



Education

LEVERAGING FLASH MEMORY in ENTERPRISE STORAGE

Luanne Dauber, Pure Storage

Author: **Matt Kixmoeller**, Pure Storage

- ◆ The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.
- ◆ Member companies and individual members may use this material in presentations and literature under the following conditions:
 - ◆ Any slide or slides used must be reproduced in their entirety without modification
 - ◆ The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- ◆ This presentation is a project of the SNIA Education Committee.
- ◆ Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- ◆ The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

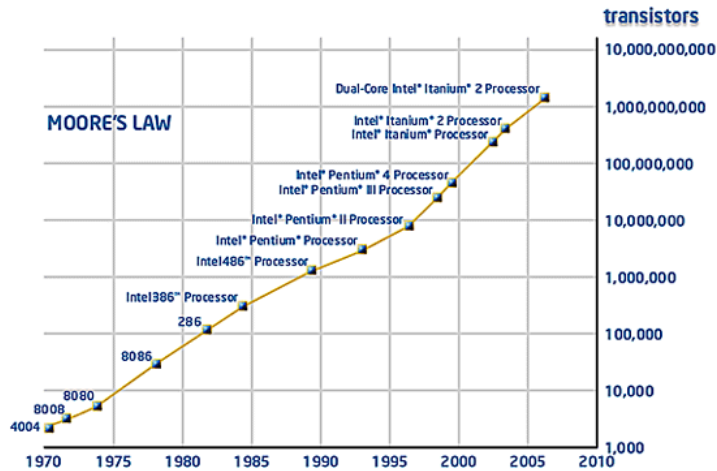
➤ Leveraging Flash Memory in Enterprise Storage

- ◆ This session is for Storage Administrators and Application Architects seeking to understand how to best take advantage of flash memory in enterprise storage environments. The relative advantages of flash tiering, caching and all-flash approaches will be considered, across the dimensions of performance, cost, reliability and predictability.

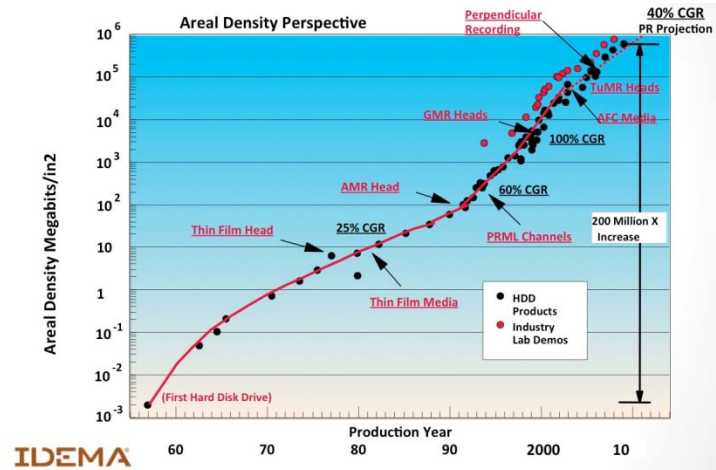
- The Storage I/O Crisis
- How to Evaluate Flash Solutions
- Approaches for Leveraging Flash
 - ◆ PCI cards
 - ◆ Flash caching in arrays
 - ◆ Flash tiering in arrays
 - ◆ All-flash arrays
- Analyzing Flash ROI

Moore's Law vs. Newton's Law

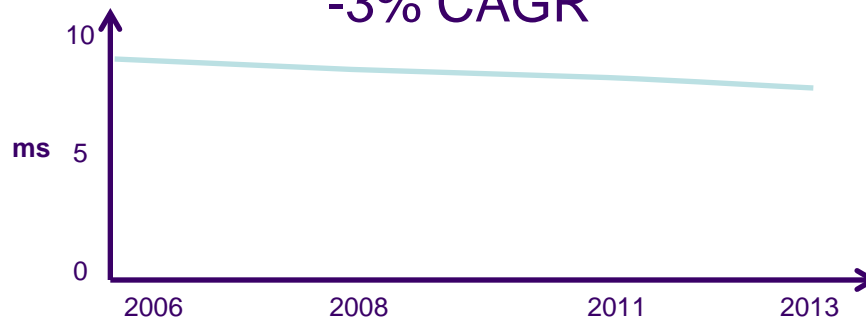
Moore's Law: 58% CAGR



HDD Areal Density: 40-100% CAGR

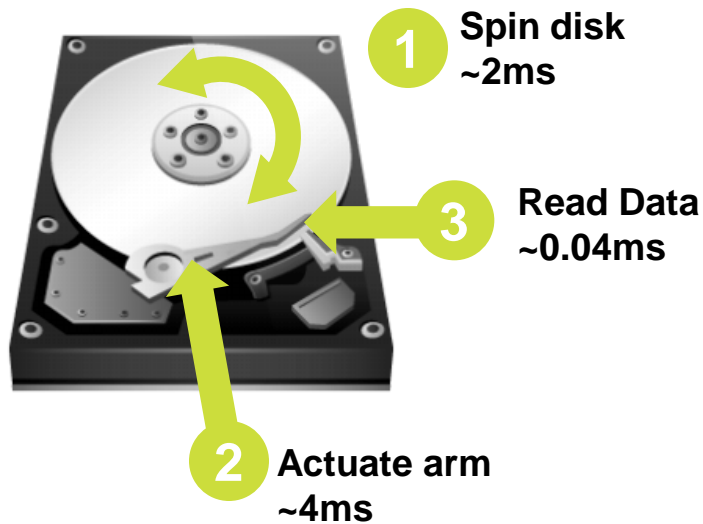


HDD Latency (Seek Time) -3% CAGR

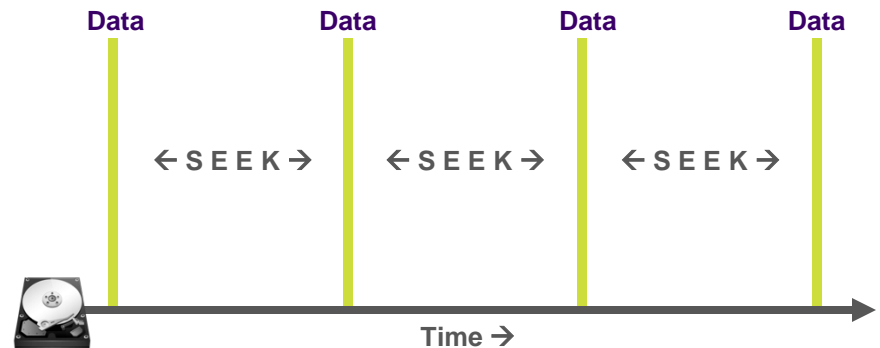


Rotating Disk Challenges

Deconstructing a Random Read

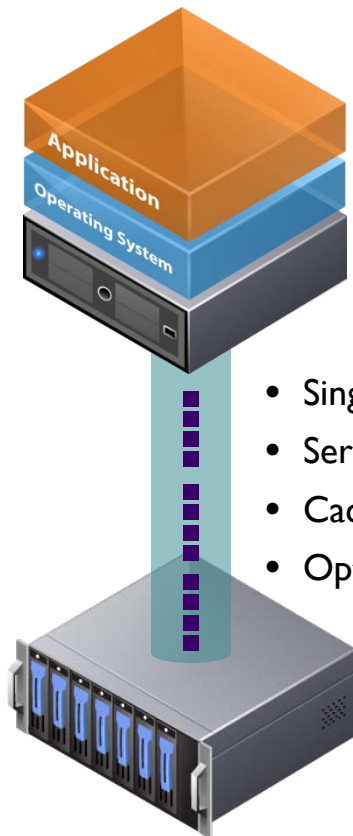


Typical Resulting Duty Pattern:



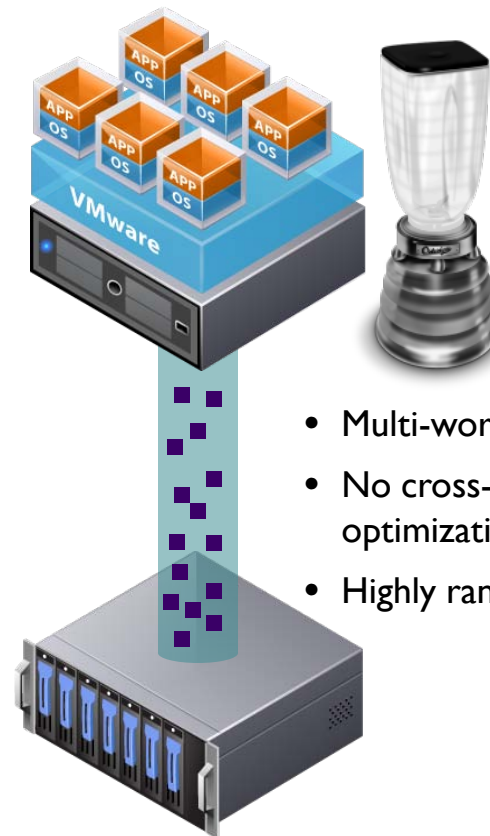
Disks spend >95% of their time seeking and rotating, not delivering data!

Traditional Architecture



- Single-workload
- Serialized
- Cached
- Optimized

Virtualized / Consolidated Architecture



- Multi-workload
- No cross-VM optimization
- Highly randomized

Summary: The Storage I/O Crisis

Performance Demands



- Multi-core CPUs
- Data growth
- “Instant” user expectations

Randomization: The “I/O Blender”



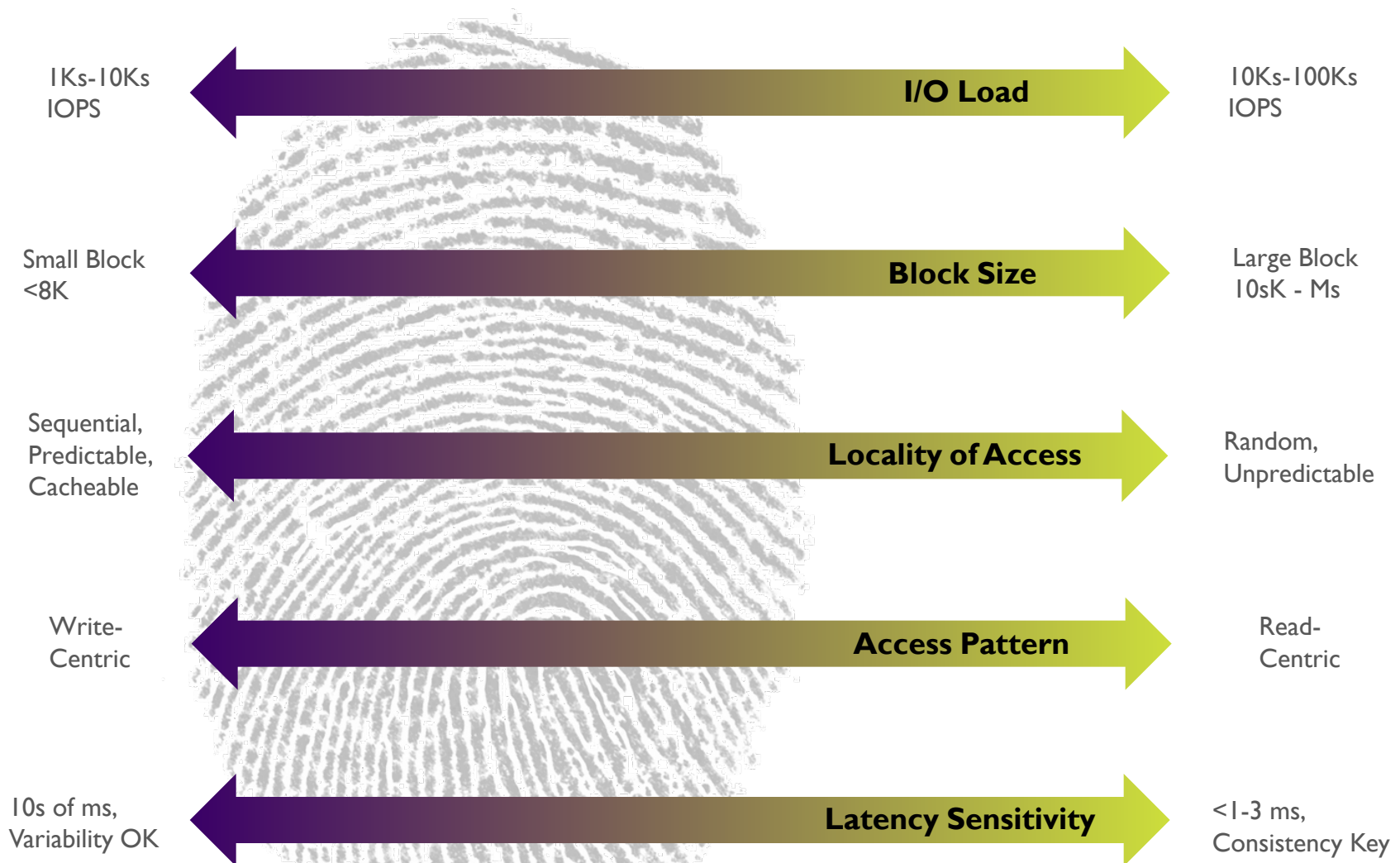
- Virtualization
- Data consolidation
- Cloud architectures

IOPS / TB



- 95% time seeking/rotating
- IOPS/TB dropping

Understanding Your Application's I/O Fingerprint



How to Evaluate Flash Alternatives



Cost: \$/GB?



\$/GB: Managed

- + Operations team

\$/GB: Operational

- + Datacenter space
- + Power / Cooling

\$/GB: Protected

- + Snapshot / replication software
- + Snapshot / replication copies
- + Backup software and media

\$/GB: Usable

- + Cost of RAID parity
- + Over-provisioning waste
- + Management software

\$/GB: Raw

- + Cost of raw disk / flash



Performance

- Read IOPS
- Write IOPS
- Latency
- Bandwidth
- \$/IOP



Power & Size

- Rackspace / floor space
- Power consumption
- Cooling



Protection

- How to make HA?
- How to backup?
- How to manage drive loss?



Integration

- Fits in current architecture?
- Requires process change?



Operational Simplicity

- Makes operations more or less complex?
- Increases TBs/admin managed?

Flash Architecture Approaches

Server-Attach PCI Flash



- PCI card in application server
- Host CPU typically used for flash management
- 100s of GB

Array with Flash Cache



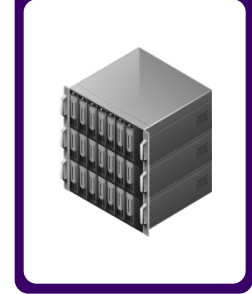
- Flash as a read and/or write cache in array, or to cache FS metadata
- All data persisted to spinning disk
- Typically <1% - 5% of total capacity

Array with Flash Tier



- Sub-LUN/FS tiering, where a LUN or FS is spread across flash and disk
- Hot blocks/files moved to flash, cold left on disk
- Typically 1% - 10% of total capacity

All-Flash LUN or array



- 100% flash LUN or array
- May or may not use DRAM for read/write caching

➤ Overview:

- ◆ Host-based PCI flash cards, typically 100s of GB in size
- ◆ Either non-RAIDed, or two cards mirrored in the server
- ◆ Either non-HA, or application clustered across multiple servers
- ◆ Leverages host CPU for flash management

➤ High-level Benefits:

- ◆ Highest-performance flash architecture possible
- ◆ Eliminates the cost/burden of shared storage
- ◆ Ultra-small footprint
- ◆ Potentially allows for the reduction of host DRAM

➤ High-level Challenges:

- ◆ Very high-cost, creates islands of expensive flash in every server
- ◆ Requires a re-architecture for most enterprises
- ◆ Expensive and difficult to protect (HA, backup)
- ◆ Requires mirroring for HA across multiple servers
- ◆ Requires app-level or external replication for DR

Server-Attach PCI Flash



◆ Overview:

- ◆ Expansion of array's DRAM cache with flash
- ◆ Implemented either via controller-connected PCI flash cards, or SSDs in drive bays
- ◆ Typically read-cache only, although some implementations of write cache as well
- ◆ Cache page size / caching scheme varies by array (typically 4-16K)

◆ High-level Benefits:

- ◆ Expands array's cache buffer from <<1% of total storage to 1-5%
- ◆ 10 latency improvement on IOs which hit the flash cache (10+ms → <1ms)
- ◆ Off-loading these IOs from disk reduces the load on disk and also makes the disk perform better
- ◆ Result: depends completely on cacheability of I/O stream, but typically 30-80% performance improvement

◆ High-level Challenges:

- ◆ Improvement depends completely on cacheability of the I/O stream, results vary
- ◆ Flash as a cache heavily exercised: requires SLC flash
- ◆ Very expensive: \$50-150/GB list prices typical from major vendors
- ◆ Cache requires time to “warm” to see the performance benefit, often doesn't persist across re-boots or HA events
- ◆ Minimal benefit for truly random I/O streams: random I/O can't be cached, it is random

Array with Flash Cache



◆ Overview:

- ◆ Creation of multiple tiers of storage (flash, enterprise FC/SAS HDD, SATA HDD) with the ability to spread a LUN / FS across them
- ◆ Typically implemented via 3.5” or 2.5” SSDs in the drive bays/shelves
- ◆ Generally 5-10% Flash, 90-95% disk in typical implementations
- ◆ Back-end storage is virtualized, and blocks/files are migrated up to flash or down to disk, based upon access patterns
- ◆ Chunk size for migration varies by vendor: typically MBs to GBs large
- ◆ Frequency varies by vendor: typically daily during low I/O windows

◆ High-level Benefits:

- ◆ Allows one to leverage flash for IOPs, HDDs for capacity
- ◆ Less cost than an all-flash solution, blends the economics of flash and HDD
- ◆ Suitable for MLC and SLC implementations of flash
- ◆ Potentially allows for significantly less disk / footprint / power if disk has been over-built for performance

◆ High-level Challenges:

- ◆ Success depends on the predictability/randomness of the I/O stream
- ◆ Only works well if the “hot blocks” are consistent
- ◆ Current implementation block sizes are large; a single hot block could pin a whole MB or GB chunk in flash
- ◆ Not a “real-time” adaptable solution, chunks only promoted/demoted once daily
- ◆ Difficult to setup and manage – is it worth the complexity of another tier to manage?

Array with Flash Tier



◆ Overview:

- ◆ Creation of a LUN on 100% flash
- ◆ Typically implemented via 3.5" or 2.5" SSDs in the drive bays/shelves
- ◆ Suitable for MLC or SLC flash implementations

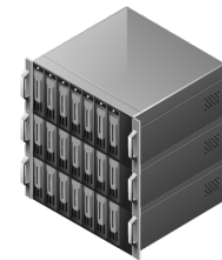
◆ High-level Benefits:

- ◆ 10x+ potential performance improvement vs. 1-2x for cached/tiered solutions
- ◆ Much more consistent latency: no "cache-miss" penalty, all IOs at the speed of flash (typically <1ms)
- ◆ Dramatically smaller size, allows for reduction of storage footprint by 4-5x
- ◆ Flash fast enough to eliminate the need for significant DRAM caching in the array
- ◆ RAID re-build times greatly improved due to the speed of flash (hours → 10s of minutes)

◆ High-level Challenges:

- ◆ Cost: flash varies from \$20/GB - \$150/GB depending on the vendor and type (MLC vs. SLC)
- ◆ Connectivity: most flash arrays have limited connectivity options compared to their more mature disk array alternatives
- ◆ HA & DR: varies by vendor, but the HA and DR models of these arrays are less mature than existing disk array alternatives

All-Flash LUN or Array



Understanding Flash ROI

Cost: \$/GB?

\$/GB: Managed

+ Operations team

\$/GB: Operational

+ Datacenter space
+ Power / Cooling

\$/GB: Protected

+ Snapshot / replication software
+ Snapshot / replication copies
+ Backup software and media

\$/GB: Usable

+ Cost of RAID parity
+ Over-provisioning waste
+ Management software

\$/GB: Raw

+ Cost of raw disk / flash



Flash Impacts...

Reduction in management cost (performance troubleshooting)

2-5x reduction in space
2-5x reduction in power

2-3x reduction in disk over-provisioning for performance

5-10x the raw cost...

- Datacenters are facing a storage I/O crisis



- Understanding your application's “I/O fingerprint” is key to choosing the best flash strategy for your environment
- Flash should be evaluated on several dimensions, it’s not all about performance



- Look at the full picture (beyond \$/GB raw!) to build your flash ROI

- Please send any questions or comments on this presentation to SNIA: tracksolidstate@snia.org