MASTERING QLC FLASH

Making flash affordable for capacity-oriented applications

PURE MADE FLASH STORAGE GREAT

The success of the Pure Storage® FlashArray™ and FlashBlade™ fundamentally derives from exploiting flash memory's advantages relative to disk—fast read response, high reliability, space efficiency, and low power consumption—while mitigating the technology's inherent shortcomings—unpredictable write performance, and most important, limited *endurance*.[1]

While managing flash *endurance* effectively,[2] Pure's arrays utilize a range of hardware and software techniques to deliver consistent I/O performance an order of magnitude greater than that of comparable disk-based ones, at equivalent cost, even as flash memory technology has evolved, becoming more challenging for array developers with each generation. Today, flash has effectively supplanted high-performance magnetic disk in high-end arrays.

But There Is Still a Place for Disk

Not all online data needs flash performance. The enormous data sets used by applications like analytics, artificial intelligence, online backup/restore, and so forth makes storage cost the driving factor in configuring systems for them. And as Figure 1 suggests, while flash is now less expensive than high-performance disk, high-capacity disks of middling performance are still significantly more affordable for very large data sets.

But that is not to say that these *capacity-oriented* applications are completely insensitive to I/O performance. Most require high throughput rather than low latency for individual I/Os, but they still tax the capabilities of arrays based on large-capacity disks.

As the I/O performance needs of capacity-oriented applications have increased, array developers have looked for ways to make flash affordable for the substantial market they represent.

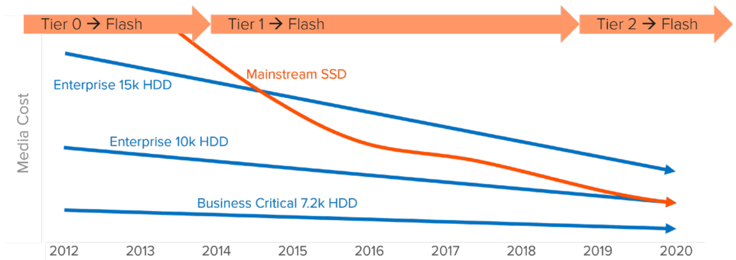


Figure 1: Storage Media Cost Projections (ca. 2019)

Some vendors have attempted to serve the capacity-optimized market with *hybrid* arrays

[1] The number of times flash cells can be overwritten before they no longer retain data reliably.

[2] So much so that the company offers lifetime guarantees on the flash devices in its arrays.

populated with large-capacity disks and smaller amounts of solid-state storage that they effectively use as cache. While these can improve the performance of capacity-oriented applications, they have disk-driven footprint and reliability, and moreover, must be integrated with data center workflows. Their most fundamental shortcoming, however, is that paradoxically, they attempt to improve throughput for low-cost arrays by adding costly components to them.

Today, the dominant factor that inhibits the use of flash for very large capacity storage is cost. While the flash cost curve has intercepted the high-performance disk one, large-capacity disks have remained a less expensive bulk storage medium…*until the introduction of QLC*.

The March of Flash

Flash memory chips store data in hierarchically organized *cells* (Figure 2), each of which is effectively a transistor that is programmed with an electric charge which it returns it in the form of current flow when a voltage is applied to its control gate.

The earliest flash devices stored a bit in each cell. Data was detected by current flow (stored 1) or its absence (stored 0) when control voltage was applied. The single level output led to the name *single-level cell* (SLC) for the technology.

The next generation of flash, *Multi-Level Cell* (MLC), stored two data bits in each cell, doubling *density* (the amount of data that a flash die could hold). The earliest FlashArrays were equipped with solid state devices (SSDs) that used MLC technology internally.

MLC flash was followed by a *Triple-Level Cell* (TLC) flash generation that increased density by a further 50%. Today, TLC is the technology of choice for high-performance arrays—the *Direct Flash Modules* (DFMs) in FlashArray//X and FlashArray//XL are based on TLC technology.

Around 2017, chip vendors introduced *Quad-Level Cell* (QLC) flash.[3] Four bits per cell increases density by 33% over TLC, potentially bringing the cost of arrays based on it within the realm of capacity-oriented applications.

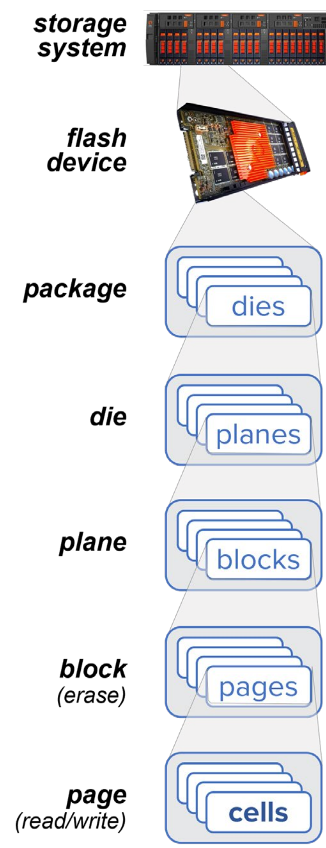


Figure 2: Flash Organization

Higher Density Isn't Free

Each step on the flash evolutionary path has delivered higher density (and therefore lower cost per-bit) storage, but each

[3] For example, https://www.tweaktown.com/news/58223/toshiba-announces-worlds-first-qlc-flash-memory/index.html

has become more challenging for both device and array developers in four important ways:

DATA INTEGRITY

Each increase in the number of bits per flash cell requires more sensitive output detection when data is retrieved. With SLC, on-off detection is sufficient; MLC devices must detect four output levels; TLC, eight levels, and QLC, 16. Moreover, reading data disturbs cell contents causing output to vary slightly from read to read. With earlier generations, the effect was ignorable, but with QLC, this *read disturb* effect is significant. Together, these factors mean that more bits per cell inevitably results in increased *raw bit error rates* (RBERs) on output

DATA RETENTION

Flash does not retain dormant data forever. Stored data must be *refreshed* periodically, especially if it is *dormant* (not retrieved). With each generation, device retention times have decreased, so refresh frequency must be managed so as to preserve stored data

MEDIA ENDURANCE

Writing to flash is inherently destructive. Each erase and overwrite slightly degrades the media's *endurance* (ability to retain data reliably). SLC degradation is negligible; chip manufacturers typically specify device endurance of around 100,000 *program/erase* (P/E) cycles. Typical MLC specifications are much lower (6,000-10,000 cycles)—manageable in enterprise arrays, as early FlashArray models demonstrated. TLC endurance is lower yet (typical specification: 3,000 cycles)—but still manageable at the array level. With QLC however, endurance is less than 1,000 cycles—very difficult to support in enterprise arrays

WRITE PERFORMANCE

Writing to flash take longer than reading, partly due to the nature of the media, but also because erasure of entire blocks is sometimes necessary before pages can be written. Typical write times for flash pages have increased with each generation—from 0.5ms with SLC to 2.5ms with TLC to ***10-20ms with QLC***. Array designers must overcome this increasing latency to provide acceptable write performance to hosts.

To deal with these challenges, flash-based SSDs contain elaborate *flash translation layer* (FTL) firmware that dynamically relocates stored data as needed and maps sector numbers used by hosts to identify data to the data's current locations on flash. FTLs add complexity and processing time, and cause *write amplification*,[4] which detracts from SSD performance and endurance.

Because flash is typically the largest cost component of an all-flash array (AFA), higher density flash can reduce array cost, but each increase in density requires more sophisticated array engineering to deliver the performance and reliability that enterprise storage needs. This is especially true with QLC, for which RBER, data retention, device endurance, and write performance all fall well below the requirements of enterprise arrays.

[4] Internal writes to flash media that SSDs use to relocate previously stored data as they persist newly arriving data from hosts, and for other "housekeeping" operations.

PURE STORAGE AND FLASH

From the outset, Pure Storage engineers recognized that while flash SSDs are superficially similar to disks, flash is a fundamentally different medium and requires different treatment at the array level. This recognition led to the development of custom *storage units* for FlashBlade and later, *Direct Flash Modules* (DFMs) for FlashArray, each with accompanying controller software that fills the role of device-level FTL. Coupled with other innovations described below, these designs have positioned the company for dealing with the unique challenges posed by QLC flash.

FLASH FOR CAPACITY-ORIENTED APPLICATIONS

Today, Pure's FlashArray//C arrays contain DFMs populated with QLC media.[5] The arrays offer somewhat lower performance than FlashArray//X and FlashArray//XL, but their capacity is greater, and due to their use of off-the-shelf chips, their cost/usable GB is significantly lower.

Integrating Purity//FA's core software architecture with DFMs and the companion *Direct Flash Software* (DFS) surmounts the limitations of off-the-shelf QLC flash to produce arrays whose price points make all-flash storage affordable for capacity-oriented applications. DFS elevates most FTL functions (including read disturb compensation) to the array controller level, allowing arrays to manage their flash *globally* (across all installed DFMs) and precisely (down to individual page level). Their flash management capabilities aren't possible with arrays that use general-purpose SSDs containing QLC flash for persistent storage:

DATA INTEGRITY

DFMs use *low-density parity check* (LDPC) forward error correction adapted from the telemetry sector and *soft decoding*.[6] They track the state of individual flash blocks and adjust programming and overwrite modes as needed to minimize RBERs of off-the-shelf flash chips

DATA RETENTION

With array-wide visibility to flash, DFS can direct DFMs to refresh dormant blocks exactly as needed while minimizing interference with other DFM operations

MEDIA ENDURANCE

DFS elevates write scheduling to the controller level, which minimizes array-wide write amplification compared to what can be achieved by device-level flash management. Both DFMs and array controllers cache data extensively to minimize flash access

WRITE PERFORMANCE

DFMs perform intra-device data movement internally as directed by DFS, but the primary mechanism for delivering consistent high write performance to hosts is the basic Purity//FA technique of staging data redundantly in high-performance NVRAMs. For FlashArray//C with QLC media, average write latency is in the 2-4 millisecond range.

---

[5] The company anticipates delivering FlashBlade models equipped with QLC flash at a future time.

[6] https://www.tutorialspoint.com/hard-and-soft-decision-decoding Other vendors also use these technologies in AFAs.

LARGE-CAPACITY FLASH STORAGE: A HOLISTIC APPROACH

While the abovementioned features make QLC flash feasible as an enterprise storage medium, the cost, performance, and data reliability of FlashArray//C arrays are also due in part to dozens of techniques that have long been part of Purity//FA (and FlashBlade's Purity//FB) software. Two of the most visible ones, originally developed to enhance flash endurance in earlier generations, are *data reduction* and *log structuring*. These have taken on heightened importance with QLC media, where minimizing the lifetime number of flash cell overwrites is crucial.

DATA REDUCTION

Both FlashArray and FlashBlade compress unique content before writing to minimize the amount of data they write to flash. In addition, FlashArrays replace duplicated content with pointers to a single stored copy of the content. While the amount by which these techniques reduce data depends on the nature of the data, the net result that less data is written to flash over an array's lifetime. With FlashArray, reductions of 80% of data written by hosts are not uncommon.

LOG STRUCTURING

When hosts overwrite disk block or file contents, FlashArray and FlashBlade do not overwrite flash directly. Instead, they organize incoming data into densely populated *logs* buffered in DRAM (protected against power loss by distributed NVRAM) and *flushed* (written) to flash in large blocks aligned with flash devices' internal organizations (Figure 2). Periodically, they *garbage collect* to consolidate live data and reclaim space occupied by overwritten data for reuse.

Based on data from nearly 20,000 deployed FlashArrays, these two techniques result in long-term reductions of ⅔-¾ in the number of writes to flash, placing both TLC and QLC devices well within chip vendor specifications for lifetime overwrites, as Figure 3 suggests (MLC is shown for historical reasons). The success of these techniques has been borne out by Pure's experience. Deployed arrays continue to exhibit extremely low incidence of media failure over flash generations, even as nominal device endurance has decreased with each. Pure's confidence in its software's ability to manage flash endurance is such that the company offers lifetime replacement guarantees on the flash devices in its arrays.

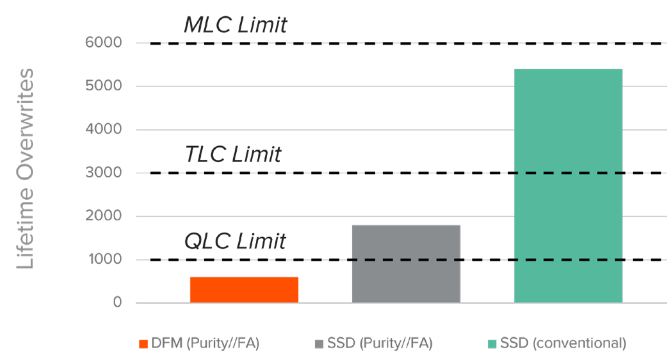


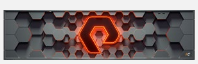
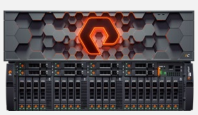
Figure 3: Flash Endurance (typical)
(combined effect of Purity//FA, DFM, & DFS)

FLASHARRAY//C: FLASH FOR CAPACITY-ORIENTED APPLICATIONS

In late 2019, Pure Storage introduced FlashArray//C, a line of full-function all-NVMe FlashArrays optimized for low-cost and high capacity rather than performance. The arrays initially used the company's custom designed TLC-based DFMs for persistent storage.

When QLC flash became available a year later, the TLC-based DFMs were replaced by QLC-based ones in new versions of the products. Today, FlashArray//C arrays can mount up to 494TB of physical QLC flash in a 3U high chassis. The more powerful FlashArray//C60 model can accommodate 49TB DFMs for an in-chassis capacity of 980TB, and can be expanded with a DFM-only chassis containing 28 DFMs, to as much as 1.37PB of physical flash.[7] Table **1** summarizes FlashArray//C physical capacities along with user data capacities that can be expected at an average 5:1 data reduction (based on automated reports from installed arrays.[8]

Table 1: FlashArray//C Maximum Capacities

Model	Rack Space	Physical Flash	Effective Capacity (5:1 reduction)
 FlashArray//C40	3U	494TB (20 x 24TB DFMs)	2.47PB
 FlashArray//C60	3U	980TB (20 x 49TB DFMs)	4.9PB
 FlashArray//C60	6U	1.37PB (20 x 24TB DFMs + 28 x 49TB DFMs)	6.8PB

[7] Assumes 20 x 24TB DFMs in the array chassis and 28 x 49TB DFMs in the expansion chassis.

[8] Arrays installed at sites that do not permit external network connections do not report data to Pure1, so their usage is not included in this calculation.

THE MODERN DATA EXPERIENCE FOR VERY LARGE DATA SETS

While FlashArray//C's consistent 2-4 millisecond latency at petabyte scale made flash storage affordable for capacity-oriented applications, it's also important to remember that they are full-function FlashArrays, offering the same:

- data services, including snapshots and replication
- field-proven 99.9999% availability
- cloud-based management via Pure1™
- Evergreen™ investment protection

that are available with other members of the FlashArray family.

With the full complement of features, including policy-driven snapshots and replication, FlashArray//C can integrate with other FlashArray family members and features to enable a variety of novel workflows that include capacity-oriented applications based entirely on flash storage. For example:

VIRTUAL MACHINE TIERING

Users can allocate virtual machines requiring high performance on FlashArray//X models, and less-demanding ones on FlashArray//C

LOW-COST DATA PROTECTION

Users can implement replication policies that back up production data from FlashArray//X to lower-cost FlashArray//C storage, and from there to public cloud storage using CloudSnap

DEVELOPMENT & TEST

Multiple production data sets from different arrays can be replicated to a single FlashArray//C for low-cost development and testing against actual production data.

Thus, in addition to making robust flash storage affordable for capacity-oriented applications, QLC-based FlashArray//C models can increase data center flexibility with low-cost flash-based storage that improves existing workflows and enables new ones.

QLC Flash and FlashArray//C:
Flash performance with FlashArray features
for capacity-oriented applications