Solid State Storage in Massive Data Environments

Erik Eyberg

Senior Analyst

Texas Memory Systems, Inc.

Agenda

- Taxonomy
- Performance Considerations
- Reliability Considerations
- Q&A

Solid State Storage Taxonomy

In-Server

- PCIe SSDs
  - PCIe card form factors
  - Sometimes act like a HDD RAID controller, sometimes more direct to Flash
- Disk interface SSDs
  - 2.5″ or 3.5″ form factors
  - Commodity controllers
  - Act like hard drives

Shared

- SAN SSDs
  - Fibre Channel, InfiniBand, or Ethernet (iSCSI)
  - Block-level access
- NAS SSDs
  - Ethernet (NFS, SMB/CIFS)
  - File-level access
- Shared PCIe and custom interface SSDs

PERFORMANCE

Workload Segmentation

- Metadata, Working Data, Archived Data
- Metadata is typically accessed the most, but takes up the least space
- Archived data is accessed the least, but takes up the most space
- Moving high-access data into a high-performance medium has the greatest impact

But the question is: what data makes sense to store on SSD?

Application Profiles

Low CPU Utilization + Low I/O Wait	=	Bad algorithm?
Low CPU Utilization + High I/O Wait	=	**Great fit for SSD!**
High CPU Utilization + Low I/O Wait	=	Put it in RAM
High CPU Utilization + High I/O Wait	=	Use asynchronous I/O Add disks for growing capacity Add SSD for same size capacity

Keys to Storage Performance

- Hardware in data path
  - FPGA & Hardware Logic
  - Faster than software-shared memory
- Software cannot add performance
  - Virtualization allows you to get away with less hardware, but it’s another layer to utilizing additional hardware
  - QoS is a software overhead to give applications priority over another on shared hardware

$$L = \lambda W$$

The long-term average number of customers in a stable system ***L*** is equal to the long-term average effective arrival rate, *λ*, multiplied by the average time a customer spends in the system, ***W*** [1]

Above is Little's Law which is just a fancy way to say that performance is based on **Latency** and **Parallelism**

[1] Paraphrased from *Little's Law*, John D.C. Little and Stephen C. Graves, MIT

$$L = \lambda \mathbf{W}$$

So, what else influences **Latency** and **Parallelism**?

$L = \lambda W$

What influences **Latency**?

- CPU Speed
  - **not** number of cores
  - **not** number of chips
- Bus architecture
  - North/south bridges
  - PCIe hierarchy
  - PCIe controller
- CPU Usage (so in a convoluted way, cores and chip counts do matter)

$L = \lambda W$

What influences **Latency**?

- Operating system and file system
  - OSes and file systems optimized for disks tend to count on slow data access to hide processing
  - Add schedulers, I/O elevators, etc to compensate for slow random access times
  - Modern OSes and file systems are now written to maximize SSD
- Driver: bridge between the OS and the hardware
  - Must be thin to decrease additional latency
  - Linux, Windows, Solaris, VMware, OS X, AIX, etc
- If measured at the application layer, middleware (for example, databases) can inject latency

$$L = \lambda W$$

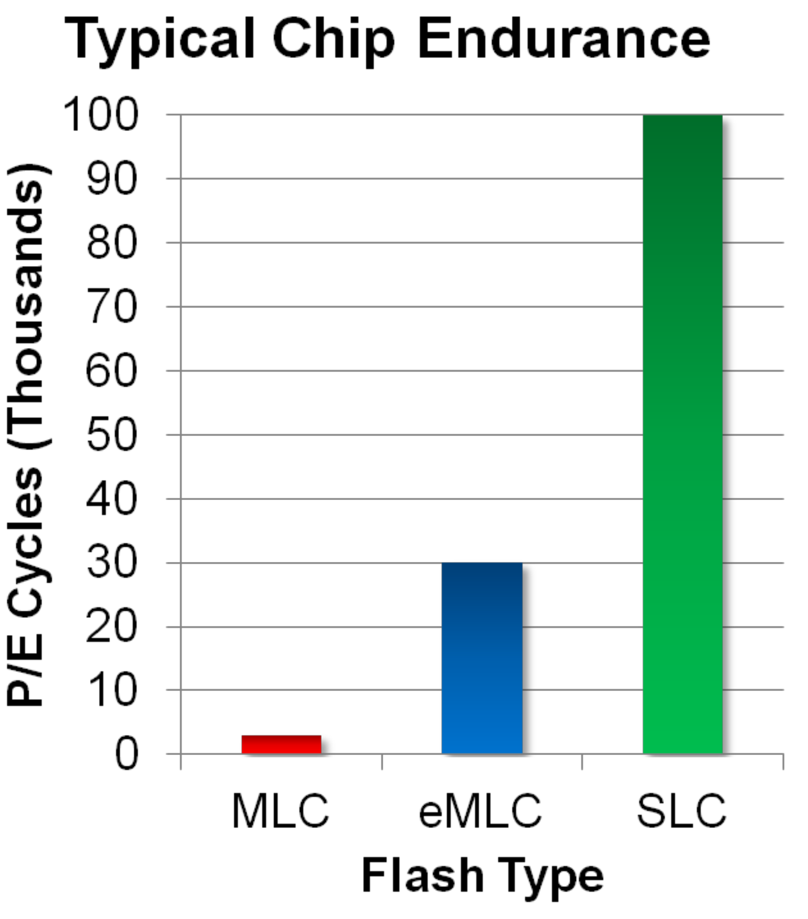
What influences **Parallelism**?

- Chunk size
- Threading: most applications either have multiple threads of synchronous I/O or a single thread that allows multiple outstanding asynchronous I/Os
  - Most high-performance middleware does just this (Microsoft SQL Server, Oracle, etc)
- Multiple applications at the same time look similar to a single application with multiple threads
  - CPU becomes more and more of a bottleneck, however—more context switching overhead

RELIABILITY

Flash Quality

- Flash type matters!
- SLC is best but most expensive/least dense
- eMLC chips last 10x longer vs. normal MLC
  - And cost about 25% more
  - Tradeoff: slower P/E times
- Failures <u>will</u> happen! How does your vendor deal with them?



Know Your Endurance!

- System endurance is calculated:

$$\frac{Flash\ Capacity \times Flash\ Quality}{Media\ Write\ Bandwidth}$$

Endurance Examples

5 TB RamSan-710 (SLC Flash)

$$\frac{5TB \times 100{,}000}{1\,GBps} = 15.8\,Years\ Endurance$$

10 TB RamSan-810 (eMLC Flash)

$$\frac{10TB \times 30{,}000}{1\,GBps} = 9.5\,Years\ Endurance$$

eMLC or (c)MLC?

- eMLC: 2x capacity for SLC cost, 30% of endurance
- MLC has 10x less endurance than eMLC
- MLC costs 25% less than eMLC
  - Sustained writes do not make sense for MLC
  - MLC will last less than a year from sustained writes at same cost and half the write workload

$$\frac{1TB \times 3{,}000}{500\ MBps} = Less\ than\ a\ year$$

Reliability Summary

- Flash is a consumable
- Two major factors:
  - How many writes?
  - How many years?
- eMLC is typically a better value than cMLC for long-term installations
- Don’t fall into the trap of “it works now”—know what will happen in *x* years

Thank you!

Questions?