REGIONAL SDC 25

BY Developers FOR Developers

Regional SDC Denver
April 30, 2025

From Standards to Practice: Implementing Effective QoS Control in NVMe® SSDs

Dan Helmick, PhD
Samsung Semiconductor

SNIA®

Agenda

- A Useful QoS framework for NVMe® SSDs
- SSD Implementation of Quality of Service (QoS)
- Future Host and SSD expectations for interoperability

TP4176 “Quality of Service for PCIe Bandwidth and IOPS for a Controller” Status

- TP4176 is in the early stages of development
  - Architectural design for the feature is in discussion
  - Specification development has not yet started

- This is an independent pre-standardization presentation based on the speaker’s knowledge & experience. These inputs will be provided to help shape TP4176’s development.

- Join NVMe to influence the feature’s development!

Overview of 2 Useful QoS Modes

- Rate Limit Mode
  - Rate limit IOPS and BW for Total, Write, and TRIM per Controller
    - Each command consumes tokens for both IOPS and BW before proceeding.
    - Ex: Writes need to consume from both the Total and Write buckets
      - 4 Token buckets would be examined for a Write to proceed
  - Writes
    - Use a constant Read/Write scale factor per SSD
    - May additionally integrate a WAF scaler
      - Potentially dynamic

Symbol for Rate Limit Mode:

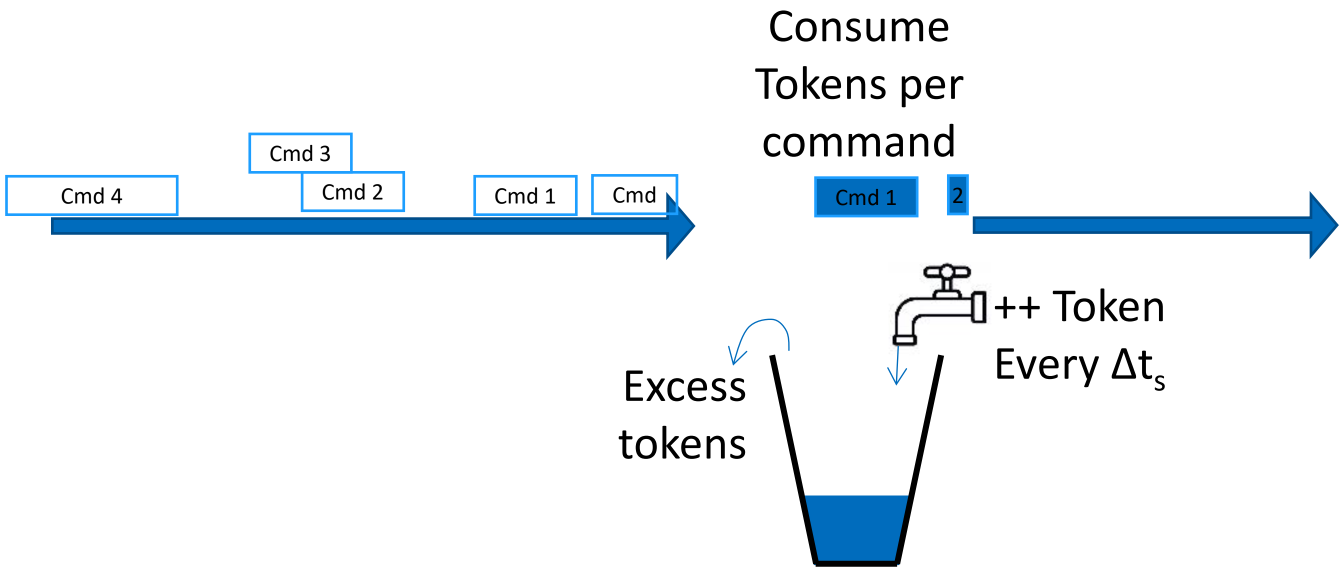
- Priority Mode
  - Targeted at reducing Head of Line Blocking (HoLB)

Priority Marked with:

URGENT

Token Buckets Can Implement Rate Limiting Mode

- Start with a bucket containing an available number of tokens
- Tokens added at a constant rate
- Excess tokens overflow and are lost
  - Cap on the quantity of tokens possible
- Arriving commands check for available tokens
  - Consume those tokens to proceed
- Commands lacking sufficient tokens are queued
  - SSDs implement Traffic Policing
  - No lost/dropped commands
- Partial command progress with partial token consumption is acceptable

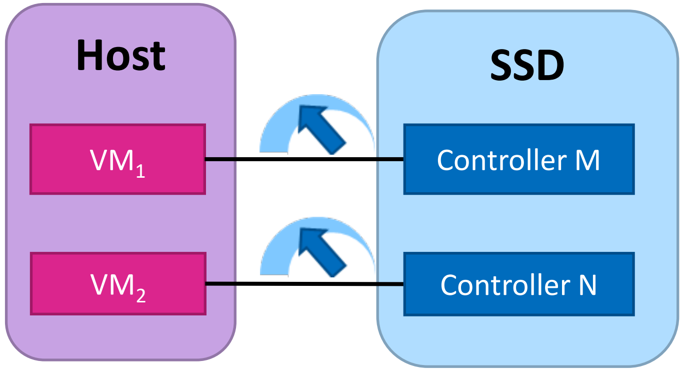


Note: Token bucket and Leaky bucket are different implementations/visualizations, but they can be translated between each other. link

Rate Limit Mode Example with Desired Behavior

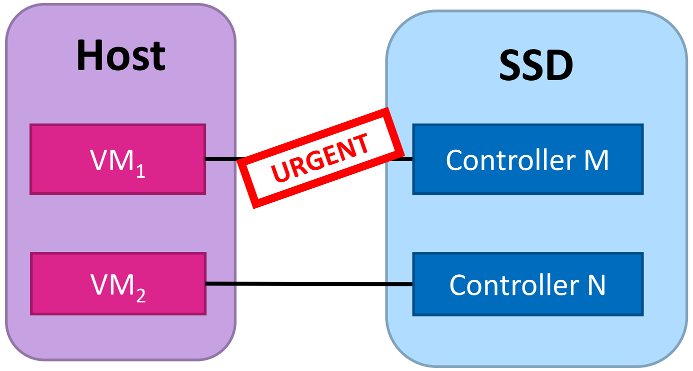
- Example 1: 1 Tenant Active
  - VM_2 is idle
  - Throttle VM_1 to 75% of drive's performance

- Example 2: 2 Tenants Active – Overprescribed SSD
  - Allow 75% of drive's performance for both VMs
  - Both VMs are active
  - Each VM shall receive 50% of drive's performance



Example Priority Mode Set-up

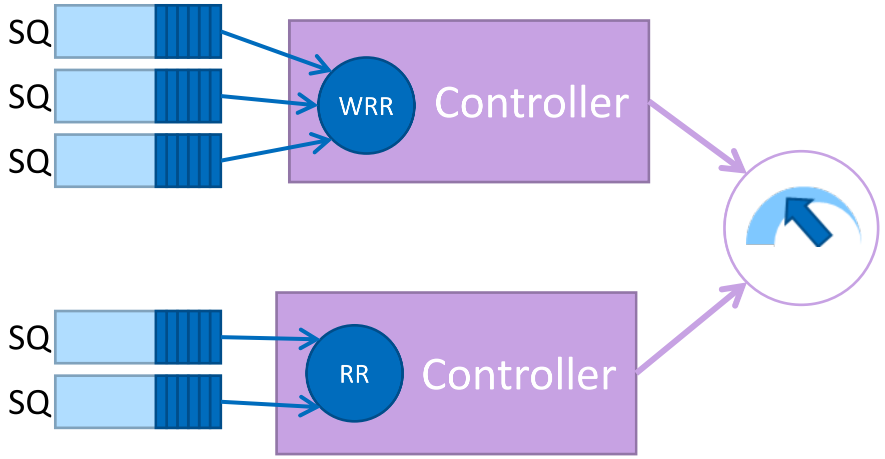
- Potential System Set-up
  - VM_1 requires high priority for short bursts of time
  - VM_2 more constant activity
  - Example uses:
    - VM_1 is a high paying AI customer with latency assurances of inference results, and VM_2 is internal company users.
    - The Host is a File System. VM_1 is the end user, and VM_2 is the FS traffic.



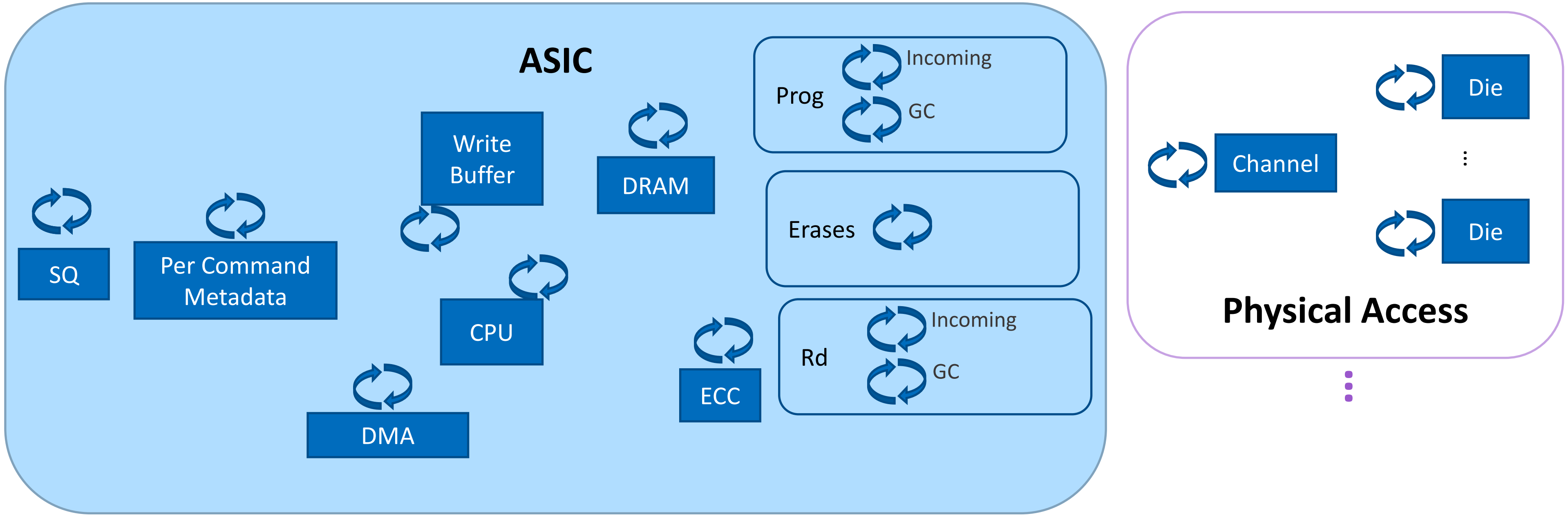
- Example Priority Mode Goal
  - VM_1 latency difference may be minimized when comparing
    - Idle VM_2
    - Active VM_2

Can Controller QoS be Integrated with Existing SQ Fetch Standards?

- Multiple Controller Behavior
  - QoS may determine which Controllers are allowed to fetch SQEs
  - Each Controller independently decides which SQ and how many SQEs to fetch
    - Adhere to existing standards for SQ fetching
    - Available Command Slots, Bursts, etc are all problems that continue to be managed by the Controller without change
  - Fetching of more than 1 controller may be interleaved as allowed by the transport if sufficient tokens
- Enables tiered and separated decisions by the SSD
- Some Reasonable Usage Recommendations
  - Weighted Round Robin (WRR) – Risks to impact QoS settings if done without care
  - Round Robin (RR) – Likely most robust expectations with repeatable testing results



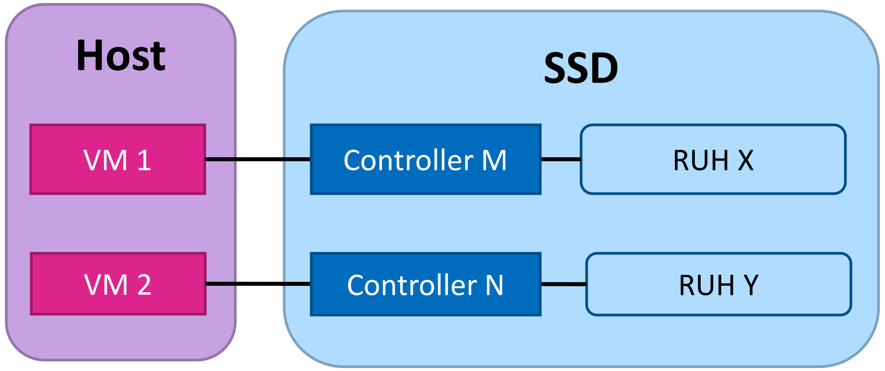
Implementing QoS in Real SSDs



- There are many potential constrained resources in an SSD
  - Bottlenecks change per workload
  - It isn't profitable to over design products for no reason.
- **Potential Idealized Goal:** QoS parameters are communicating media access targets

Reasonable QoS Write Scaling Choices

- Total = Read + Write + Deallocates
  - Writes and Deallocates must be scaled for "Total" to make sense
  - Example:
    - Writes can be scaled by time per bit of Program to Read ratio
    - $Wr_{Scaler} = \alpha * T_{Program} / T_{Read}$
    - Achieves media access relationships
  - Only works with Sequential Writes
- Extending to Non-Sequential Writes
  - WAF_{Scaler} determined by the drive
  - WAF_{Scaler} = constant representative of nominal Write traffic characterization
  - WAF_{Scaler} = proportional to Controller's WAF
    - May require the association of FDP RUHs per Controller
  - Other solutions possible



Some QoS Complexities

- Deallocates (TRIMs)
  - Are DRAM and CPU bound operations in most SSDs
    - Interactions with Reads and Writes can be very complex
  - May be executed in foreground or background
  - May have nonlinear performance variations depending on TRIM length
  - **Potential Reframed Goal:** Rate limit Deallocates for proportional impediment to media access for Reads and Writes
- Transient Workloads
  - Examples:
    - 70/30 transitioning to 30/70
    - Sequential transitioning to Random
    - Bursty workloads (idle interleaved with periods of QD4-QD32)

- Writes misaligned to Indirection Unit (IU)
  - These Writes will cause RMW
  - Example:
    - Small Writes
    - Offsets of Head and Tail
  - Impacts are more common with increasing SSD capacities and QLC SSDs
- How do we resolve or tolerate QoS Complexities?
  - Expect discussions on these topics in NVMe

Using QoS Parameters in Practice

- Standardized QoS Parameters
  - Will need to work across all SSDs (vendors, generations, client/enterprise, capacities, etc.)
  - Must be simple enough for a poorly informed Host to use meaningfully
  - Therefore, they are going to be simplified from perfect parameters

- Recommended Customer Actions
  - Identify a small representative subset of target workloads
  - Describe the test environment – Enclosure, CPU settings, etc.
  - Set QoS performance requirements with acceptable variations for Customer Quals

- Recommended SSD Vendor Actions
  - Design for target workloads
  - Examine sensitivity to variations in settings and workloads
  - Confirm bounded SSD behavior during transitions

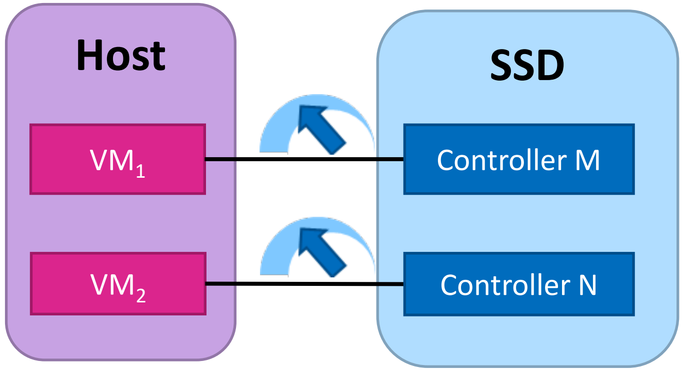
Example Rate Limit Mode – Customer Qual

- Example 1: 1 Tenant Active
  - VM_2 is idle
  - Throttle VM_1 to 75% of drive’s performance

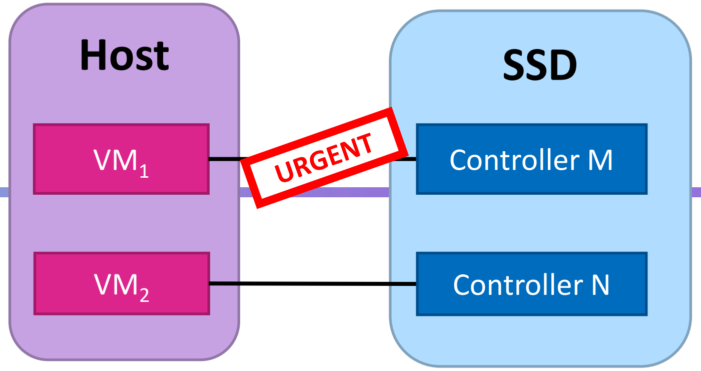
- Measure 100% perf
- Enable QoS
- Measure perf
- Accept within bounds of 75%

- Example 2: 2 Tenants Active – Overprescribed SSD
  - Allow 75% of drive’s performance for both VMs
  - Both VMs are active
  - Each VM shall receive 50% of drive’s performance

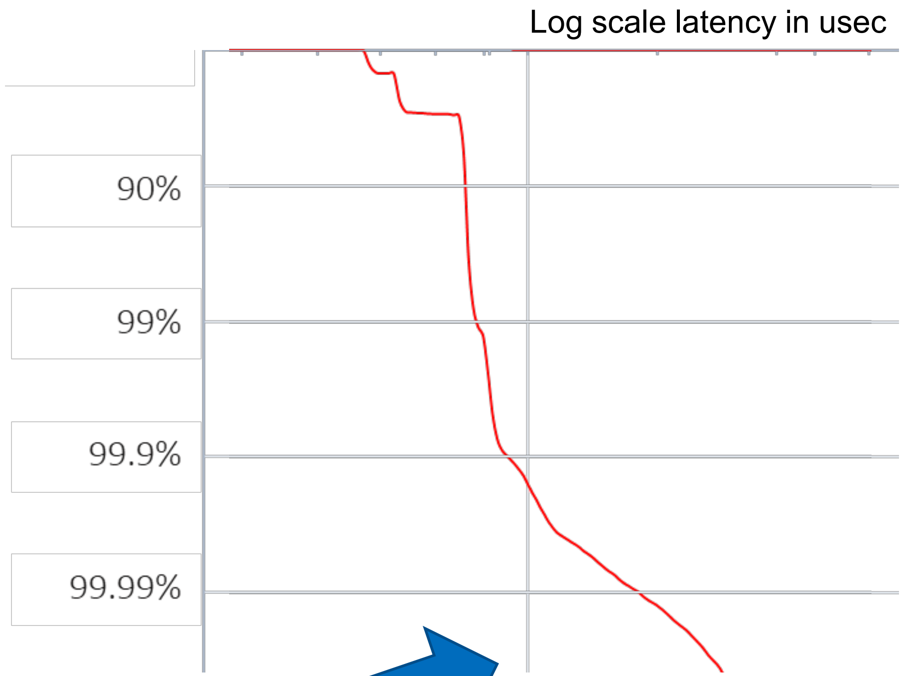
- Enable QoS
- Measure perf
- Accept within bounds of 50% average and variation of median latency



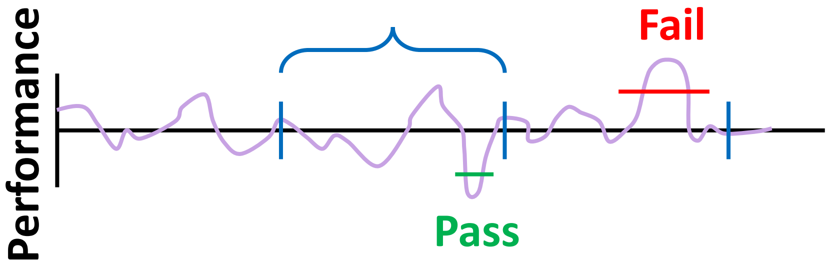
QoS Priority Mode Qual Examples



- Example 1
  - Workload: VM$_1$ QD4 Random Reads; VM$_2$ QD128 70/30
  - Example Goal: VM$_1$ 99.9% latency degrades by no more than 50%
- Example 2
  - Workload: VM$_1$ 128 Random Reads submitted every 1 second; VM$_2$ QD128 Seq Wr
  - Example Goal: VM$_1$ 99.999% latency stays below 10ms
- Measuring Variations
  - Recommend: Measuring in 9's
    - Scales for every SSD performance without re-examination every generation
    - SSDs can propagate to internal design targets
  - Discourage: Variations over time
    - Must set variation detectability bounds
    - Peak excursion vs detectability bounds should additionally be spec'ed
    - Should examine measurement period with every processor and SSD generation



Translations exist



Conclusions for QoS in Practice

- TP4176 “Quality of Service for PCIe Bandwidth and IOPS for a Controller”
  - Will be a game changer for enabling the sharing of large capacity SSDs
  - Enables an SSD to differentiate traffic per host tenant like never before

- QoS parameters will be simplified – Do not expect perfection
  - Providing nominal workloads enables SSDs to test against goals with variations
  - Unexpected results if operating an SSD far outside of design goals
    - Extreme QoS parameters
    - Extreme workloads

- Latency
  - Latency quantifications increase in importance during Customer Quals
  - Providing latency targets in numbers of 9’s is more scalable, and it translates into internal design targets in the design of SSDs.