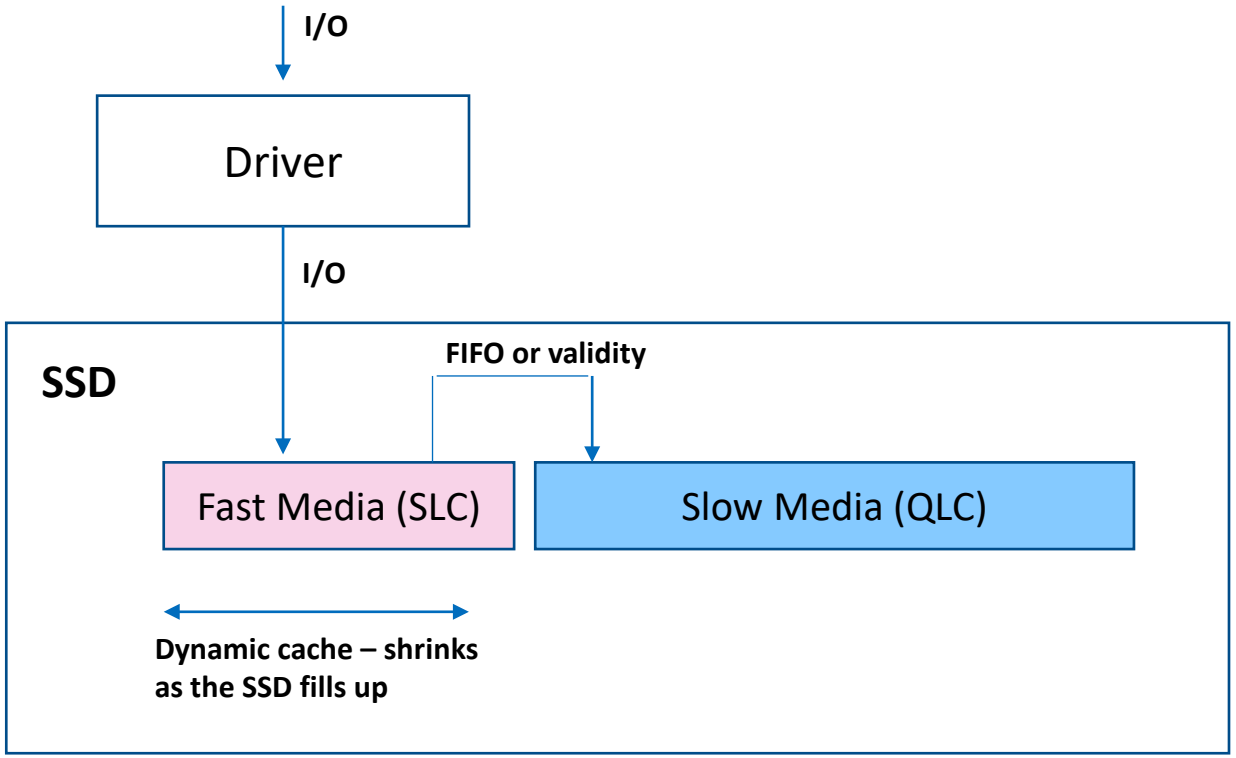
Flash Memory Summit

SOLIDIGM

# Evolution of Client SSD Architecture

Sanjay Subbarao

# Traditional Client SSD caching architectures




- First generation of SSDs capable of storing multiple bits per cell. With those drives came the advent of the SLC (single-level cell) cache. Virtually every SSD sold today uses a cache of this kind as a sort of "fast access" area on the drive
- Things in this relatively small cache can be loaded quickest, while others stored in the larger area outside of the cache tend to load more slowly.

- As your SSD fills up, the cache space shrinks to make room for additional files. This is an industry-standard behavior called "dynamic" caching.

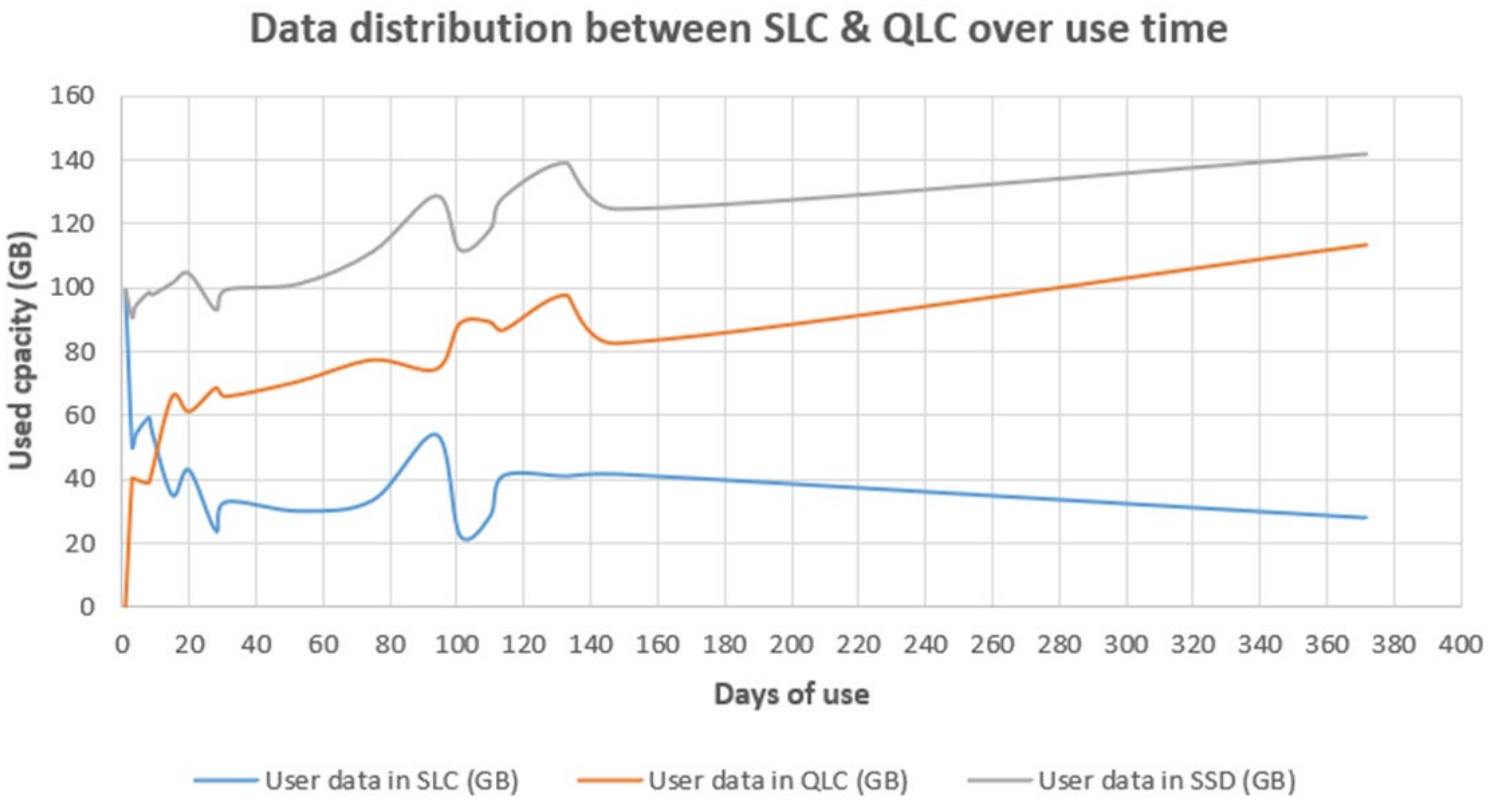
# Traditional Client SSD issues



- SSDs are simplistic in their use of cache space: every new thing written to the drive (i.e., every file saved) goes straight into the cache, where it stays until newer things eventually kick it out. We call this the "data treadmill" effect.

- Consequently, when your drive is 50% or more full, two things are compounding.
  - There's more data on the drive that you (presumably) will want to access at some point.
  - There's less cache space available to serve it up quickly. This leads to a general storage slowdown over the life of a typical SSD.

# Traditional Client SSD issues - Continued



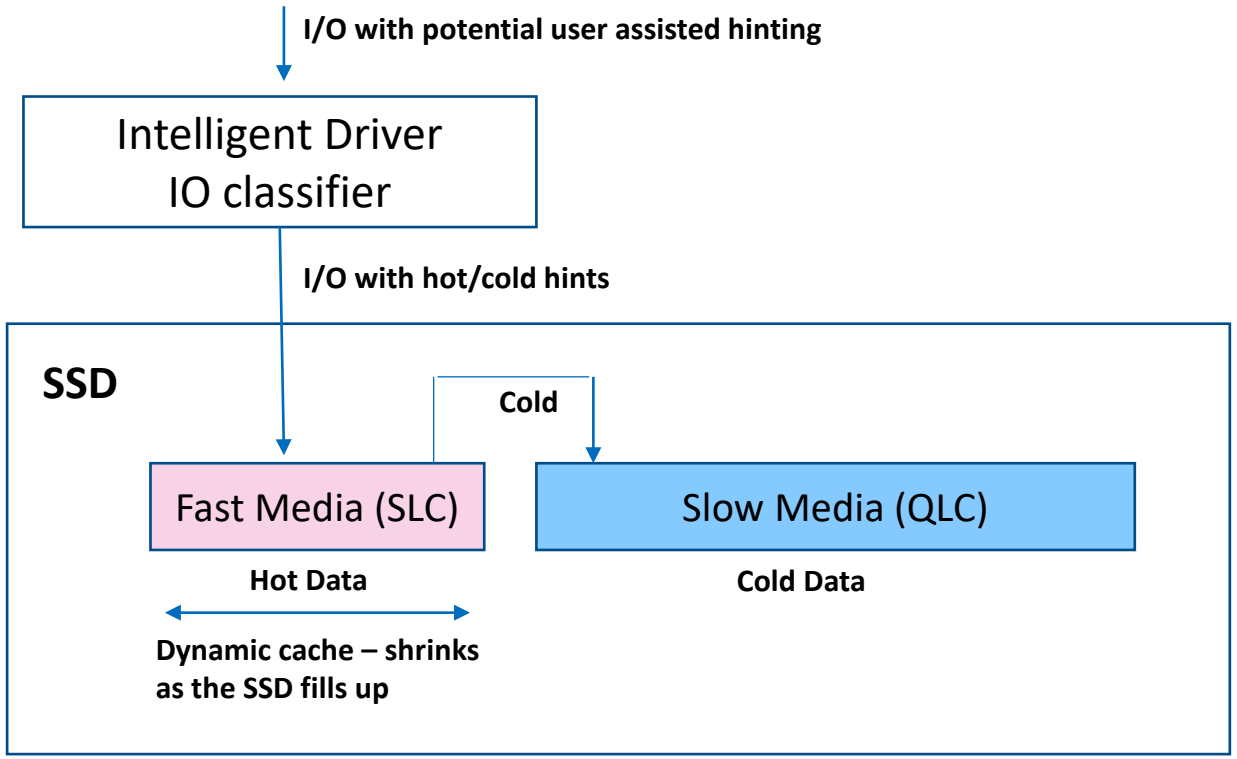

- **Traditional SSDs**: SLC is used as write buffer with opportunistic read cache. Majority of the data gets moved to QLC in short time (after few weeks) and once moved, there is no intelligence to bring it back to SLC.

- **Fast Lane - Proposed caching solution:** SLC is used as write buffer and Intelligent caching to keep “hot” data in SLC as much and as long as possible

# Fast Lane SSD caching



Storage Driver identifies, which data is hot or cold and passes these as hints to SSD Firmware. SSD uses the Hints to keep the Hot data in Fast media and Cold data in Slow media



# Fast lane SSD caching



- Is increasingly important to make better use of your limited cache space. Fast Lane observes your actual usage to determine which files and applications are most important to you, and then prioritizes keeping those items in the cache.

- The main benefit to in this approach is performance consistency. By optimizing what's in your cache, you increase the likelihood that your most important data will be available quickly. That likelihood goes up relative to conventional caching as the drive fills—and your cache size consequently decreases—because the conventional approach was already making sub-optimal decisions about what to cache (based solely on write recency) and is now doing so in an increasingly smaller cache space.

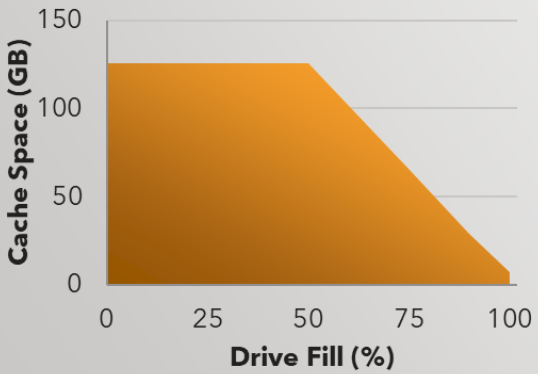
- With Fast Lane, we see improvements of up to 120% in QD1 random read speed (a key driver of responsiveness) on drives that are 50% full compared to the same systems without the feature.

# Fast Lane benfits

Flash Memory Summit

## The Problem

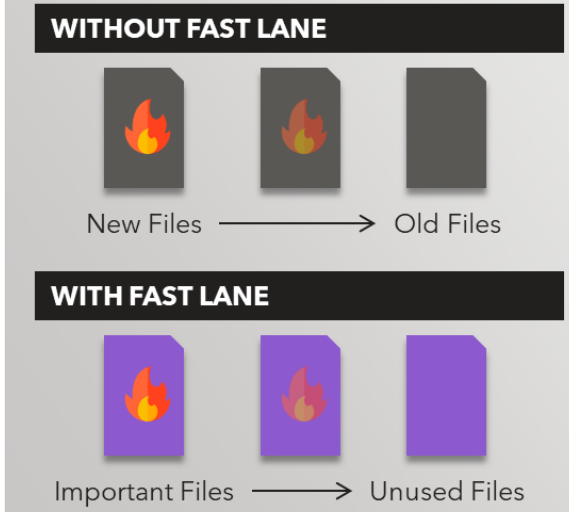
As your SSD fills up, cache space (the "fast access" area) is reduced to make room for new data, leading to general storage slowdown.




Example: Dynamic cache sizing on a 1TB SSD

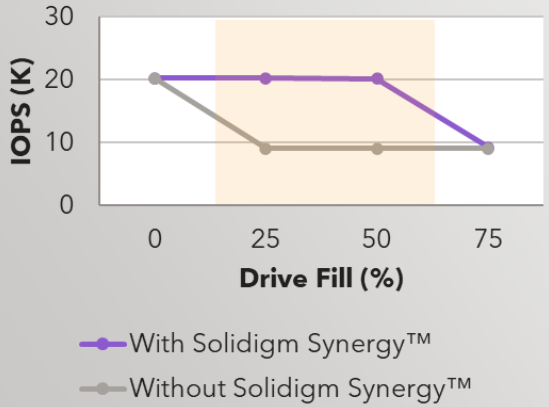
## The Solution

Fast Lane tracks which files and applications you use most often – and prioritizes those within the cache.




## The Result

Access to important files and applications remains fast, even as the drive fills up and the cache shrinks.




Up to **120%** **Faster Reads** on a 50% Full SSD[1]

35 YEARS OF NAND FLASH MEMORY INNOVATION

# Future Possibilities



- Machine learning (ML) for IO prediction
- User assisted hinting – File pinning

# 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.

Contact your Solidigm representative or your distributor to obtain the latest specifications before placing your product order.

For copies of this document, documents that are referenced within, or other Solidigm literature, please contact your Solidigm representative.

All products, computer systems, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

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.

Solidigm or Intel optimizations, for Solidigm or Intel compilers or other products, may not provide optimized performance to the same degree for non-Solidigm or Intel products. Solidigm or Intel technologies may require enabled hardware, software, or service activation.

Your costs and results may vary.

Solidigm does not control or audit third-party data. You should consult other sources to evaluate accuracy.

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.