New Developments in Cloud Storage Acceleration Layer (CSAL)

CSAL, A Host Based FTL in SPDK

Kapil Karkra, Sr. Principal Engineer at Solidigm
Legal Disclaimers

All product plans, roadmaps, specifications, and product descriptions are subject to change without notice.

Nothing herein is intended to create any express or implied warranty, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, or any warranty arising from course of performance, course of dealing, or usage in trade.

The products described in this document may contain design defects or errors known as "errata," which may cause the product to deviate from published specifications. Current characterized errata are available on request.

© Solidigm. "Solidigm" is a trademark of SK Hynix NAND Product Solutions Corp (d/b/a Solidigm). "Intel" is a registered trademark of Intel Corporation. Other names and brands may be claimed as the property of others.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase.

Performance results are based on testing as of dates shown in the configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure.

Your costs and results may vary.

Some results have been estimated or simulated using internal Solidigm analysis or architecture simulation or modeling and provided to you for information purposes only. Any differences in your system hardware, software or configuration may affect your actual performance.

© 2023 Solidigm. All rights reserved.
Imagine if we had…

A sandbox to explore, add capabilities, and drive down data center TCO…
Agenda

1. TCO Benefits of large capacity (D5-P5536 61.44TB) QLC
2. How to Further Expand the Reach of QLC?
3. Creating an Easy Button with CSAL and a Reference Storage Platform (RSP)
4. Performance Results and TCO Benefits
5. Summary and Call to Action
Motivation and Problem
Wide Range of Use Cases/Customers Adopting QLC

**Example: Alibaba Local Disk Use Case**

- Alibaba replaced HDDs with Solidigm’s QLC D5-P5316 QLC SSDs in their 3rd generation big data local disk ECS instances to double the performance vs. 2nd generation while holding the price to their customers constant.
- TCO was the same between the two generations
  - While the CAPEX was higher, the 2x density led to rack tax (building, personnel, land, etc.) and OPEX savings to offset the higher CAPEX.

Alibaba local disk use case is a great proof point of QLC successfully replacing HDDs

Collaboration with Alibaba, as one of the foundational QLC customers, resulted in co-development of CSAL

Alibaba uses Optane SSDs in their local disk use case. What can we replace Optane SSDs (P5800X) with?
D7-5810 as Optane SSD (P5800X) Replacement for O+Q Deployments

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Worst-case Use Case Demands</th>
<th>P5800X (Optane) Supplies</th>
<th>P5810 (SLC) Supplies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endurance</td>
<td>37 DWPD</td>
<td>100 DWPD</td>
<td>50 DWPD</td>
</tr>
<tr>
<td>rand 4k write</td>
<td>8 KIOPS</td>
<td>20 KIOPS</td>
<td>10 KIOPS</td>
</tr>
</tbody>
</table>

Based on Solidigm internal analysis

![SLC BW Utilization By CSAL Activities](image.png)

D7-P5810 is engineered to provide best cost-performance-endurance balance to replace P5800X in O+Q deployments

© 2023 Solidigm. All rights reserved.
Even Greater TCO Benefits with 61.44TB P5336 and P5810 SSDs

<table>
<thead>
<tr>
<th>Config</th>
<th>8xOptane + 8xQLC Optane = 400GB QLC = 15.36TB</th>
<th>8xSLC + 8xQLC SLC= 800GB QLC = 61.44TB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1x Drive capacity (GB)</td>
<td>15360</td>
<td>61440</td>
</tr>
<tr>
<td>Total storage cap per node (TB)</td>
<td>128</td>
<td>442</td>
</tr>
<tr>
<td>Incremental CAPEX for compute (vCPU + host DRAM) and storage (SLC + QLC)</td>
<td>base</td>
<td>+$18K</td>
</tr>
<tr>
<td>OPEX per node (5 years)</td>
<td>base</td>
<td>- ($0.5K)</td>
</tr>
<tr>
<td>Data center tax per node (5 years)</td>
<td>base</td>
<td>- ($0.3K)</td>
</tr>
<tr>
<td>Virtual drive capacity (GB)</td>
<td>16000</td>
<td>16000</td>
</tr>
<tr>
<td>Virtual drives per node</td>
<td>8</td>
<td>27</td>
</tr>
<tr>
<td>% TCO savings per virtual drive (5 years)</td>
<td>base</td>
<td>2.5x</td>
</tr>
</tbody>
</table>

Applied CSAL to Solidigm’s D5-P5336 61.44TB QLC and D7-P5810 SSDs.

How can we bring these TCO benefits of hyper-dense QLC to everyone?

Open-source CSAL is part of the answer!
Solution

Cloud Storage Acceleration Layer (CSAL)
Refresher from last SDC: What is Cloud Storage Acceleration Layer (CSAL)

**What is CSAL?**

- Open-source cloud-scale shared-nothing Flash Translation Layer (FTL bdev) in Storage Performance Development Kit (SPDK)
- Ultra fast cache and write shaping tier to improve performance and endurance to scale QLC value
- Consistent performance in multi-tenant environment
- Flexible scaling of NAND performance and capacity to the user/workload needs

### Functionality SLC Provides in CSAL?

<table>
<thead>
<tr>
<th>Region</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Region (C)</td>
<td>• Write shaping to adapt large IU/ZNS/FDP to 4k block interface.</td>
</tr>
<tr>
<td></td>
<td>• Boosts perf of workloads with temporal locality, also extends QLC endurance.</td>
</tr>
<tr>
<td></td>
<td>• Write buffering to absorb bursts, early write completion.</td>
</tr>
<tr>
<td></td>
<td>• Full-stripe RAID cache for RAID backends</td>
</tr>
<tr>
<td>L2P Region (L)</td>
<td>• DRAM cost savings with paged L2P</td>
</tr>
<tr>
<td>FTL Metadata Region (M)</td>
<td>• Crash consistency</td>
</tr>
<tr>
<td></td>
<td>• FTL consistency and TTR after power failure</td>
</tr>
<tr>
<td></td>
<td>• Superblock</td>
</tr>
</tbody>
</table>
What Changed in CSAL since last SDC

- CSAL open sourced (SPDK v22.09)
- Solidigm acquired CSAL team (Feb. 2023)
- New CSAL capabilities:
  - SLC as Optane Replacement
  - Mitigated in CSAL the need for VSS for crash consistency and power fail safety
  - RAID5F (in progress)
  - ZNS/FDP (in progress)
CSAL’s Core Capabilities Expand the QLC Benefits

**Capability #1:** Write shaping to enable a reduced DRAM footprint with Large IU drives and SLC caching tier can provide additional ~2x endurance and perf benefit for locality workloads vs. TLC

<table>
<thead>
<tr>
<th></th>
<th>1xTLC</th>
<th>1xSLC + 1xQLC</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>64K rand write zipf 1.2</td>
<td>1875</td>
<td>3317</td>
<td>MiB/s</td>
</tr>
</tbody>
</table>

Please see Test Configuration #2 under Sources, References and Test configs section on slide 22

**Capability #2:** CSAL tiered arch enables full-stripe RAID5F ~2x more efficient than traditional RAID5 to improve system fault tolerance

\[
\frac{raid5fWritePerf}{raid5WritePerf} = \frac{(N-1) \times \text{diskWritePerf}}{N \times \left(\frac{\text{diskWritePerf}}{\text{diskReadPerf}} + 2\right)} \approx 2
\]

**Capability #3:** CSAL enables pooling a large QLC capacity that can be shared across multiple cloud tenants to increase capacity and performance utilization.

**Capability #4:** CSAL writes sequentially to QLC to adapt to emerging interfaces e.g., ZNS to a regular 4k block, and enable multi-tenant isolation.

CSAL has key abstractions to extend the use of high-density QLC NAND and adapt the emerging interfaces (ZNS/FDP) to 4k block interface

In addition to CSAL, we are taking a community-driven approach to create a Reference Storage Platform (RSP) for everyone
Solution

Reference Storage Platform (RSP)
Why A Reference Storage Platform & Community-driven Approach?

- Open-source building blocks are a complex mix of parts, often challenging to assemble.
- This hampers rapid development, assessment, and deployment of storage technologies.
- A Reference Storage Platform brings it all together into a turnkey solution.
- A community-driven approach enables faster innovation, transparency, and easy evaluation and adoption of "part" technologies inside a unified "whole" solution.

The first instantiation of the reference platform already done!
Thanks to our Reference Storage Platform (RSP) partners, we have created the first instantiation of an open-source Reference Storage Platform sandbox.

Reference Storage Platform provides an easy button:
- SPDK NVMe-oF TCP target packaged in a turnkey VM image
- SPDK NVMe-oF TCP target packaged in a turnkey Container
- GUI to manage a pool of high-density NAND
- Reference hardware platform is an off the shelf commodity server from typical OEMs (Dell and Supermicro to start) with Intel CPUs.
- Getting started guide on spdk.io and Solidigm’s website

Next, we used this reference storage platform to illustrate the benefit of our hyper-dense QLC P5336 61.44 TB.
Solution
Performance Results and TCO Benefits
Disaggregated CSAL+SLC+QLC Perf/TCO vs. Incumbent TLC

<table>
<thead>
<tr>
<th></th>
<th>10x TLC</th>
<th>3xSLC + 7xQLC</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total available capacity</td>
<td>138</td>
<td>344</td>
<td>TB</td>
</tr>
<tr>
<td>Raw aggregate R/W BW</td>
<td>14</td>
<td>10.3</td>
<td>GB/s</td>
</tr>
<tr>
<td>Network Bound</td>
<td></td>
<td>11.4</td>
<td>GB/s</td>
</tr>
<tr>
<td>vDrive Capacity</td>
<td></td>
<td>16000</td>
<td>GB</td>
</tr>
<tr>
<td>vDrive Count per server</td>
<td>8</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Min(Demanded, Delivered) Perf/vDrive</td>
<td>500</td>
<td>500</td>
<td>MiB/s/16TB</td>
</tr>
</tbody>
</table>

Both TLC and QLC saturate the 80% of the 100Gbps network, but CSAL+SLC+QLC does it with 35% better TCO with D5-P5336.

The 35% TCO gain is attributable only to greater density; disaggregation, caching, raid5f, ZNS/FDP, etc. capabilities further the TCO reduction…

Based on Solidigm internal analysis
Solution
Future and Roadmap
CSAL and Reference Storage Platform (RSP) Add Capabilities Over Time

- CSAL: ZNS, FDP, RAID, ... (including Multi-tenancy, QoS)
- RSP: More Use cases, More Platforms, ... (including More Architectures)

© 2023 Solidigm. All rights reserved.
Summary and Call to Action

Summary:

- **Solidigm’s D7-P5810** provides optimal balance of cost, performance, and endurance to replace P5800X Optane SSDs and enable O+Q use cases
- CSAL has the necessary host-based FTL abstractions you need for the emerging SSD interfaces and high-density NAND
- We have provided a turnkey open-source disaggregated NVMe-oF TCP target as our first instantiation of the Reference Storage Platform
- CSAL and Reference Storage Platform provide an easy way to adopt new technologies
- Continuing technology additions drive TCO lower

Call to Action:

- Develop CSAL with us in the SPDK open source
- Use the NVMe-oF target we’ve provided and see the great things you can do with it.
- Add your own capabilities and create more reference implementations
- Let’s grow this community for the benefit of the entire storage industry!

We invite you to play with us in the sandbox!
Please take a moment to rate this session.

Your feedback is important to us.
Sources and References
Back up content supporting the main presentation
Sources and References

1. CSAL whitepaper – A Media-Aware Cloud Storage Acceleration Layer (CSAL)
2. CSAL+ZNS presentation – Zoned Storage in the Cloud
3. CSAL solution brief – A CSAL-Based Reference Storage Platform
4. Reference Storage Platform – Main Download Page
5. CSAL SDC 2022 Presentation – Enabling Unprecedented Perf and Capacity with Optane and QLC Flash

Author Contact: kapil.karkra@solidigm.com
## Test Configuration #1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BIOS Version</strong></td>
<td>1.4b</td>
</tr>
<tr>
<td><strong>OS</strong></td>
<td>Fedora 37 (Server Edition)</td>
</tr>
<tr>
<td><strong>Kernel</strong></td>
<td>6.3.8-100.fc37.x86_64</td>
</tr>
<tr>
<td><strong>CPU Model</strong></td>
<td>Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz</td>
</tr>
<tr>
<td><strong>NUMA Node(s)</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>DRAM Installed</strong></td>
<td>756GB (16x16GB DDR4 3200MT/s [3200MT/s])</td>
</tr>
<tr>
<td><strong>Huge Pages Size</strong></td>
<td>2048 kB</td>
</tr>
<tr>
<td><strong>NIC Summary</strong></td>
<td>Ethernet Controller X710 for 10GBASE-T, Ethernet Controller X710 for 10GBASE-T</td>
</tr>
<tr>
<td><strong>Drive Summary</strong></td>
<td>3x SLC+ 7x QLC: SLC is Solidigm's first generation SLC for cache device; QLC is a P5336 D5-P5336 61TB</td>
</tr>
<tr>
<td><strong>SPDK</strong></td>
<td>22.09</td>
</tr>
<tr>
<td><strong>CSAL</strong></td>
<td>1.0</td>
</tr>
<tr>
<td><strong>FIO</strong></td>
<td>3.29</td>
</tr>
</tbody>
</table>
Test Configuration #1: Example FIO job file

```yaml
[global]
ioengine=spdk_bdev
spdk_json_conf=${FTL_JSON_CONF}
filename=${FTL_BDEV_NAME}

# SPDK cores, FTL core mask should avoid core 0
spdk_core_mask=${SPDK_CORE_MASK}

# CPUS allowed fio threads cannot interleave with SPDK cores
cpus_allowed=12

direct=1
thread=1
buffered=0
time_based
norandommap=1
randrepeat=0
scramble_buffers=1
rw=randrw

[POR]
bs=4k
rwmixread=70
numjobs=1
iodepth=128
runtime=3600s
time_based=1
```
## Test Configuration #2

<table>
<thead>
<tr>
<th><strong>Storage Server - SuperMicro SYS-220U-TNR System Configuration</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOS Version</td>
</tr>
<tr>
<td>OS</td>
</tr>
<tr>
<td>Kernel</td>
</tr>
<tr>
<td>CPU Model</td>
</tr>
<tr>
<td>NUMA Node(s)</td>
</tr>
<tr>
<td>DRAM Installed</td>
</tr>
<tr>
<td>Huge Pages Size</td>
</tr>
<tr>
<td>NIC Summary</td>
</tr>
</tbody>
</table>
| Drive Summary       | **1. TLC** is a Solidigm TLC SSD D7-P5520 15.36 TB  
|                     | **2. 1x SLC+ 1x QLC:** SLC is Solidigm's first generation SLC for cache device; QLC is a P5336 D5-P5336 61TB |
| SPDK                | 22.09                 |
| CSAL                | 1.0                   |
| FIO                 | 3.29                  |
Test Configuration #2: Example FIO job file

```plaintext
[globals]
ioengine=spdk_bdev
spdk_json_conf=${FTL_JSON_CONF}
filename=${FTL_BDEV_NAME}
# SPDK cores, FTL core mask should avoid core 0
spdk_core_mask=${SPDK_CORE_MASK}
# CPUs allowed fio threads cannot interleave with SPDK cores
cpus_allowed=12
cpus_allowed_policy=split
direct=1
thread=1
buffered=0
time_based
norandommap=1
randrepeat=0
scramble_buffers=1
rw=randrw

[POR]
bs=64k
numjobs=1
rw=randwrite
random_distribution=zipf:1.2
iodepth=128
runtime=3600s
time_based=1
```
# Test Configuration #3

<table>
<thead>
<tr>
<th>Storage Server - SuperMicro SYS-220U-TNR System Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BIOS Version</strong></td>
</tr>
<tr>
<td><strong>OS</strong></td>
</tr>
<tr>
<td><strong>Kernel</strong></td>
</tr>
<tr>
<td><strong>CPU Model</strong></td>
</tr>
<tr>
<td><strong>NUMA Node(s)</strong></td>
</tr>
<tr>
<td><strong>DRAM Installed</strong></td>
</tr>
<tr>
<td><strong>Huge Pages Size</strong></td>
</tr>
<tr>
<td><strong>NIC Summary</strong></td>
</tr>
<tr>
<td><strong>Drive Summary</strong></td>
</tr>
<tr>
<td><strong>SPDK</strong></td>
</tr>
<tr>
<td><strong>CSAL</strong></td>
</tr>
<tr>
<td><strong>FIO</strong></td>
</tr>
</tbody>
</table>
Test Configuration #3: Example FIO job file

```plaintext
[global]
ioengine=spdk_bdev
spdk_json_conf=${FTL_JSON_CONF}
filename=${FTL_BDEV_NAME}
# SPDK cores, FTL core mask should avoid core 0
spdk_core_mask=${SPDK_CORE_MASK}
# CPUS allowed fio threads cannot interleave with SPDK cores
cpus_allowed=12
cpus_allowed_policy=split
direct=1
thread=1
buffered=0
norandommap=1
randrepeat=0
scramble_buffers=1

[WRITE_SEQ]
bs=4k
numjobs=1
rw=write
iodepth=128
size=100%
exitall

[WRITE_RAND]
bs=4k
numjobs=1
rw=randwrite
iodepth=128
runtime=1000d

[WRITE_ZIPF_0_8]
bs=4k
numjobs=1
rw=randwrite
random_distribution=zipf:0.8
iodepth=128
runtime=1000d
time_based=1

[WRITE_ZIPF_1_2]
bs=4k
numjobs=1
rw=randwrite
random_distribution=zipf:1.2
iodepth=128
runtime=1000d
time_based=1
```
## Storage Server - SuperMicro SYS-220U-TNR System Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOS Version</td>
<td>1.4b</td>
</tr>
<tr>
<td>OS</td>
<td>Fedora 37 (Server Edition)</td>
</tr>
<tr>
<td>Kernel</td>
<td>6.3.8-100.fc37.x86_64</td>
</tr>
<tr>
<td>CPU Model</td>
<td>Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz</td>
</tr>
<tr>
<td>NUMA Node(s)</td>
<td>2</td>
</tr>
<tr>
<td>DRAM Installed</td>
<td>756GB (16x16GB DDR4 3200MT/s [3200MT/s])</td>
</tr>
<tr>
<td>Huge Pages Size</td>
<td>2048 kB</td>
</tr>
<tr>
<td>NIC Summary</td>
<td>Ethernet Controller X710 for 10GBASE-T, Ethernet Controller X710 for 10GBASE-T</td>
</tr>
<tr>
<td>Drive Summary</td>
<td><strong>1x SLC+ 1x QLC:</strong> SLC P5810 800GB SLC for cache device; QLC is a P5316 D5-P5316</td>
</tr>
<tr>
<td>SPDK</td>
<td>22.09</td>
</tr>
<tr>
<td>CSAL</td>
<td>1.0</td>
</tr>
<tr>
<td>FIO</td>
<td>3.29</td>
</tr>
</tbody>
</table>
Test Configuration #2: Example FIO job file

```
global
ioengine=spdk_bdev
spdk_json_conf=${FTL_JSON_CONF}
filename=${FTL_BDEV_NAME}
# SPDK cores, FTL core mask should avoid core 0
spdk_core_mask=${SPDK_CORE_MASK}
# CPUS allowed fio threads cannot interleave with SPDK cores
cpus_allowed=12
cpus_allowed_policy=split
direct=1
thread=1
buffered=0
time_based
norandommap=1
randrepeat=0
scramble_buffers=1

[POR]
bs=4k
numjobs=1
rw=randwrite
iodepth=128
runtime=3600s
time_based=1
```