Enabling 3D QLC NAND flash

Radu Stoica, Roman Pletka, Nikolaos Papandreou, Nikolas Ioannou, Sasa Tomic, Haris Pozidis

IBM Research – Zurich

Agenda

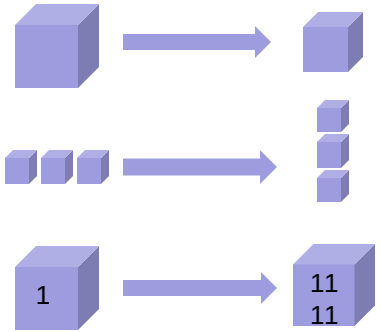
- Part I: Implications of 3D QLC NAND flash
  - Understand 3D QLC issues
  - Controller design challenges
  - Techniques for enabling 3D QLC

- Part II: Novel ideas to enable 3D QLC NAND
  - Dynamic block mode switching

Note: This talk assumes familiarity with NAND flash

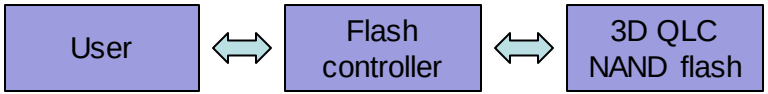
NAND flash trends

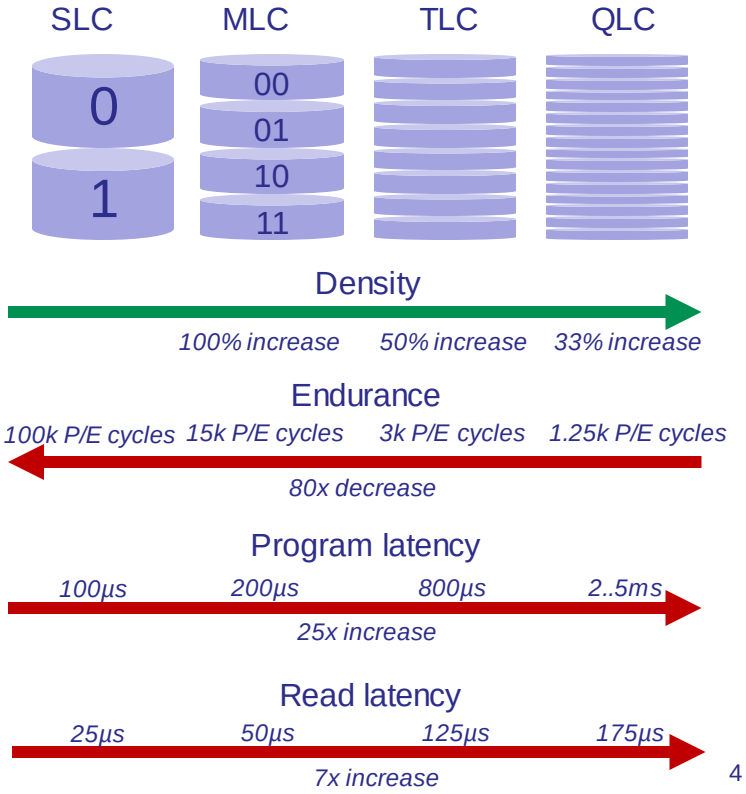
- Storage technology with highest density & fastest growth in density (compared to DRAM & HDD)
- Growth in density enabled by three approaches:
  - Feature size shrink
  - Vertical stacking (3D NAND flash)
  - Increase in bit density (QLC flash)
- Current 3D QLC technologies:
  - Feature size: ~10 nm
  - Vertical stacking: 64-128 layers
  - 4 bits per cell



3D QLC NAND specs

- All major quality metrics degrade when increasing bit density.
- The metrics, on their own, are misleading
  - SSDs != 3D QLC NAND

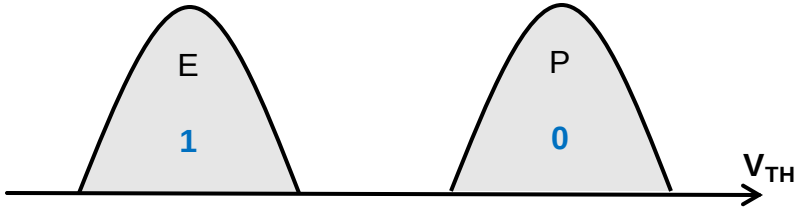




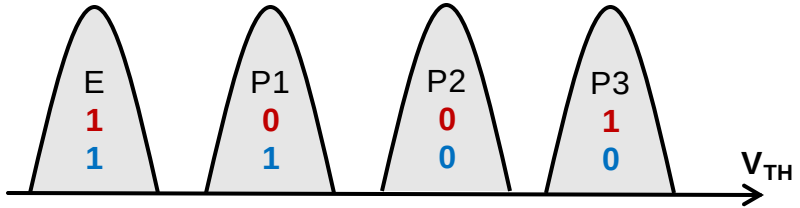
In the first part, we will briefly cover:

- **Why** does higher bit density affect quality metrics?
- **What** are the implications on controller design?
- **How** to overcome controller design challenges?

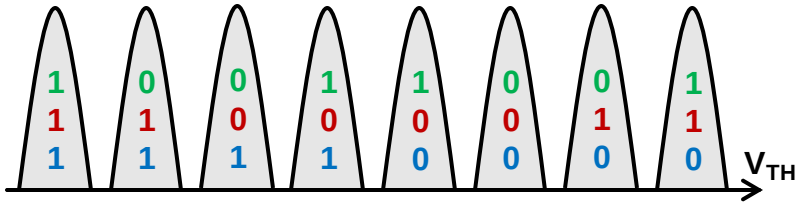
Data representation in NAND flash



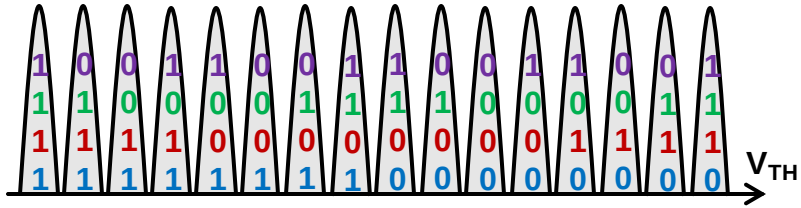
Single **L**evel **C**ell (**SLC**): 2 States (1 Erase + 1 Pgm)
= 1 bit of information per cell



Multi **L**evel **C**ell (**MLC**): 4 States (1 Erase + 3 Pgm)
= 2 bits of information per cell
= 2x capacity of SLC



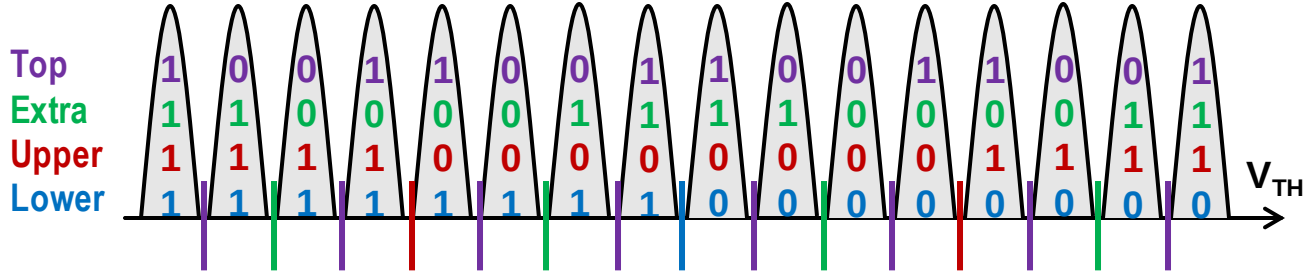
Triple **L**evel **C**ell (**TLC**): 8 States (1 Erase + 7 Pgm)
= 3 bits of information per cell
= **1.5x capacity of MLC**



Quad **L**evel **C**ell (**QLC**): 16 States (1 Erase + 15 Pgm)
= 4 bits of information per cell
= **1.3x capacity of TLC**

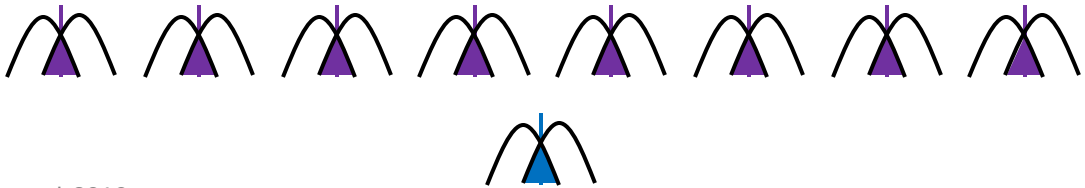
QLC read non-uniformity

- Several low-level reads required for retrieving the data stored in a page
- Successive a voltage thresholds are applied to differentiate between the voltage ranges
- The number of voltages ranges depends on the page type
- Read latency is grows proportionally with the number of voltage levels applied



Top: 8 levels
Extra: 4 levels
Upper: 2 levels
Lower: 1 level

- Read error likelihood also increases with the number of voltage levels applied



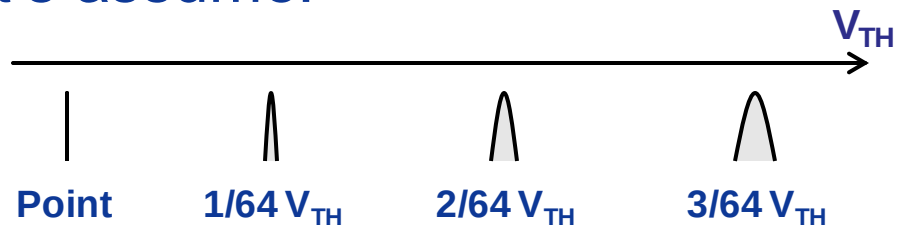
Top: 8 error regions
Lower: 1 error region

Implications of bit density growth

Simple thought experiment. Let's assume:

Voltage range:

Distribution widths:



SLC → QLC

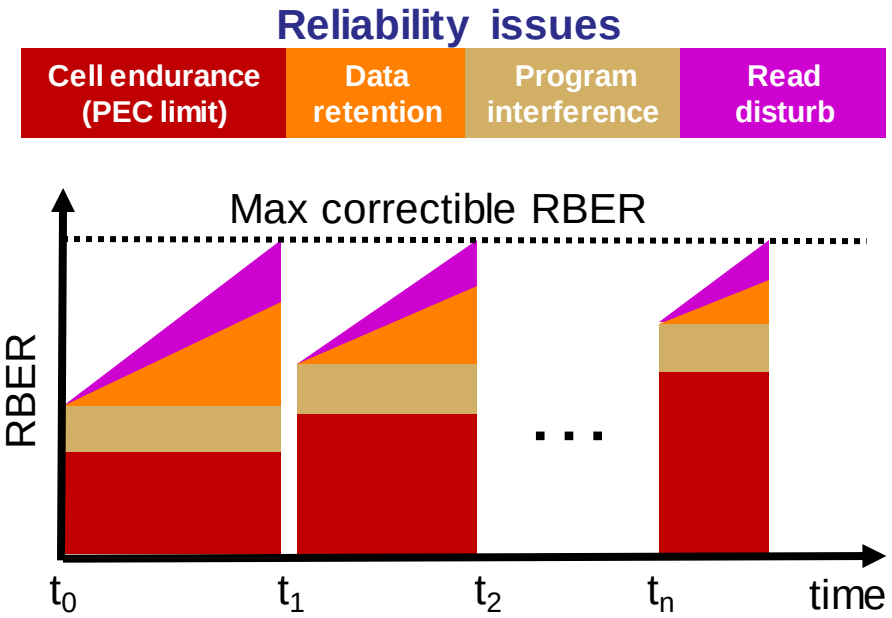
V_{TH} margins shrink by:

Intrinsic drop in V_{TH} margins due to exponential increase in voltage states

Further drop in V_{TH} margins due to voltage distribution width

NAND Flash reliability issues

- Bit increase (SLC→MLC→TLC→QLC) affects all error sources
- PEC limit not necessarily the most limiting issue
  - SSDs see much less writes than expected
- PEC-related errors do not necessarily cause data loss
- PEC limit is flexible
  - Defined by ECC, TVS, retention target, page program techniques, etc.



Page program constraints

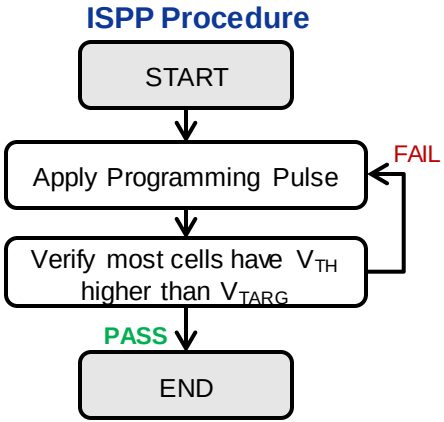
V_{TH} distributions width minimized through page program techniques:

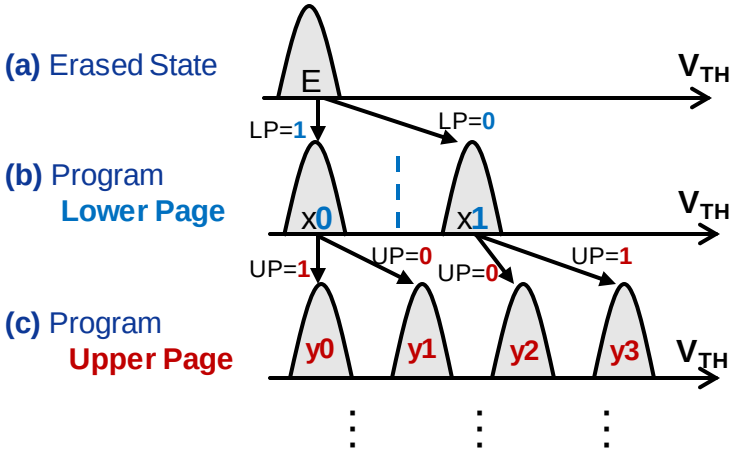
1. Incremental Step Pulse Programming (ISPP)
2. Program one page at a time
3. Program pages along dominant axis of interference
4. Timing constraints

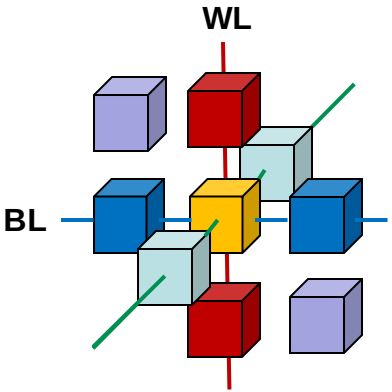
Low write bandwidth

Latency tails

Large data in-flight



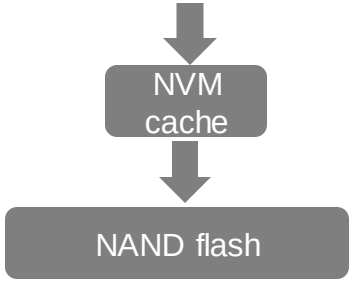




Write caching in SSDs

- Most SSD controllers cache data:
  - ***Decouples user I/O latency from page program latency***
  - Simplifies data striping over all flash I/O units
  - Guarantees full block writes & program timing
  - Simplifies recovery
  - Mitigates first read issue
- Controller implications:
  - NVM cache size is becoming an issue
  - Data persistency in case DRAM is used

  (low bandwidth to flush cache in event of power loss)



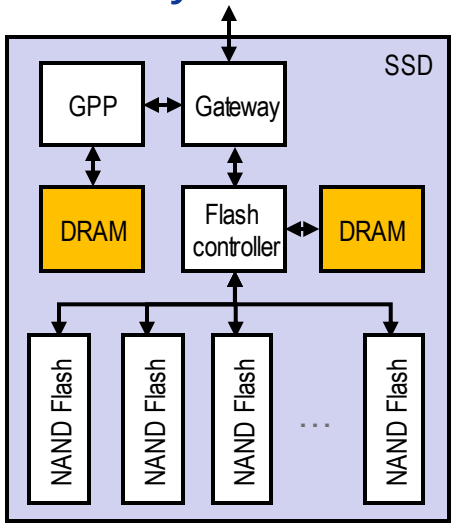
Hypothetical SSD

Avg. program time	*2.5ms*
Page size	*16kB + 2kB*
Planes/die	*4*
Pages/block	*4096*
Dies/SSD	*16*

*18kB * 4096 * 4 * 16 = 4.5GB*
Multiple write streams required!

Metadata management

- Typically metadata does not fit in controller DRAM and must be paged
  - Some metadata accesses are in the critical I/O path
- QLC puts pressure on metadata management in several ways:
  - Metadata growth due to increase in storage capacity
  - Additional types of metadata sources
    - Finer grained threshold voltage shift values
  - High & variable read latencies discourage paging
- Increase in DRAM not desirable
  - Higher $/GB, takes real estate, power concerns



Techniques for enabling 3D QLC

Reliability			Performance			HW	
Cell endurance (PEC limit)	Data retention	Read disturb	Low write throughput	Slow reads	Latency tails	NVM cache	DRAM size

Reuse:

- Write heat segregation
- Data reduction
- Wear levelling
- Health binning
- Data refresh
- Stripe-level parity
- Bad block mgmt.

Adapt:

- ***Voltage shifting****
- ECC
- I/O scheduling
- I/O prioritization
- Command suspend

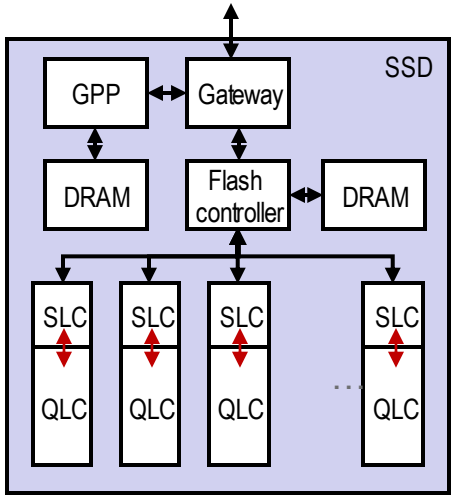
Adopt:

- Dynamic block mode
- Leverage page properties

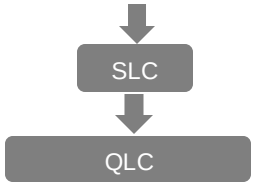
*Patrick Breen, Flash Characterization Engineer, IBM
"Component-Level Characterization of 3D TLC, QLC, and Fast SLC NAND",
Wednesday, August 7th, FTEC-201-1, 3:20-5:45 PM

Dynamic block mode

- Most flash chips support multiple bit modes:
  - High endurance, low capacity mode => SLC mode
  - Low endurance, high capacity mode => QLC mode
- A hybrid SLC/QLC controller leverages the multiple bit modes
  - Flash blocks split into SLC & QLC pools
- Venues for improving SSD behavior:
  - Longer device life: real workloads are skewed and frequently updated data can be stored in SLC
  - Higher bursty write bandwidth: larger SLC cache can absorb write spikes
  - Better tail latencies: SLC writes have >10x lower latency, SLC reads 8x
  - Customization: allows users to configure device according to their needs
  - Replace (most) NVM cache
  - Reduce DRAM by allowing more efficient paging



Architectural view

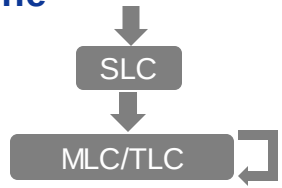


Logical view

Background on hybrid SSD controllers

1st Generation: Fixed-size SLC cache

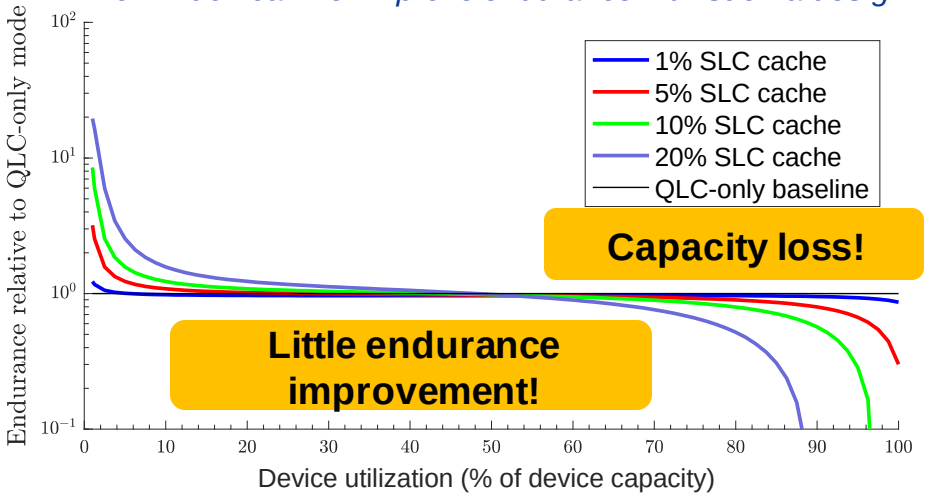
- Controller design:
  - Use a small region of the Flash as a **static SLC cache**.
  - Data is first written to SLC, then destaged to MLC/TLC when SLC cache is full



- Benefits:
  - NVM cache replacement
  - Higher bursty throughput
  - Read latency reduction for data read from SLC
- Extensions:
  - Some manufacturers provide extensions (e.g., on-chip copy from SLC to MLC/TLC)
- Challenges:
  - Write speed drops significantly when SLC cache is full
  - Static assignment of blocks leads to unequal wear
  - Capacity reduction & increased cost

- Experiment:
  - Random write workload
  - Remaining utilized space holds static data
  - Controller parameters:
    - 1, 5, 10, 20% of physical blocks set to SLC mode
    - 20% total over-provisioning irrespective of the SLC size
    - Assuming SLC endurance 40x higher than QLC endurance

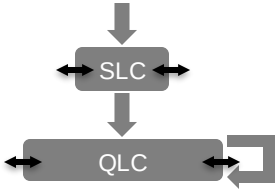
How much can we improve endurance with such a design?

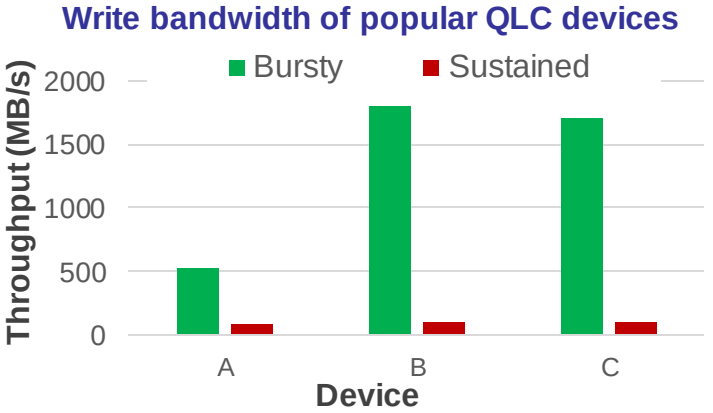


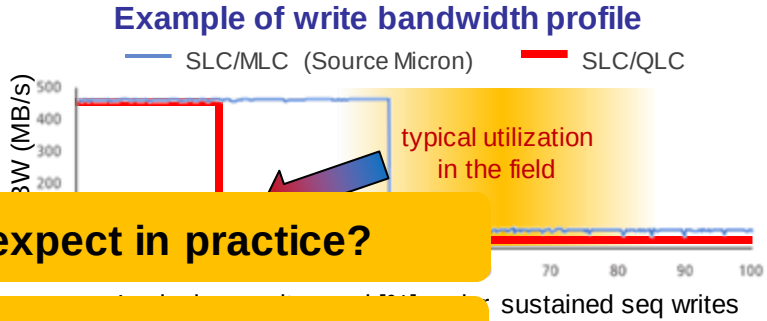
Background on hybrid SSD controllers

2nd Generation: Adaptive SLC caching
(i.e., Dynamic Write Acceleration DWA, Intelligent Dynamic SLC-Caching)

- Controller characteristics:
  - Dynamically switch block modes
  - SLC size depends on logical capacity used
  - SLC destage performed in the background
- Benefits:
  - Read latency reduction for data read from SLC
  - Higher throughput for bursty write workloads
  - No user capacity reduction
- Challenges:
  - Write speed drops when utilization reaches a certain level
  - Low endurance specifications
  - SLC cache policies can be further improved
  - Require





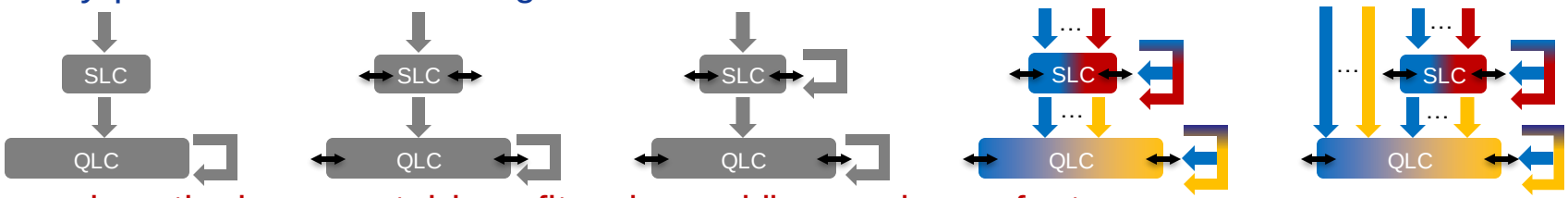


What endurance & performance to expect in practice?

How much more can we improve?

How to quantify hybrid controller benefits?

- Many possible controller designs:



  => show the incremental benefits when adding each new feature

- Optimal configuration depends on the workload, device utilization, flash parameters

  => explore all possible configurations and select the best

- Write endurance & performance depend on SSD resources

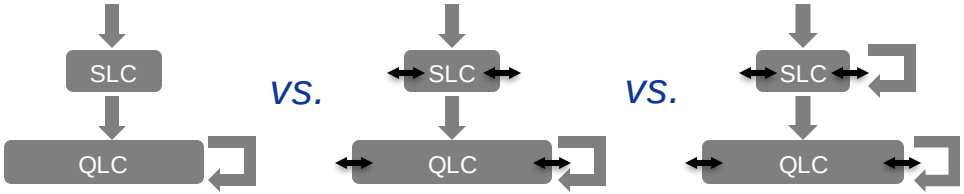
  => Endurance: show improvement over QLC-only controller. Write: show chip busy time

- Real workloads are typically skewed and device utilization varies considerably

  => Vary device utilization & use skewed workloads

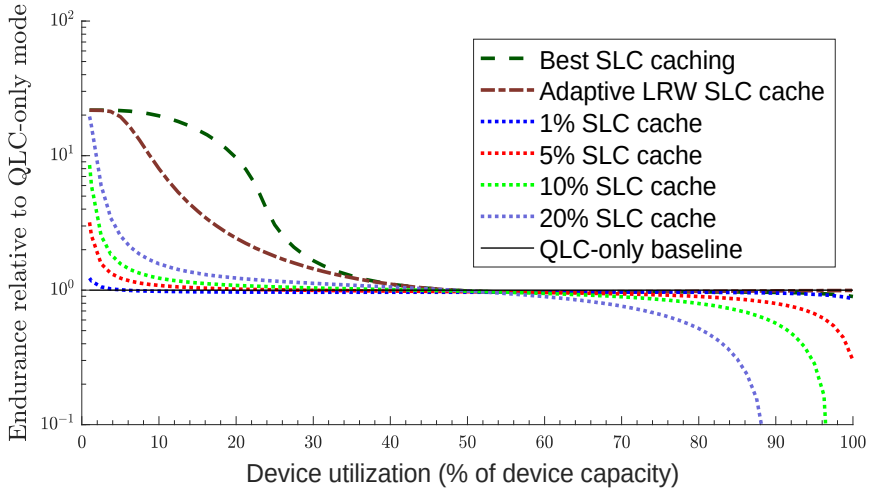
Hybrid controllers with SLC cache - endurance

Fixed vs. optimally sized SLC destage buffer:



Experiment:

- Random write workload to all occupied address space
- Controller with fixed SLC cache:
  - 1, 5, 10, 20% of physical blocks set to SLC mode
- Controller with adaptive SLC cache:
  - Uses optimal SLC/QLC ratio for given utilization
- Controller with SLC cache & SLC-to-SLC relocation:
  - Uses optimal SLC/QLC ratio & SLC occupancy



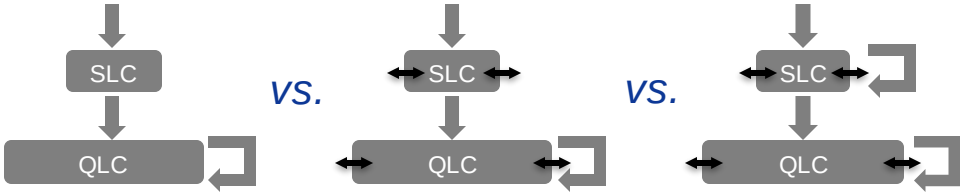
Significant gains for dynamic SLC caches

No improvement

Fixed-size caches are counterproductive

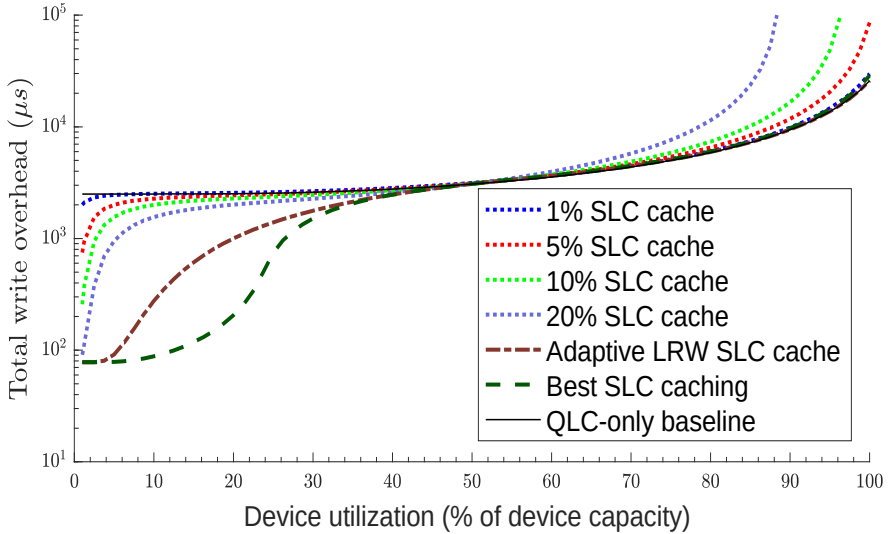
Hybrid controllers with SLC cache – performance

Fixed vs. optimally sized SLC destage buffer:



Experiment:

- Random write workload to all occupied address space
- Controller with fixed SLC cache:
  - 1, 5, 10, 20% of physical blocks set to SLC mode
- Controller with adaptive SLC cache:
  - Uses optimal SLC/QLC ratio for given utilization
- Controller with SLC cache & SLC-to-SLC relocation:
  - Uses optimal SLC/QLC ratio & SLC occupancy
- **Showing chip busy time (total write overhead)**



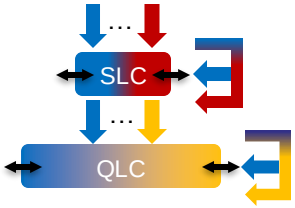
Significant gains for dynamic SLC caches

No improvement

Fixed-size caches are counterproductive

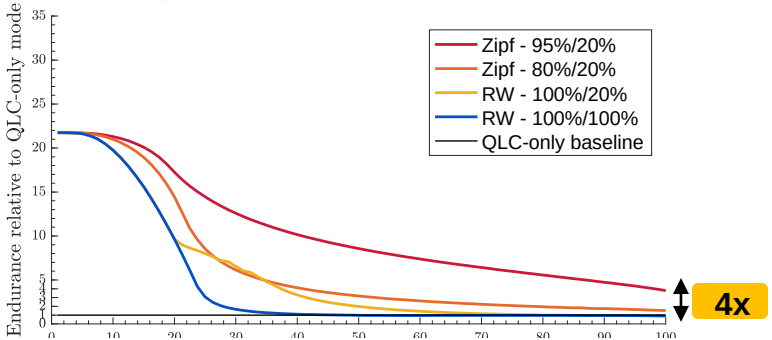
Exploiting write skew – Controller designs with heat segregation

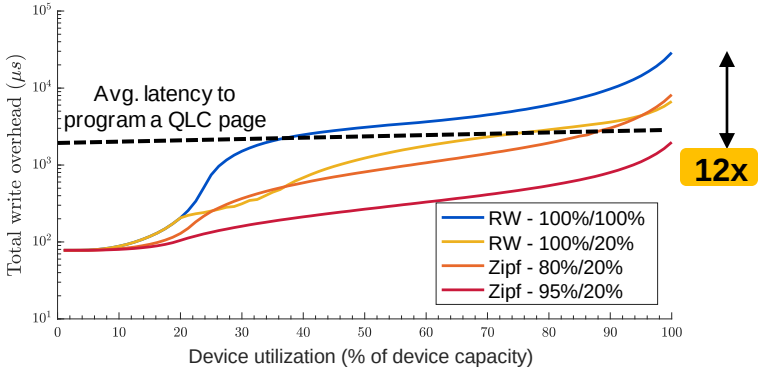
Quantify the benefits of using write heat



Experiment:

- Workloads with varying amount of skew
  - No skew - random to all utilized capacity
  - Some skew – random up to 20% of device capacity
  - Skewed – Zipfian 80%/20%
  - Highly skewed – Zipfian 95%/20%
- Controller with full write heat segregation:
  - Detects write heat
  - Determines optimal SLC/QLC ratio for given utilization.
  - Determines optimal size of hot dataset to keep in SLC

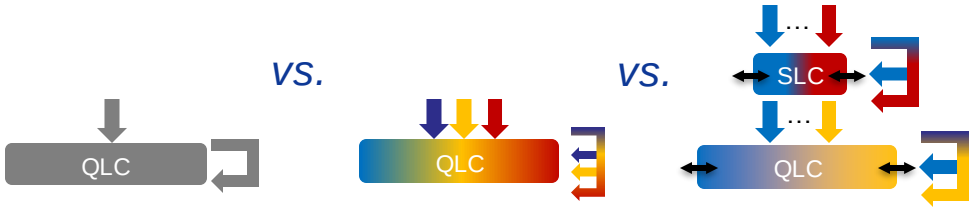




Significant improvement in both endurance and write performance for the skewed workloads

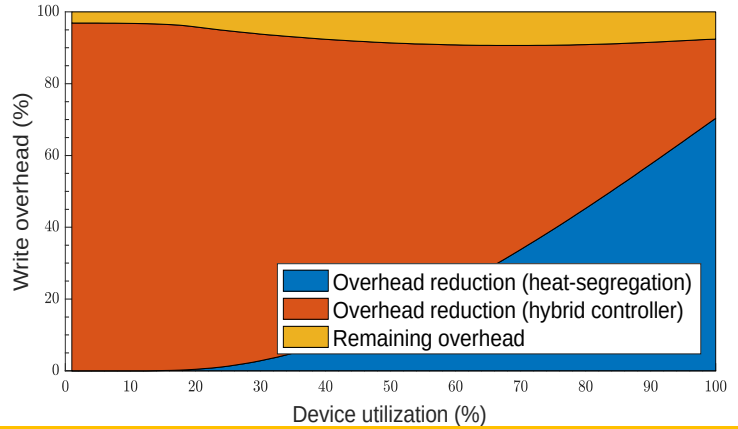
Breakdown of the benefits of using write heat

Write heat information helps reduce both cleaning overhead and data destage. Which effect dominates?



Experiment:

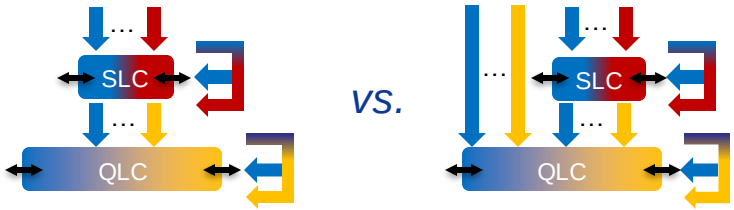
- Zipfian 95/20 workload
- Controller no heat information and no SLC tier
  - LRW cleaning policy
- QLC controller that uses heat information
  - Applies write heat segregation to reduce GC overhead
- SLC/QLC controller that uses heat information
  - Applies write heat segregation to reduce GC overhead
  - Prioritizes data destages to QLC



For skewed workloads, a hybrid controller can improve write performance even at high device utilization

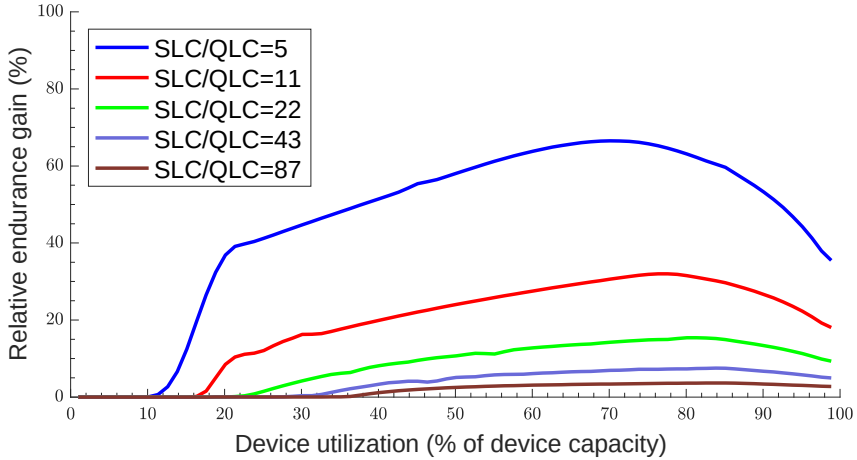
Hybrid controllers with SLC cache

Hybrid controller with SLC cache vs. SLC/QLC tiering



Experiment:

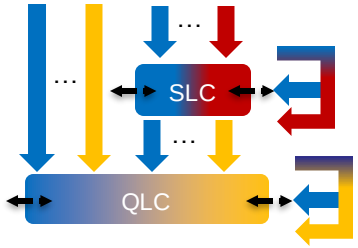
- Random write workload to 20% of the address space. Remaining utilized space holds static data.
- Controller with adaptive SLC cache:
  - All data first written to SLC, then evicted to QLC
  - Controller leverages write heat information
- Controller with SLC/QLC tiers
  - Data written to either SLC or QLC
  - Controller leverages write heat information
- Vary relative SLC/QLC endurance



Benefits of bypassing the SLC cache grow as QLC endurance improves

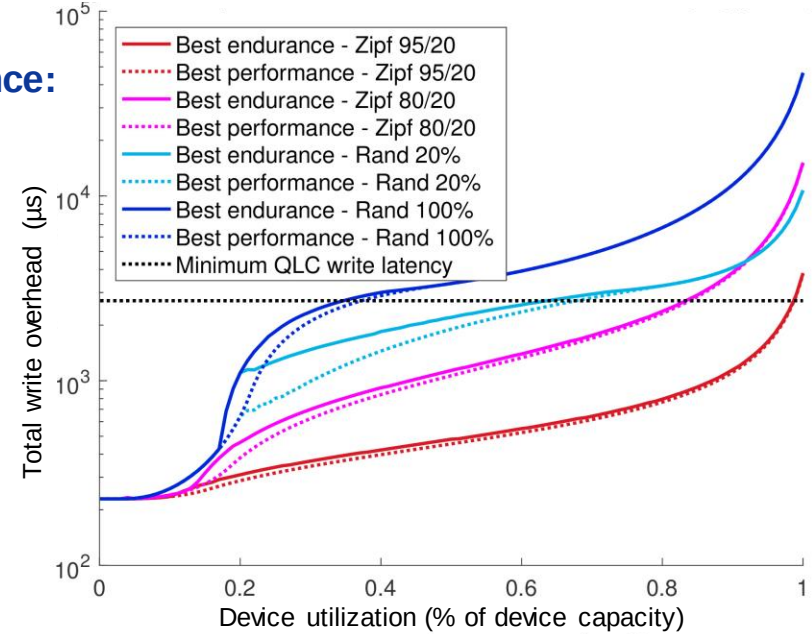
Performance vs. endurance

Comparing the write overhead for a controller that maximizes endurance vs. maximizes write performance:



Experiment:

- Using the same controller architecture but changing the optimization target
- Workloads with varying amount of skew
- Compute the average write overhead, i.e., the busy time required by to service one user write
  - Indicative of the sustained write throughput but **not** of the user experienced latency.



A hybrid controller can achieve both endurance and performance at the same time

Conclusion

- Move to QLC leads to many reliability, performance & HW controller design challenges
- Hybrid controllers that use dynamic block mode shifting can bridge many of these issues
- An appropriate hybrid controller design can achieve significant endurance & performance
  - Fixed-sized SLC destage buffers only achieve marginal endurance improvements
  - Latest generation SSD controllers that only adapt SLC size to device utilization are underperforming
- New approaches are fundamental to maximizing the benefits of a hybrid controller
  - Leverage workload properties to improve write heat segregation, data placement, tier sizing
- Combined with other existing Flash management techniques, we can achieve enterprise-level endurance & performance
- **More details in upcoming paper: Stoica et al., *"Understanding the design trade-offs of hybrid flash controllers", MASCOTS'19***

Flash Memory Summit

Thank You !

Questions ?

www.research.ibm.com/labs/zurich/cci/