SAS-1</td>
<td>300</td>
</tr>
<tr>
<td>SATA-3</td>
<td>600</td>
</tr>
<tr>
<td>SAS-2</td>
<td>600</td>
</tr>
<tr>
<td>SAS-3</td>
<td>1,200</td>
</tr>
<tr>
<td>NVMe 3.0 x2</td>
<td>1,969</td>
</tr>
<tr>
<td>NVMe 3.0 x4</td>
<td>2,400</td>
</tr>
<tr>
<td>DDR4-2666</td>
<td>3,938</td>
</tr>
</tbody>
</table>
Synchronous Write Performance

- Why are sync writes interesting? (choose two)
  - Fast
  - Safe
  - Cheap
- Fast + safe: usually, power-fail-safe device cache in use
  - HDDs are slow
  - Write is painful for NAND Flash, too
- With Flash, it’s easier for read to be “fast enough”
  - Usually limited by controller or interconnect
Synchronous Write Performance Testing Methodology

- Used `diskinfo -wS` on various FreeBSD hosts
- Quick single-threaded sync write test
- Not incredibly scientific:
  - Different hosts
  - Devices tested range from new to well-aged
- Despite this, gives us some ballpark numbers to work with
Synchronous Write Performance Testing
HDD-Based Storage

- Pure HDD ~= 10s of ms
- Hybrid SSHD ~= 1ms for small I/O

Keep the representative samples:
- 10TB HDD
- SSHD
Synchronous Write Performance Testing
Consumer SATA SSD

- Consumer SATA SSDs ~ 1ms
- May not be power-fail-safe!
- Cheap + Fast
- Keep the new representative sample:
  - Crucial MX300
Enterprise SATA SSD ~= 1ms
A bit slower than consumer
Power-fail-safe
Enterprise SAS SSDs ~= 0.1 ms
Keep the new representative sample:
HGST SS300
Synchronous Write Performance Testing

Consumer NVMe

- Consumer NVMe same as normal SATA Flash \(\approx 1-2\) ms

- Keep the new representative sample:
  - 250G 960 EVO
Prosumer and Enterprise NVMe beats Enterprise SAS Flash $\approx$ 20-30 us

Keep the new representative sample:
- 1.2T Intel 750
Early (small) Optane devices are interesting.

- Range from 10s of us to 10s of ms

- Keep the new representative sample:
  - 32G Optane
Synchronous Write Performance Testing

Newer Optanes more closely match or exceed Enterprise NVMe NAND Flash

Keep the new representative sample:

- 480G Optane 900p
NVDIMM is nearly an order of magnitude faster than Optane

- 400k IOPS
- 2.5 us
Synchronous Write Performance Testing
HDD-Based Storage

- Pure HDD \(~= 10\text{s of ms}\)
- Hybrid SSHD \(~= 1\text{ms for small I/O}\)

- Keep the representative samples:
  - 10TB HDD
  - SSHD
Consumer SATA SSDs ~= 1ms
- May not be power-fail-safe!
- Cheap + Fast
- Keep the new representative sample:
  - Crucial MX300
Enterprise SATA SSD ~= 1ms
- A bit slower than consumer
- Power-fail-safe

Enterprise SAS SSDs ~= 0.1 ms

Keep the new representative sample:
- HGST SS300
Synchronous Write Performance Testing
Consumer NVMe

- Consumer NVMe same as normal SATA Flash \(\approx 1-2\) ms

- Keep the new representative sample:
  - 250G 960 EVO
Prosumer and Enterprise NVMe beats Enterprise SAS Flash \(\approx 20-30\) us

Keep the new representative sample:
- 1.2T Intel 750
Early (small) Optane devices are interesting.

- Range from 10s of us to 10s of ms.

- Keep the new representative sample:
  - 32G Optane.
Synchronous Write Performance Testing
Newer Optane

- Newer Optanes more closely match or exceed Enterprise NVMe NAND Flash
- Keep the new representative sample:
  - 480G Optane 900p
NVDIMM is an order of magnitude faster than Optane.
Real-world example:
OpenZFS SLOG Device
OpenZFS Intro

- Adaptive Replacement Cache “ARC” uses system memory (DRAM) and stores:
  - All incoming data
  - “Hottest” data
- Level 2 ARC “L2ARC” is optional per-pool and is typically one or more Flash devices - it stores:
  - “Warm” data that does not fit into ARC
- ZFS Intent Log “ZIL” stores a copy of in-progress synchronous write ops
- Uses disks in the pool, OR
- Can specify optional Separate Log “SLOG” device(s)
OpenZFS Intro:

**ARC**

- “ARC” = Adaptive Replacement Cache
- Resides in system memory
- Shared by all pools
- Used to store/cache:
  - All incoming data
  - “Hottest” data
  - Metadata
- Balances cache between
  - Most Frequently Used
  - Most Recently Used
OpenZFS Intro:

L2ARC

- “L2ARC” = Level 2 Adaptive Replacement Cache
- Resides on one or more storage devices
  - Usually Flash
- Added to a single pool
  - Only services data held by that pool
- Used to cache:
  - “Warm” data and metadata that do not fit into ARC
OpenZFS Intro:

ZIL

- "ZIL" = ZFS Intent Log
- By default, ZIL resides on the data disks in the pool
- Used to quickly store synchronous write operations onto persistent storage
- Client request acknowledged once data logged to ZIL
- Data later written into pool from ARC via transaction group
OpenZFS Intro: SLOG

- "SLOG" = **Separate (ZFS Intent) Log**
- Optional SLOG resides on one or more storage devices
  - Flash or better
  - High endurance
  - Added to a single pool
  - Only services that pool
  - Allows ZIL to be separated from primary pool storage
  - Can improve performance
OpenZFS: Asynchronous Write

1. Write arrives at NIC/HBA
2. OpenZFS accepts write
   a. Data written to ARC
3. Write acknowledged to host
   a. Data only in ARC (RAM)
4. At next Transaction Group (TXG)
   a. Data written to data drives in pool
5. Data remains in ARC
   a. As Most Recently Used (MRU) data in cache
   b. No longer dirty
      i. Data is protected on disk
OpenZFS: Synchronous Write (With SLOG)

1. Write arrives at NIC/HBA
2. OpenZFS accepts write
   a. Data written to ARC
   b. **Data written to SLOG**
3. Write acknowledged to host
   a. **Data protected by SLOG**
4. At next Transaction Group (TXG)
   a. Data written to drives in pool
   b. Data in SLOG overwritten by a future TXG after this TXG
5. Data remains in ARC
   a. As Most Recently Used (MRU) data in cache
   b. No longer dirty
      i. Data is protected on disk
Real-World Example: OpenZFS
SLOG Device Comparison: Random 4 KiB

- Baseline is with sync=standard set
  - ZIL not used
- All other runs, sync=always
  - Force use of ZIL (on SLOG) for all writes
- Three thread counts measured
  - Each thread writes as fast as possible
Real-World Example: OpenZFS
SLOG Device Comparison: Sequential 128 KiB

- Baseline is with sync=standard set
- ZIL not used
- All other runs, sync=always
- Force use of ZIL for all writes
- SSD SLOG hits device write MiB/s limit
Real-World Example: OpenZFS
SLOG Device Comparison: Sequential 128 KiB

- Pool bandwidth can exceed the limits of a single SAS SSD SLOG
- Latency much higher
  - Sync Write to HDD ~= 10-30ms
  - Contention on HDDs (ZIL + Data storage)
- More concurrency needed with no SLOG
- Up to 16 threads/LUN shown
Questions?

Thanks to iXsystems for making this talk possible!

For more info on iXsystems Storage and Servers, see https://www.ixsystems.com

iXsystems is the corporate sponsor of FreeNAS! https://www.freenas.org