



Flash Memory Summit

SSD Performance Shaping

York Chen

Senior Marketing Manager of Enterprise SSD Controller
Silicon Motion Technology Corp.

Legal Notice and Disclaimer

- Nothing in these materials is an offer to sell any of the components or devices referenced herein.
- The content of this document including, but not limited to, concepts, ideas, figures and architectures is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Silicon Motion Inc. and its affiliates. Silicon Motion Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this document.
- Silicon Motion Inc. may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Silicon Motion, Inc., the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.
- © 2022 Silicon Motion Inc. or its affiliates. All Rights Reserved.
- Silicon Motion, the Silicon Motion logo, MonTitan, the MonTitan logo are trademarks or registered trademarks of Silicon Motion Inc.



Why Performance Shaping matters in SSD?



What SSD Performance Shaping work?



How: Modelling and Simulation

Performance Shaping: Problem Statement

One challenge to QoS for multi-tenancy SSD is inconsistent tenancy behavior. Noisy tendency may impact QoS of other tenancies who behaves consistent. Isolation is needed, BUT:

- Restrict isolation (share nothing) has problems:
 - Difficult to implement: can't physically divide/isolate all kinds of resources in the device, into small independent pieces.
 - Even if you do so, leads to fragmentation and waste.
- NVMe provides submission queue arbitration mechanism based on weighted round robin (WRR) priority with urgent priority class inside an NVMe controller. However:
 - Only 4 level of priorities/weights.
 - Only on the submission queue level, not in IO command level with performance parameters (IOPS, or throughput (GB/S)) as weights.
 - No mechanism for arbitration between NVMe controllers on an NVMe subsystem which supports multiple PCIe ports and functions.

Agenda



Why Performance Shaping matters in SSD?



What SSD Performance Shaping work?



How: Modelling and Simulation

Performance Shaping Mechanism

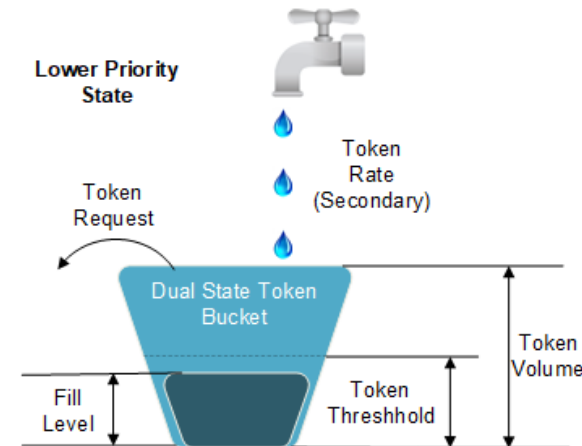
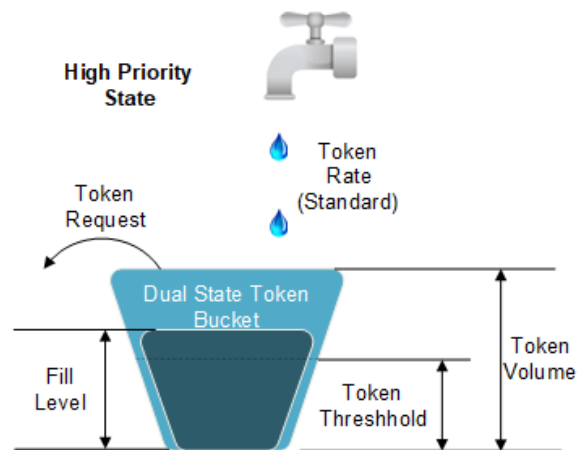
A mechanism to shape IO requests per **QoS set**. The shaping algorithm is based on **Dual State Token Bucket algorithm**.

- A QoS set is a group of one or multiple host tenants, and/or internal tasks (reclamation, etc.), which initiates IO type operations.
- Each QoS set is assigned with a token bucket:
 - One token is a permission for an IO cmd, or some amount of KiB's.
 - Token rate: at which speed tokens fill the token bucket, configurable and variable.
 - Token volume / bucket size: max token number the token bucket can hold.
 - When a QoS set / client requests n tokens:
 - If the bucket has $\geq n$ tokens, grant permission to go.
 - Otherwise, the request waits until the bucket accumulates enough tokens.
 - Token threshold : see Dual-State Token Bucket Algorithm, next page.

Performance Shaping Mechanism Cont.

• Dual-State Token Bucket Algorithm:

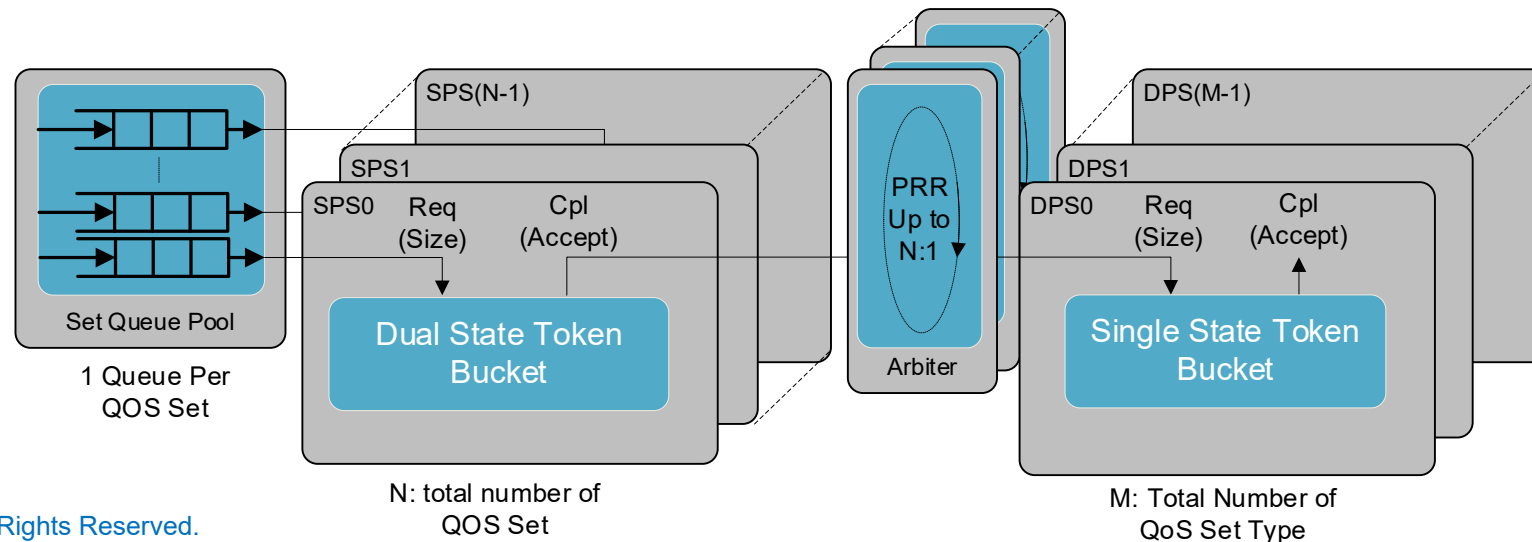
- Purpose: dual rates to allow the client to request more but given lower priority, processed opportunistically.
- Token fill level \geq token threshold: the token rate will be a standard token rate, and any token request will be accepted with high priority.
- Otherwise, the token rate will be set to a secondary token rate ($>$ standard token rate), and any token request will be accepted with low priority.



Performance Shaping Mechanism Cont.

Two-Stage Shaping

- Token bucket shaping smooths IO requests and limits its outliers to certain extent. The dual state token bucket algorithm allows more IO burstiness, in order to optimize the utilization of the device bandwidth.
- However, the device bandwidth is limited. When we have multiple noisy/demanding tenants, we need to make sure the device is not over-booked. Thus, we propose a second stage token bucket, namely Device Level Token bucket:
 - Simply one-state token bucket with a token rate = device bandwidth
 - Can have multiple of it used for different type of IO performance controls, e.g. IOPS, throughput (GB/S), read and write, etc.



Agenda



Why Performance Shaping matters in SSD?



What SSD Performance Shaping work?



How: Modelling and Simulation

Performance Shaping Modeling

Goal of Performance Shaping Demo through Modeling

- ☐ Smooth out fluctuations
- ☐ Isolate noisy neighbors
- ☐ Fully utilize the system bandwidth

Key Components

- ☐ Host Workload Generator
- ☐ Simulator
- ☐ Output Analysis

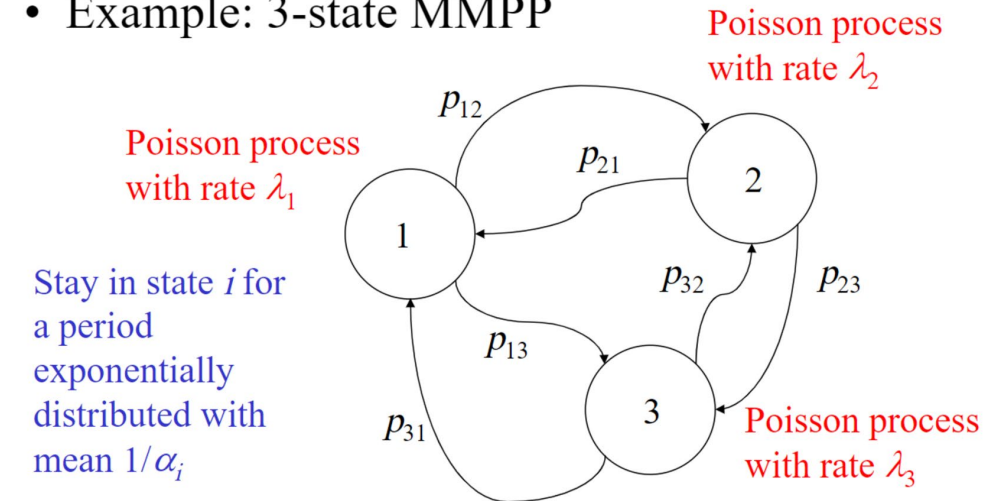
Workload Generator



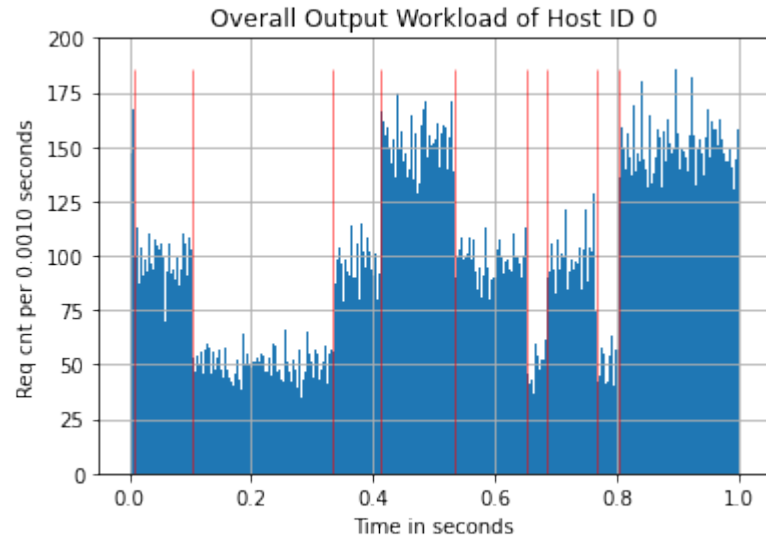
Flash Memory Summit

- Target:
 - To emulate a host application that:
 - Multiple internal states and transition among these
 - Each internal state has its own IO rate that follows Poisson process
 - → MMPP (Markov-modulated Poisson process)
 - Poisson processes by N, each with its own rate.
 - Continuous Time Markov chain (CTMC): $N * N$ matrix
- Tool:
 - Python Random
 - Exponential Random Var:
 - Generate Poisson processes
 - Determines the time to stay in one state
 - Random Choice Var: choose the next state
- Output:
 - Trace: List of (NLBA, time)
 - NLBA = 1 for the purpose of evaluating IOPS

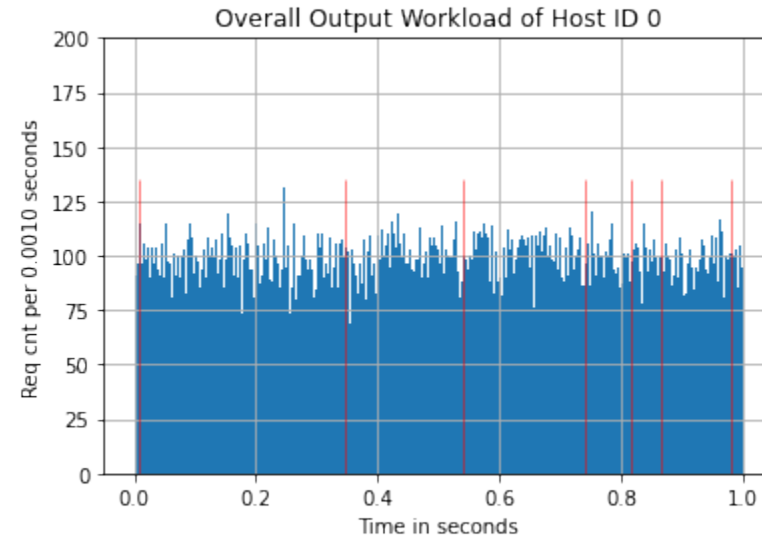
- Example: 3-state MMPP



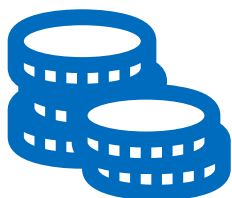
Workload Examples



- Poisson's: 100K/150K/50K
- Noisy neighbor



- Poisson's: 100K/100K/100K
- Good neighbor



Shaping Engine: Token Buckets + Arbiter

Bucket size:

- How much token to save for peaks

Token count threshold:

- If tokens are used up quickly (peaks), switch to high rate but mark as low_priority

Two token rates:

- Normal rate: \approx the average workload rate
- High rate: allows peaks to pass through



Tool:

Simpy (a Discrete Event Simulator in Python)



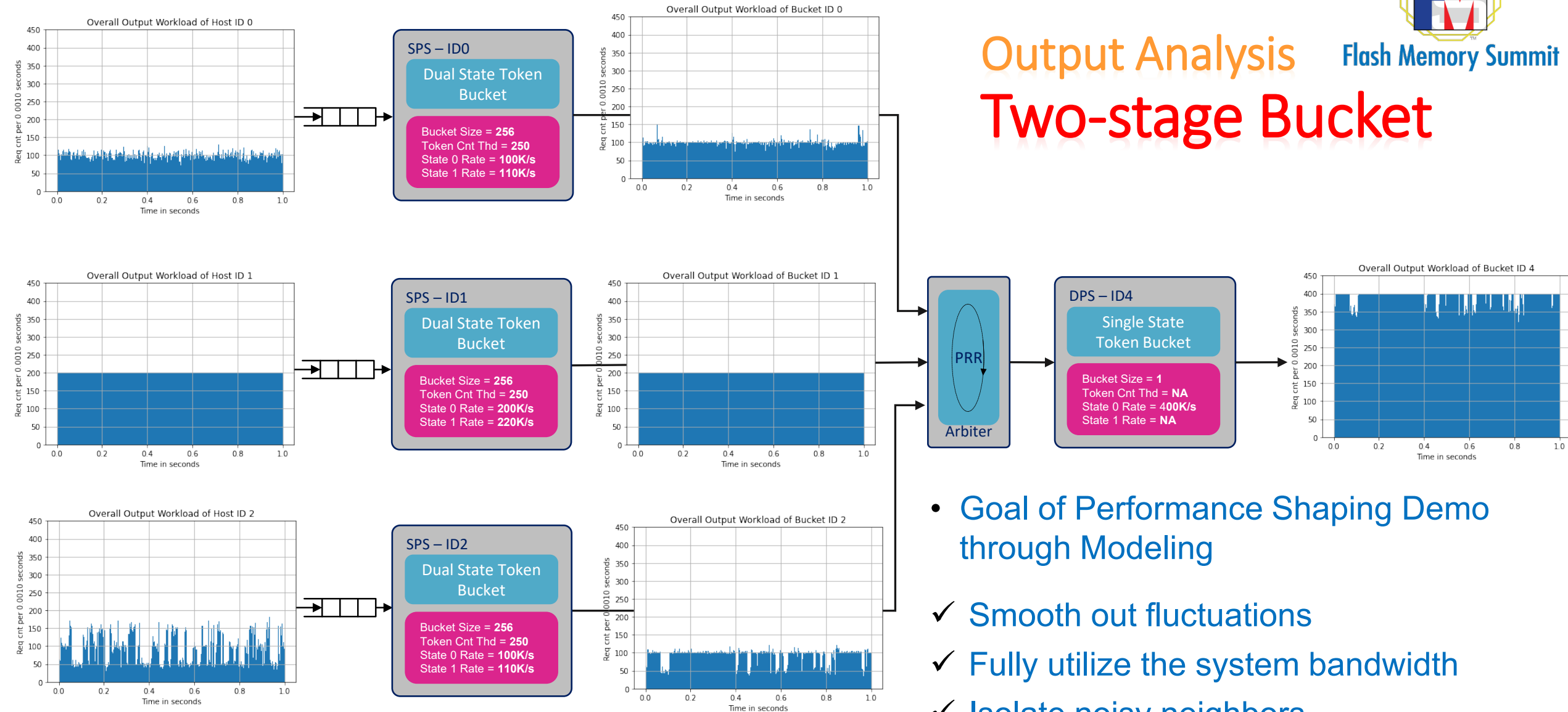
Output:

List of (NLBA, time, priority)



Output Analysis

Two-stage Bucket



- Goal of Performance Shaping Demo through Modeling
- ✓ Smooth out fluctuations
- ✓ Fully utilize the system bandwidth
- ✓ Isolate noisy neighbors