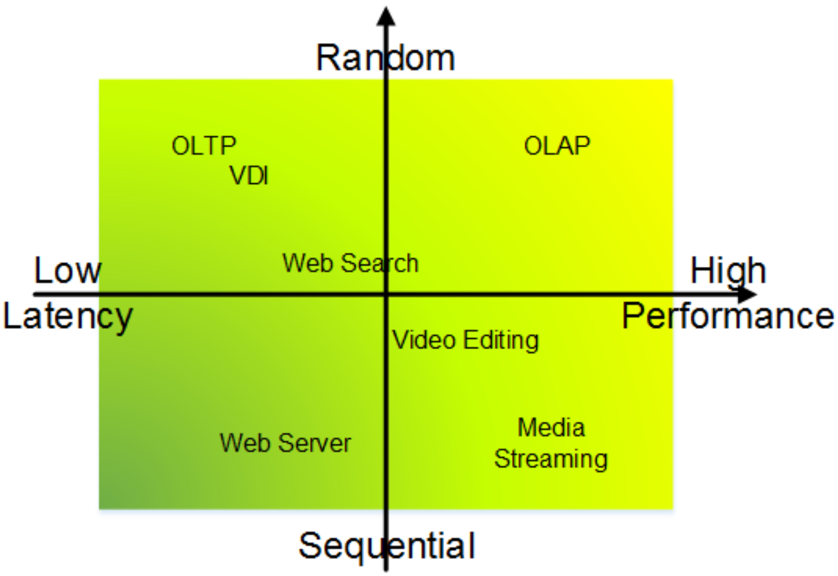
Optimizing SSD Latency and Performance in Enterprise Applications

Xiangfeng Lu | CTO, Memblaze

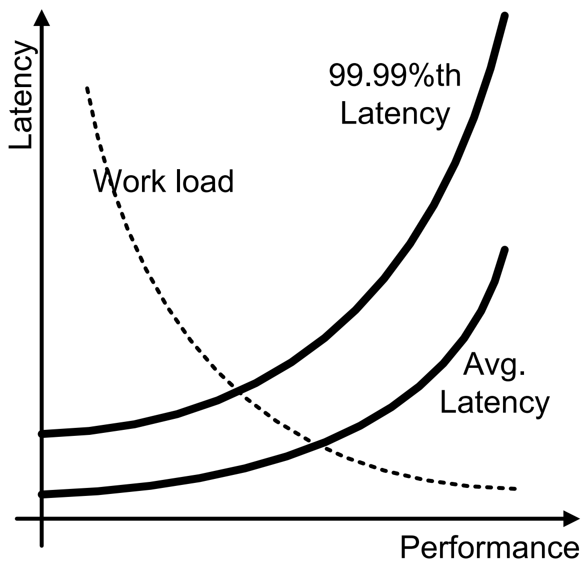
Presenter: Taile Zhang | VP, Memblaze

Enterprise Application IO Requirement

- Enterprise applications have different IO statistics and different IO service level agreement.
- SSD vendors, system operators and application designers can leverage this characteristics to optimize overall behavior.



SSD Latency and Performance



- IO latency and IO performance are interrelated.
  - Application work load curve is determined by application IO concurrency status and workload's dependency on IO completion.
  - SSD work load curve is determined by SSD media and firmware behaviors, IO access pattern, over-provision, etc.
  - Cross point is the real work point for this system. For QoS driven applications, 99.99%th latency may be more important.

SSD Performance & Latency Joint Optimization

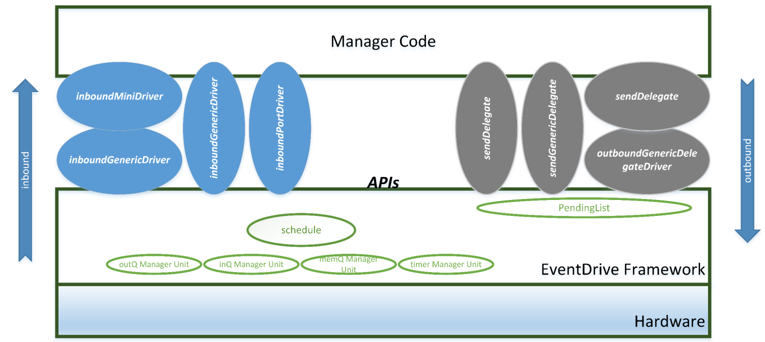
- Optimization targets

Usage Scenario	Example	Requirement
Small block read intensive	Read cache, web search	Random read high bandwidth, read IOs have higher priority.
Read latency sensitive	Database service	Low read and write latency. Stable read latency for fluctuated write performance.
Write intensive	Webserver log	Sequential write high bandwidth, stable performance.

- Optimization approaches
  - IO process flow optimization in firmware.
  - Utilize NAND flash data low latency access commands.
  - Improve garbage collection efficiency.

IO Process Flow Optimization

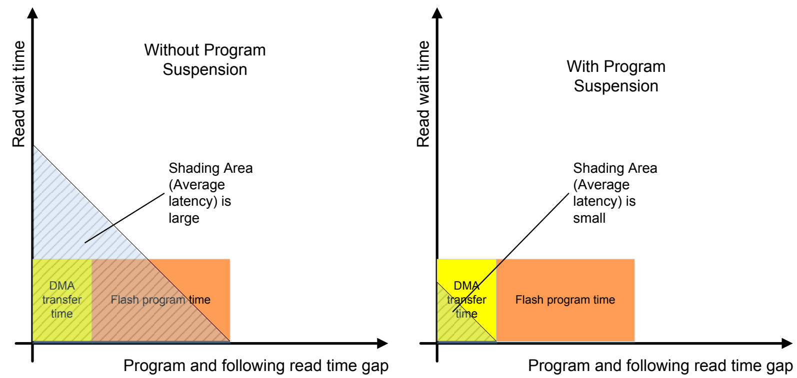
- IO execution time in firmware should be proportional to IO performance. Reduce non-IO execution time.
  - Use event driven framework to lower non-IO execution time and improve IO execution efficiency.
  - Use event priority to optimize key task execution responsiveness. Use fair scheduler to optimize QoS.

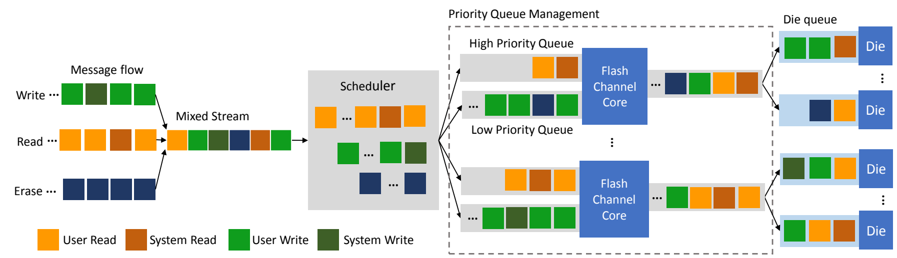


Workload	Baseline	Optimized Arch	Gap
Sequential Read (MB/s)	3168	3472	9.5%
Random Read (k IOPS)	808	916	13.3%

Optimize Read Latency

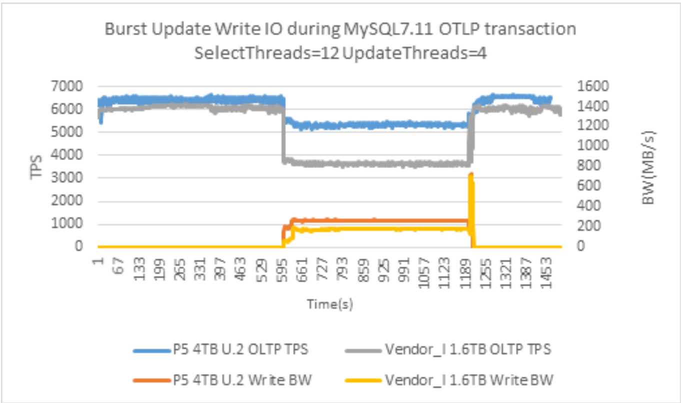
- NAND flash data low latency access commands.
  - NAND flash can be read in fast mode (Such as SNAP read)
  - Program suspension can lower read latency significantly. This should be done with policy so that suspension does not hurt flash data integrity.
- Priority queue management.
  - With priority queue management, received read requests can be re-ordered to high priority queue.

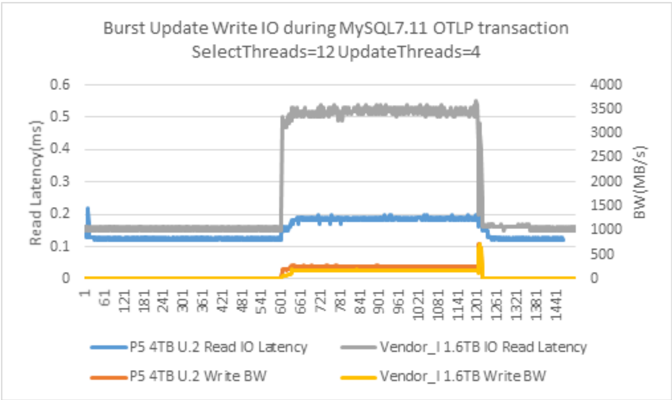




Optimize Read Latency (Cont'd)

- Random read throughput and latency with and without QoS enable.



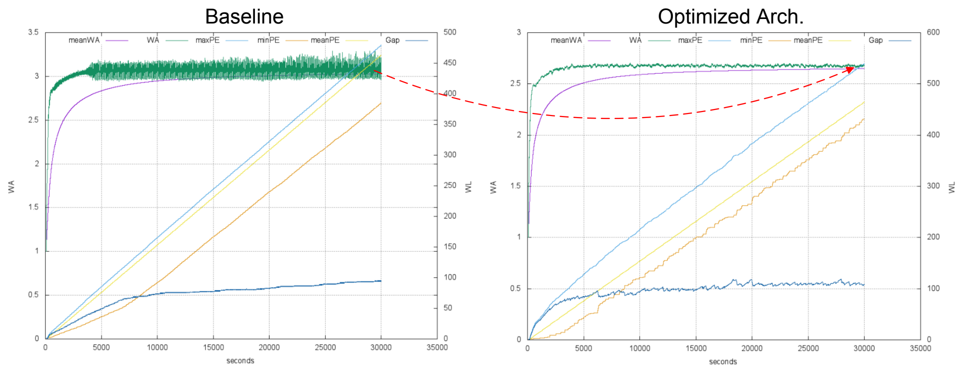


Optimize Write Latency

- Use of cache to optimize write latency.
  - Consider IO location affinity - Combine neighbor IOs.
  - Consider IO temporal affinity – Merge consecutive IOs.
- Parallelize hardware and firmware execution.
- Batch background IO execution to improve IO process efficiency.

Optimize GC & Wearleveling Efficiency

- GC & Wearleveling affects IO performance and latency indirectly by consumes DMA/ flash operation bandwidth and CPU computation capabilities.
  - Improve GC efficiency so that write amplification remains low in changing work loads.
  - Reduce wearleveling overheads in GC process.
  - Adaptive GC policy so that learning and adaption is possible.



Work Load: 90% random write bandwidth on 10% LBA range and 10% random write bandwidth on 90% LBA range.

Conclusion

- Enterprise applications have variable demands for SSD's IO performance and latency. SSDs can leverage software engineering and system level knowledge to optimize these targets. Flash domain knowledge can also be used to improve overall effects.
- SSDs may have different work load curves, system owners should choose most appropriate configurations to trade off performance and latency targets.

Please visit our booth at #523.

Taile Zhang

VP, Memblaze