Implications of Non Volatile Memory on Software Architectures

Nisha Talagala

Lead Architect, Fusion-io

Overview

- Non Volatile Memory Technology
- NVM in the Datacenter
- Optimizing software for the ioMemory Tier
  - As storage
  - As memory
- Near and long term futures
- Fusion-io

Non Volatile Memory

- Flash
  - 100s GB of NVM per PCIe device
  - Access latency 25-50us read, 150-300us write
  - SLC for highest performance, MLC for highest capacity
  - Trend of MLC – increase in density, reduction of write cycles

- PCM and others
  - Still in research
  - Potential of extreme latency reduction
    - 100s ns read, 10s us or less writes

750 MB/s
145,000 IOPs
640 GB

ioDrive®

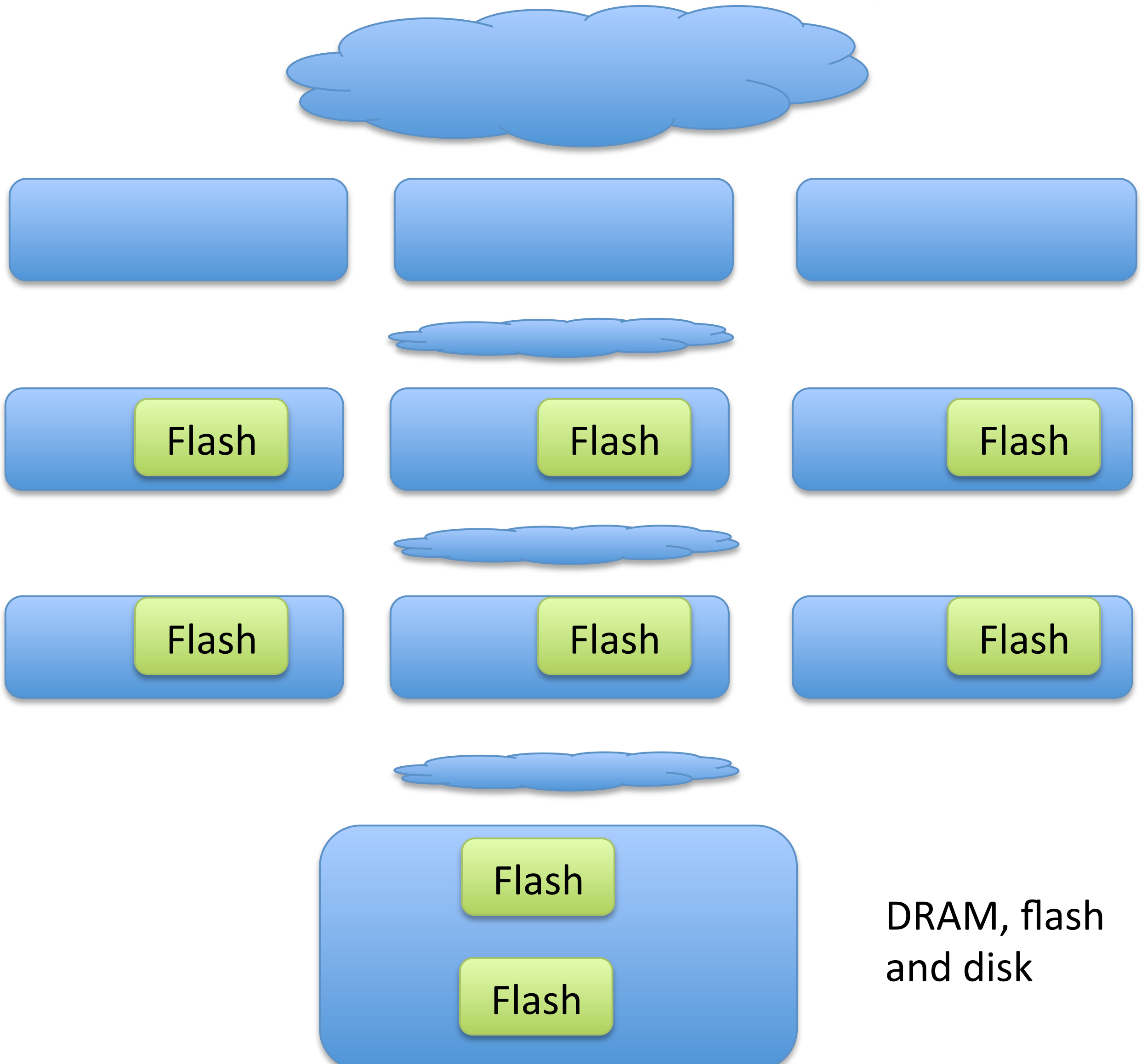
1.5 GB/s
278,000 IOPs
1.28 TB

ioDrive Duo®

6 GB/s
1,180,000 IOPs
5.12 TB

ioDrive® Octal

NVM in the Data Center Today



Web front Ends

Caching tiers — DRAM and Flash

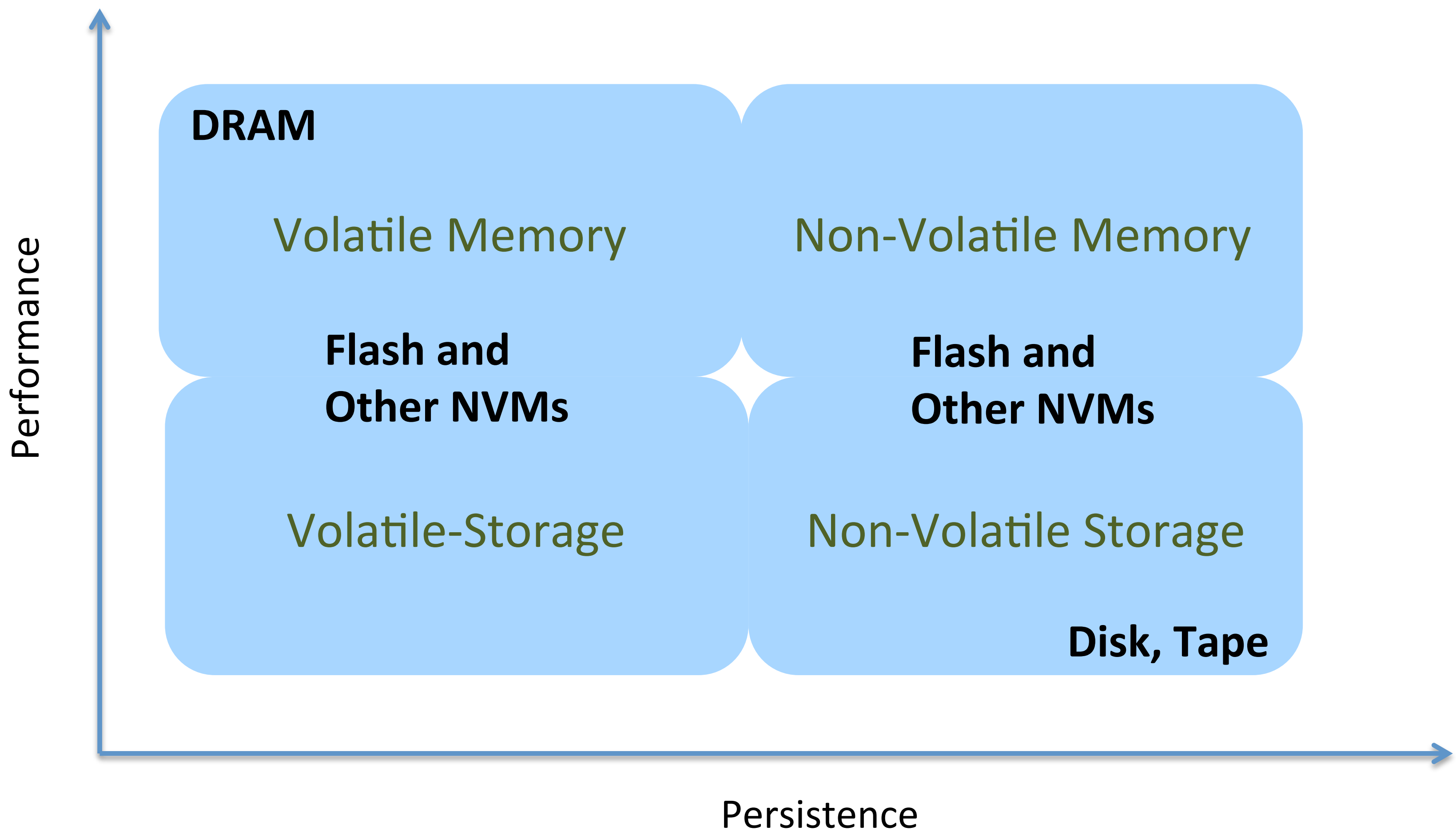
Database tiers — DRAM, flash and disk

Storage — DRAM, flash and disk

NVM in the Data Center (Contd)

- Performance
  - Closer to CPU is best – highest bandwidth, lowest latency
  - Server (compute) side flash complements storage side flash
- Hierarchy of DRAM, flash, disk
- Disk displacement usages
  - Caches – server and storage side
  - Scale out and cluster file systems
    - flash in metadata server
    - storage server
  - Staging, checkpoint
- DRAM displacement usages
  - Improved paging, semi-external memory

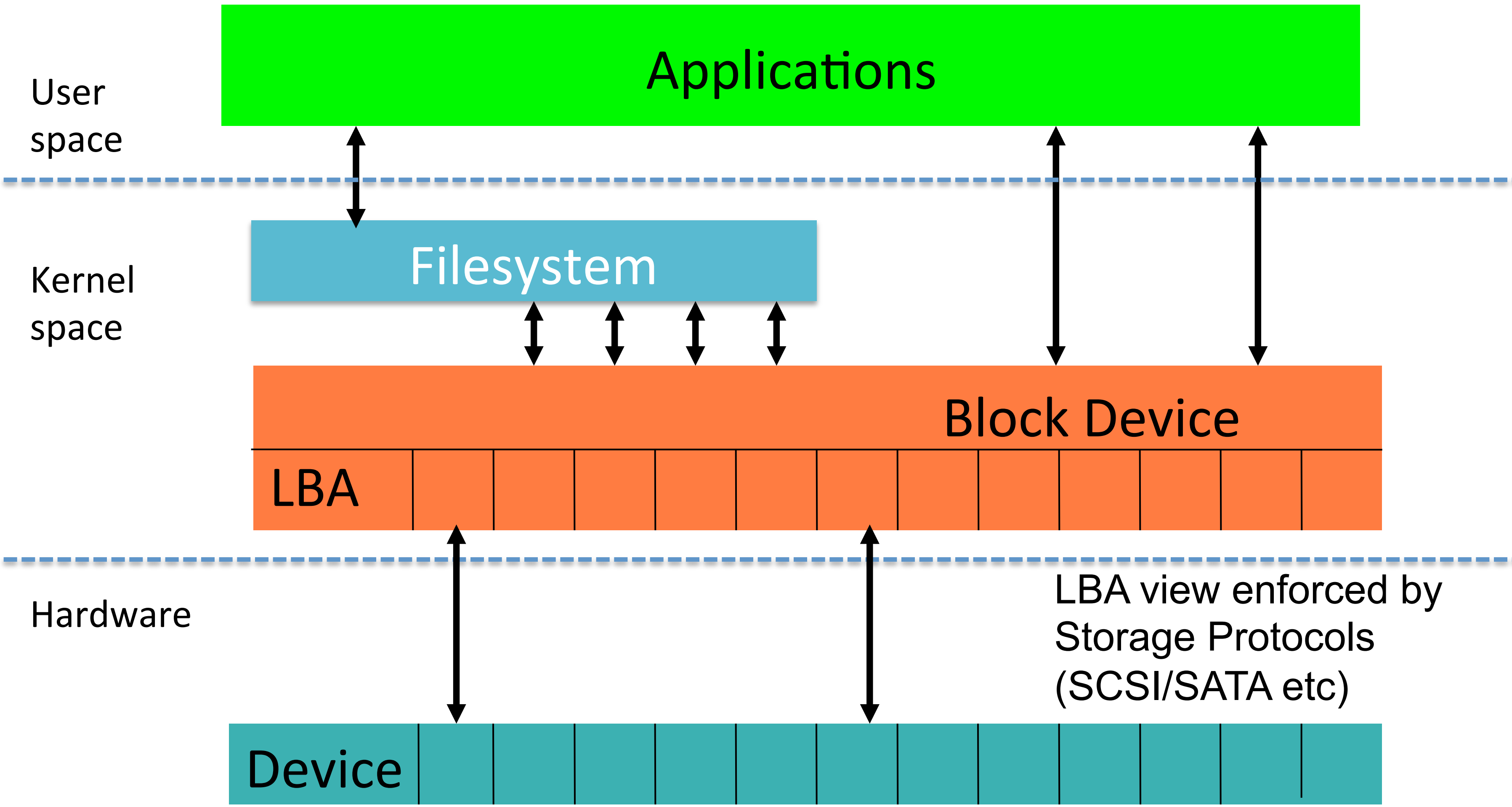
Flash – Storage or Memory?



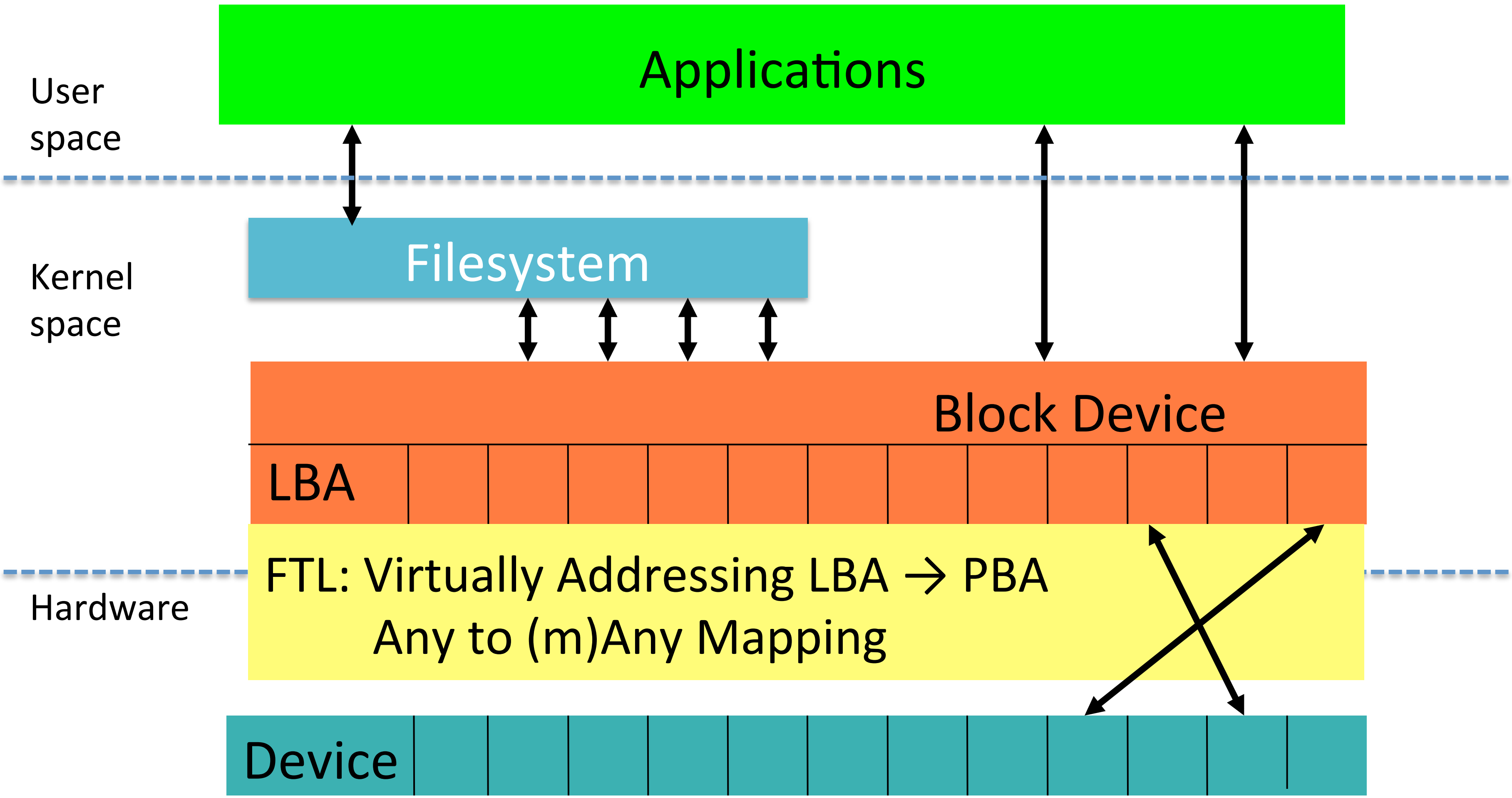
Overview

- Non Volatile Memory Technology
- NVM in the Datacenter
- Optimizing software for the ioMemory Tier
  - As storage
  - As memory
- Near and long term futures
- Summary

Traditional Storage Stacks



Flash in Traditional Storage Stacks



Traditional Storage Stacks and Flash

FUSiON-iO

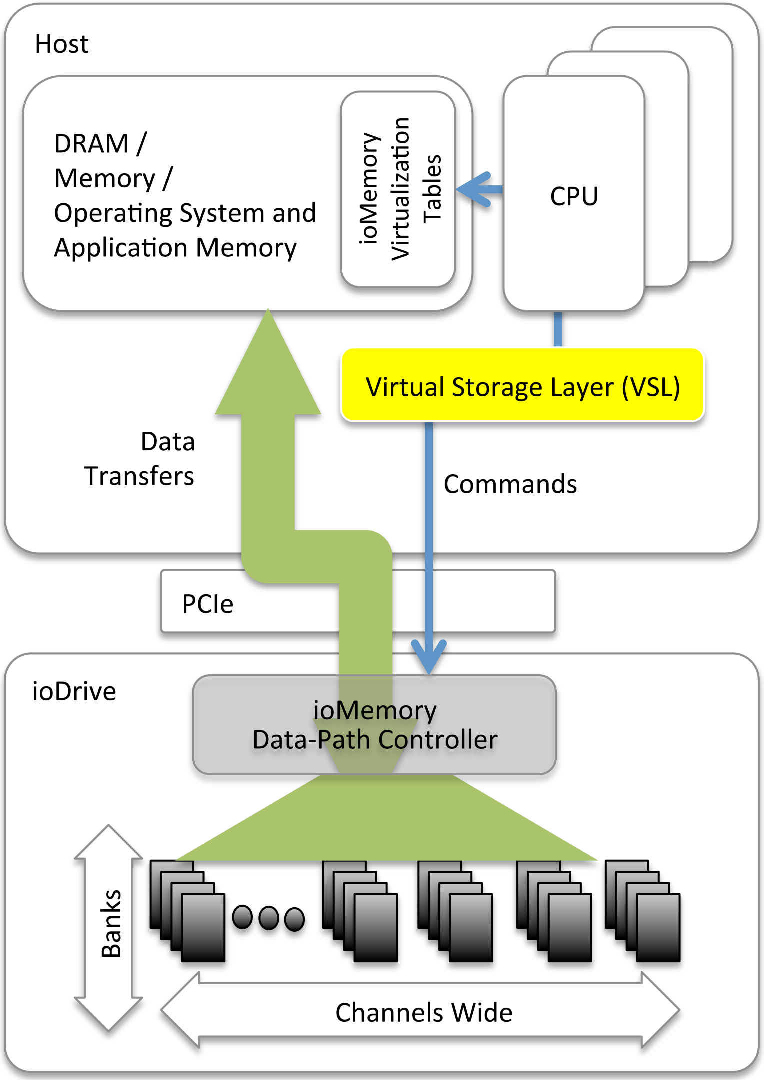
Area	Hard Disk Drives	Flash Devices
Logical to Physical Blocks	Nearly 1:1 Mapping	Remapped at every write
Read/Write Performance	Largely symmetrical	Heavily asymmetrical
Sequential vs Random Performance	100x difference. Elevator scheduling for disk arm	>10x difference. No disk arm – NAND die
Background operations	Rarely impact foreground	Regularly impact foreground – garbage collection
Wear out	Largely unlimited writes	Limited writes
IOPS	100s to 1000s	100Ks to Millions
Latency	10s ms	10s-100s us

Block storage stacks are sub optimal for Flash

Virtual Storage Layer

FUSION-IO

- Cut-thru architecture – avoids traditional storage protocols
  - Scales with multi-core
- Traditional block access methods for compatibility
- New access methods and primitives natively supported by FTL



Flash Optimized Storage Primitives and Usages

- Atomic Writes
- Dynamic allocation – primitives and thin provisioning

Atomic Writes

- Storage stacks today
  - No atomicity guarantees
  - Upon failure – data can be old, new, mixed or other
- Flash Translation Layers enable atomic writes
  - Transactional semantics for one or group of writes
  - Data writes occur in their entirety or not at all
  - Moves responsibility for atomics from applications to storage devices
- Reduces the significant work at applications and file systems to guarantee data integrity during failure

EXAMPLE PERFORMANCE RESULTS (MYSQL INSERT WORKLOAD)

FUSION-IO

Regular MySQL	Trans./sec	Data Written	Avg. Latency	95% Latency
ACID-compliant test	11.7K	24.3GB	2.73ms	18.24ms

Atomic-write	Trans./sec	Data Written	Avg. Latency	95% Latency
ACID compliant – atomic write prevents torn sectors	15.8K	12.15GB	2.02ms	7.21ms

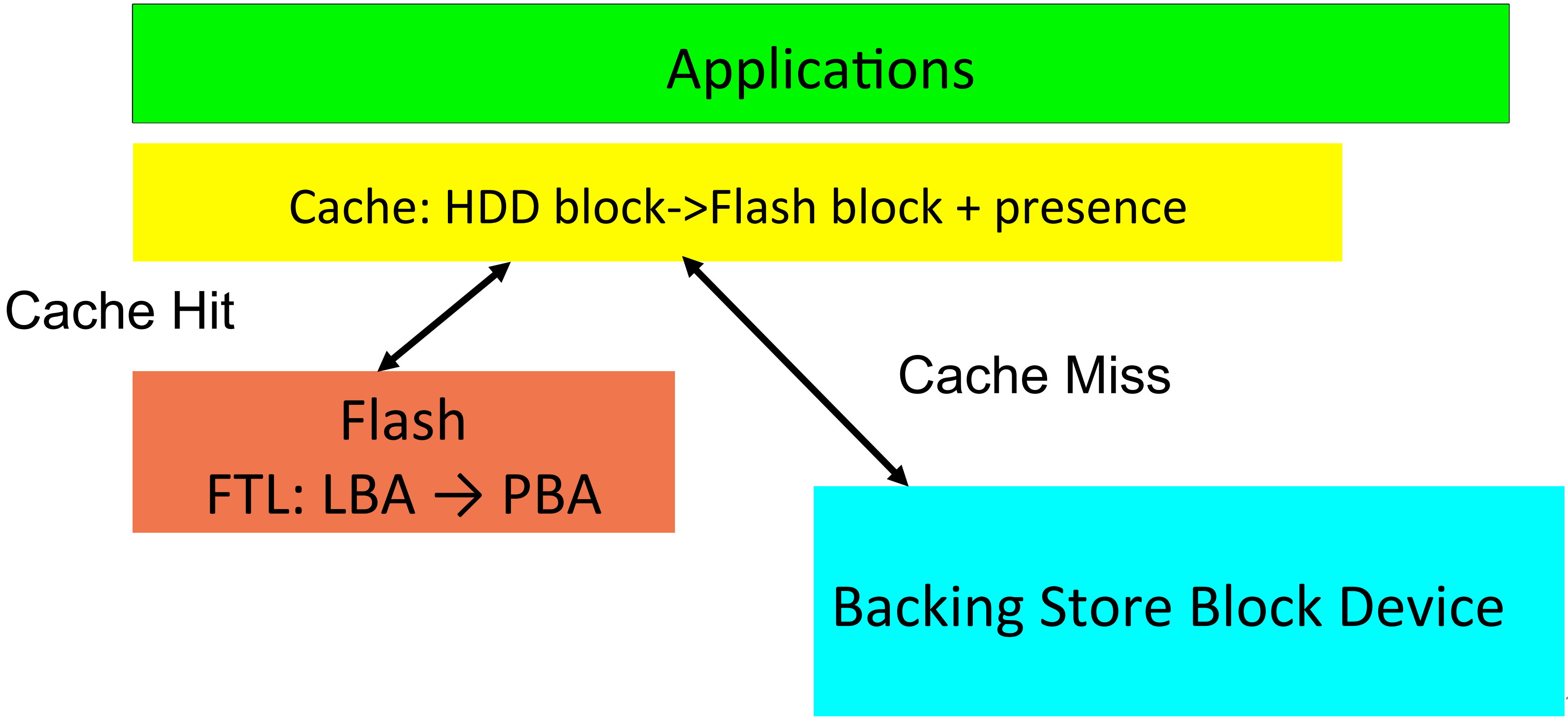
- **Config: 1,000,000 inserts in 32, 2-million-entry tables, using 32 threads**
- **Bulk of improvement by eliminating logging safeties (double write) required with non-atomics**
- **Summary - ~35% TPS improvement, ~2.5x 95% latency redux, 2x drive endurance improvement**

Flash Optimized Storage Primitives and Usages

- Atomics
- Dynamic allocation – primitives and thin provisioning

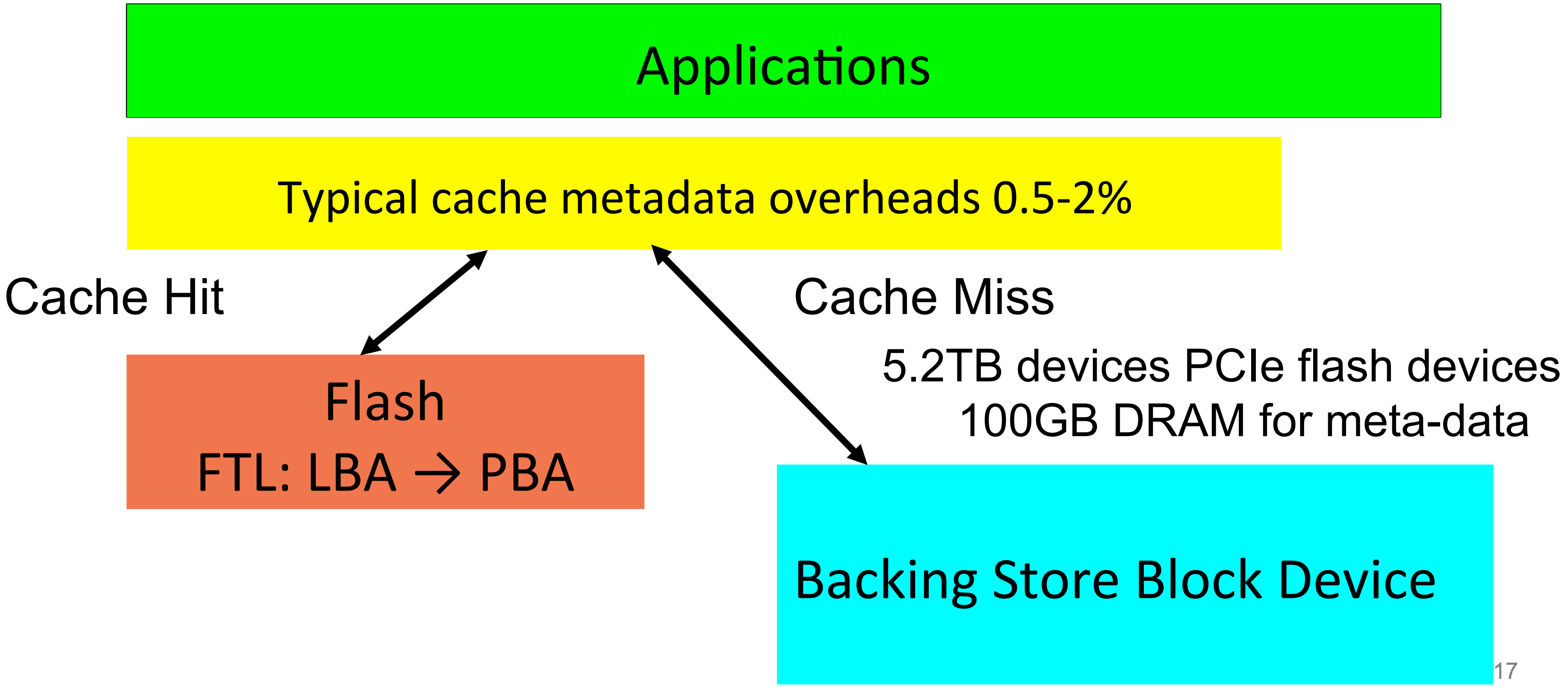
Example - Block based caches

Block cache – intercept block traffic – direct high traffic blocks to flash based cache



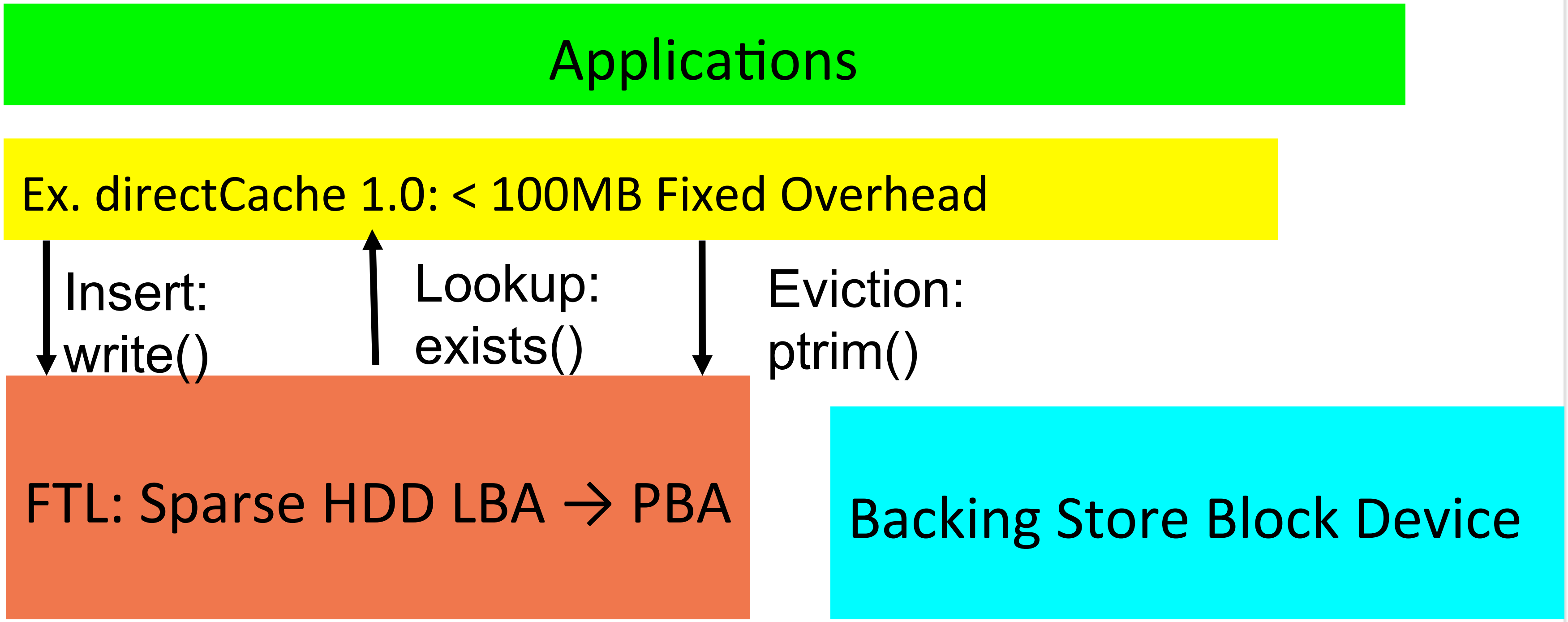
Mapping tables and overheads

Mapping required to translate disk locations to flash locations – overhead per flash block, per disk block



Using flash optimized dynamic allocation

Leverage FTL mapping and dynamic allocation
Enables “zero metadata” caches – fixed metadata cost



Summary – Optimizing Storage Usages

Flash based storage is virtually addressed, embrace it!

Power of FTL enables new primitives that simplify applications, increase performance and improve efficiency

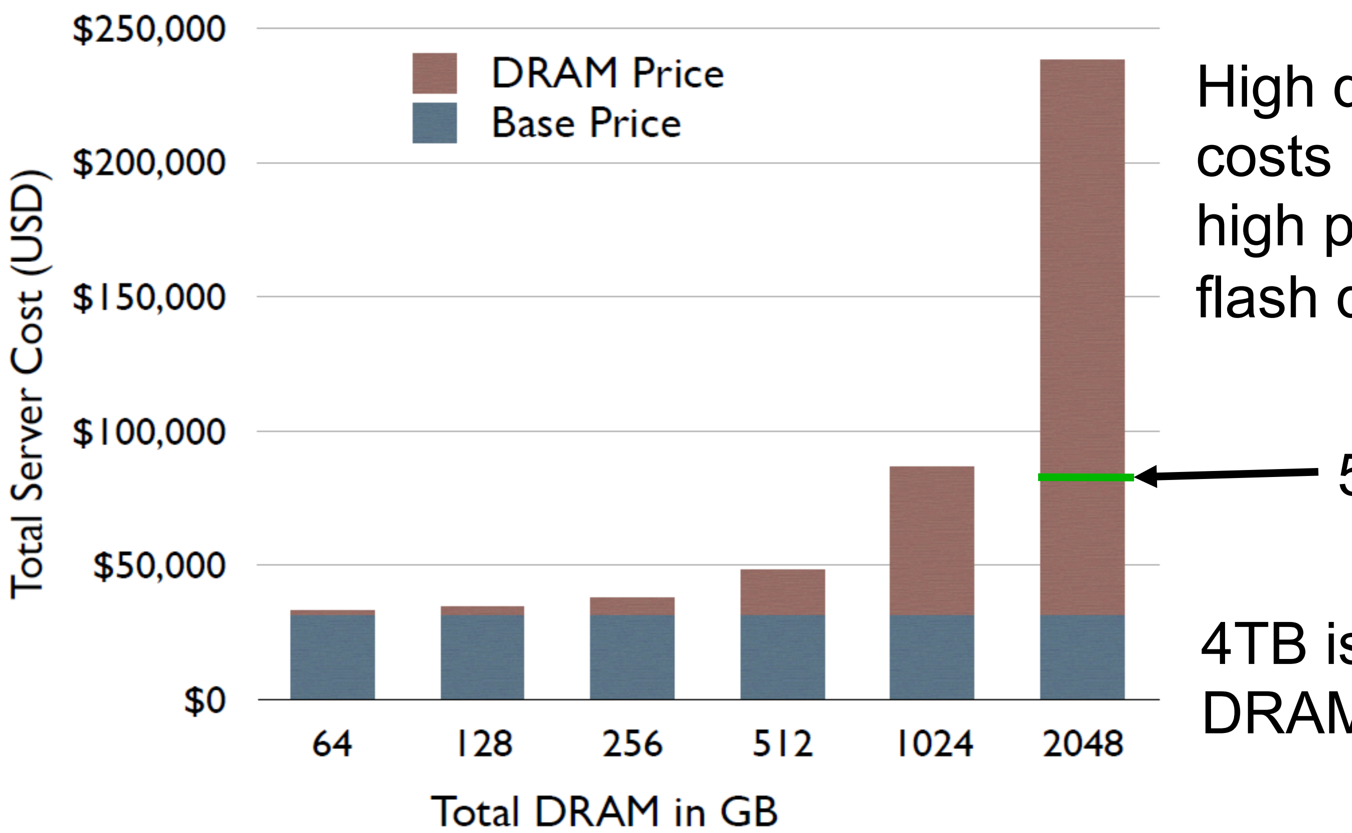
Backwards compatible with block interface – enables application evolution

Overview

FUSiON-iO

- Non Volatile Memory Technology
- NVM in the Datacenter
- Optimizing software for the ioMemory Tier
  - As storage
  - As memory
- Near and long term futures
- Summary

Why Use NAND as Memory?



High density DRAM costs 10x higher than high performance flash devices

4TB is maximum capacity DRAM in single systems

Why Use NAND as Memory?

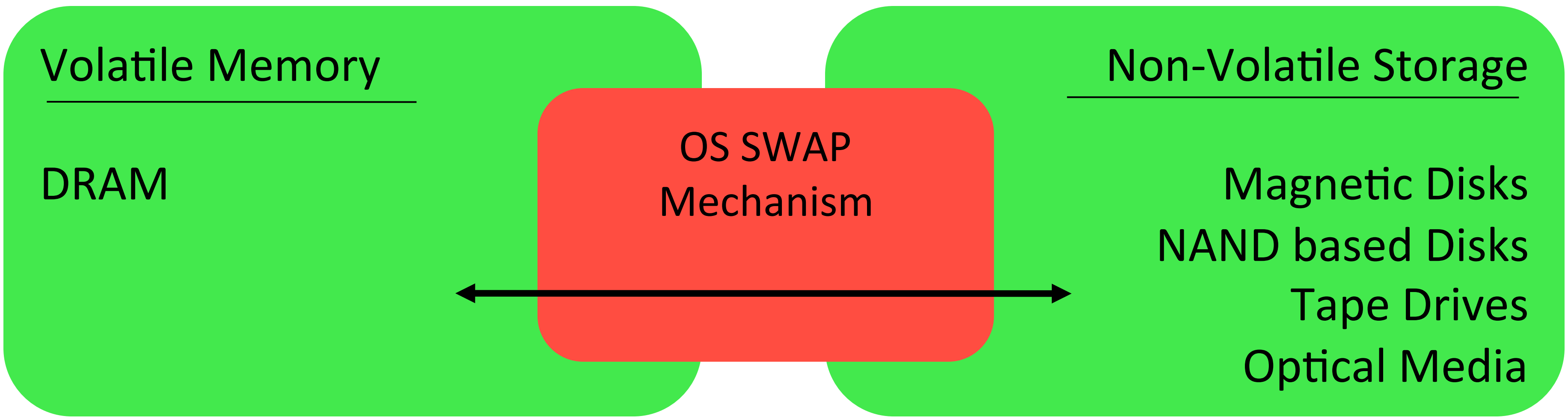
High Density PCIe NAND-flash

5 rack units, 45TB capacity, 1.2kW power consumption

High Density DRAM

DRAM (GB)	128	256	512	1024	2048	4096
Space (RU)	6	6	10	40	80	80
Power (kW)	1.1	1.4	2.7	6.5	7.3	14.4

NAND as Virtual Memory



Traditional SWAP was never designed for performance
“Last resort” - before OOM

<= 30MB/s throughput
10-100ms software overhead

Transparent Expansion of Application Memory*

FUSION-iO

- Application Transparency: No source code modification!
  - Layers under malloc()
- Unhindered Access to DRAM
  - User level paging
  - Low overhead tiering: Must not inhibit flash performance
- Intelligent paging decisions including application hints

* Fusion-io and Princeton University Collaboration

DRAM, SWAP, and TEAM

Xeon 3.43Ghz with DDR3 1333Mhz running Linux

- 10,900,000 Random 64Byte Memory IOPS
- 120,000 Random 512B NAND-flash IOPS

- Linux SWAP: 11k Random NAND-memory IOPS
- TEAM Tiering: 93k Random NAND-memory IOPS

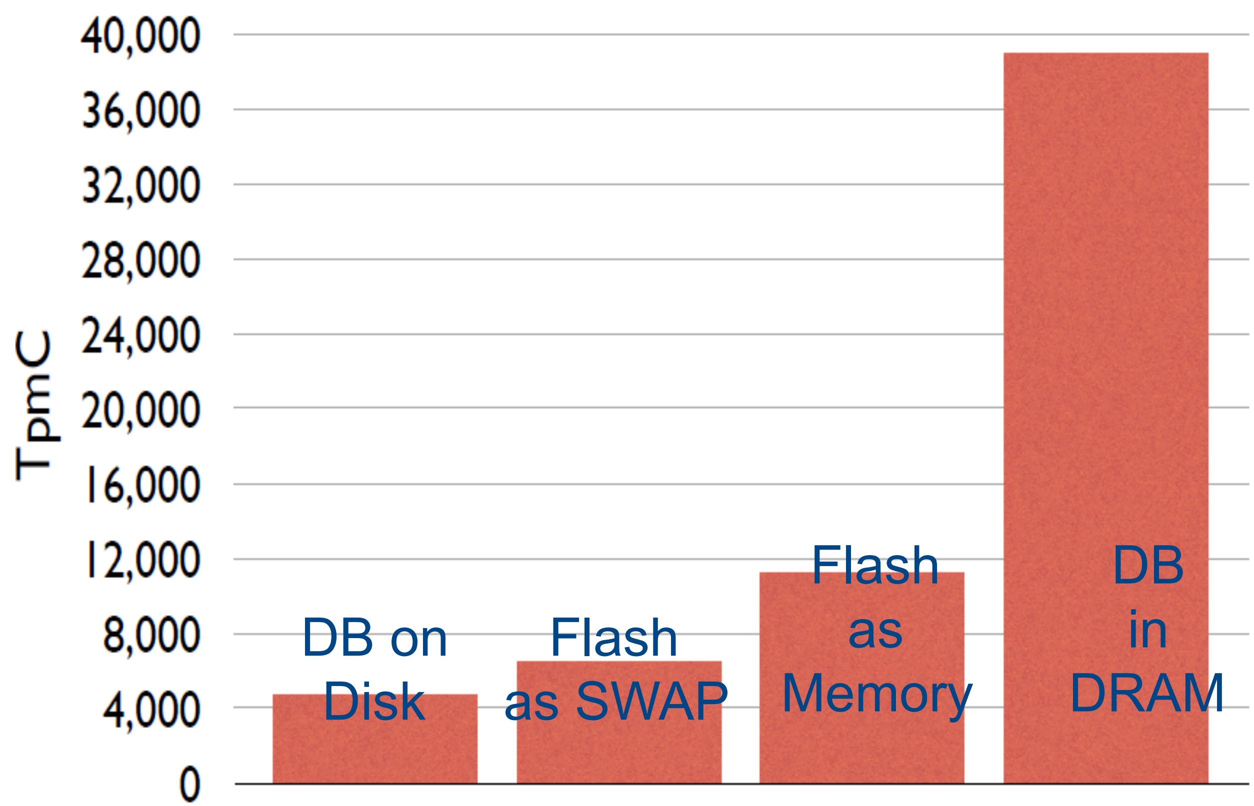
Observation – NAND performance is much lower than DRAM, but TEAM can access near full NAND perf

Example: Databases that fit in DRAM

FUSiON-iO

Percona MySQL 5.5
TPC-C

Flash as virtual Memory achieves 33% performance of an all DRAM solution



24 core Xeon, 40G DRAM, 140G Fusion-io NAND-flash: 40G DB size

Summary – Optimizing for Flash as Memory

FUSiON-iO

NAND-flash is ***not ready*** as a wholesale DRAM replacment.

Dense, power efficient, cheap. Too slow.

However NAND-flash as a memory displacement can improve performance/$/watt

Example:

33% the performance of all DRAM, 8% the TCO, and 5% power consumption.

NAND-flash is a ***cost effective*** way to build large memory systems, but software work is required to reap the benefits

Overview

- Non Volatile Memory Technology
- NVM in the Datacenter
- Optimizing software for the ioMemory Tier
  - As storage
  - As memory
- Near and long term NVM futures
- Summary

The future of Non Volatile Memory

- New technologies
  - Phase Change Memory, Memristors, etc
  - Promise 10x-1000x performance improvement
  - Still asymmetric
  - Wear is 10x-1000x better
- New opportunities and challenges
  - NVM on the memory bus may become possible
  - Fundamental changes to CPUs, OSes, even compilers and programming languages, are possible

FUSION-IO

THE WALL STREET JOURNAL.

20PB

2006	2007	2008	2009	2010	2011
Company founded Set out consolidate memory and storage	Virtual ioMemory technology unveiled	First native PCIe NAND flash device	HP OEMs products IBM OEMs products	Dell OEMs products Tens of thousands of units shipped WSJ names Fusion-io #1 young InfoTech company	20 Petabytes shipped More than 1000 end customers

We're hiring!

- Operating systems
- Management software
- Virtualization
- Hardware
- Device physics
- Java, C, C++, Python
- Software engineers, HW engineers, Research Scientists, Architects

References

- DFS – A File System for Virtualized Flash Storage, FAST 2010
- Beyond Block I/O – Rethinking Traditional Storage Primitives, HPCA 2011
- SSDAlloc – Hybrid SSD/RAM Management Made Easy, NSDI 2011
- Multithreaded Asynchronous Graph Traversal for In-memory and Semi-Extended-Memory, SC 2010
- PTRIM + EXISTS – Exposing New FTL primitives to Applications, UCSD NVM Workshop 2011
- TEAM – Transparent Extension of Application Memory – Under submission

THANK YOU

ntalagala@fusionio.com