

# This is your Database on Flash: Insights from Oracle Development



ORACLE

# Disclaimer

THE FOLLOWING IS INTENDED TO OUTLINE OUR GENERAL PRODUCT DIRECTION. IT IS INTENDED FOR INFORMATION PURPOSES ONLY, AND MAY NOT BE INCORPORATED INTO ANY CONTRACT. IT IS NOT A COMMITMENT TO DELIVER ANY MATERIAL, CODE, OR FUNCTIONALITY, AND SHOULD NOT BE RELIED UPON IN MAKING PURCHASING DECISION. THE DEVELOPMENT, RELEASE, AND TIMING OF ANY FEATURES OR FUNCTIONALITY DESCRIBED FOR ORACLE'S PRODUCTS REMAINS AT THE SOLE DISCRETION OF ORACLE.

Garret Swart  
Roopa Agrawal  
Sumeet Lahorani  
Kiran Goyal

# The Promise of Flash

- Replace expensive 15K RPM disks with fewer Solid State devices
  - Reduce failures & replacement costs
  - Reduce cost of Storage subsystem
  - Reduce energy costs
- Lower transaction & replication latencies by eliminating seeks and rotational delays
- Replace power hungry, hard-to-scale DRAM with denser, cheaper devices
  - Reduce cost of memory subsystem
  - Reduce energy costs
  - Reduce boot/warm-up time



# SSD's have been around for years: What's different?

- The old SSD market was latency driven
  - The drives were quite expensive
- Consumer applications have driven Flash prices down
- The new SSD market is latency and \$/IOPS driven

# What will we learn today?

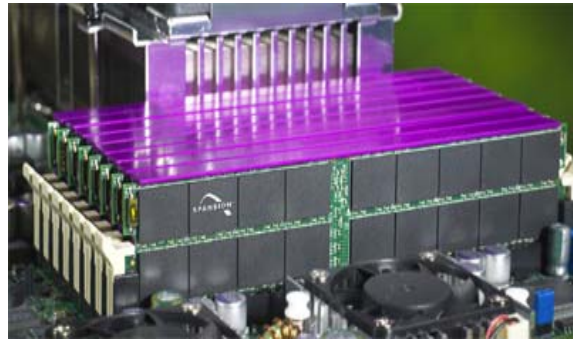
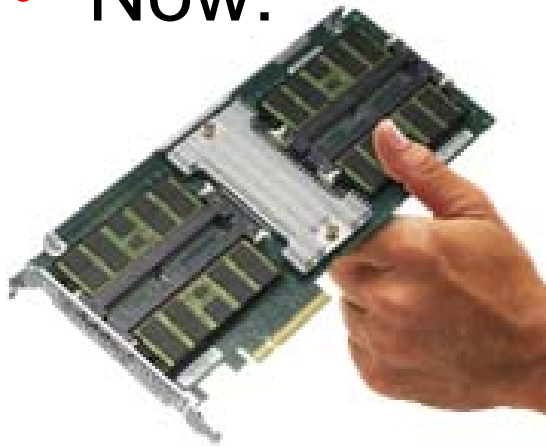
- Lots of innovative Flash products
  - Why they are tricky to evaluate
- Which Oracle workloads can benefit from Flash today
- How to use IO Intensity to assign objects to storage tiers
- Oracle Performance results

# A Cambrian Explosion of Diversity: In Flash Technology

- 530,000,000 years ago:



- Now:



# Comparing SSDs and Enterprise Disks

- Great random read IOPS ( $> 100\times$ )
  - 20 to 80K 8K reads/second (vs. 300)
- Very good random write IOPS ( $> 50\times$ )
  - 5 to 30K 8K writes/second (vs. 300)
- Very good latency ( $< 1/50$ )
  - 100 to 200 us (vs. 7000)
- Good sequential bandwidth ( $> 3\times$ )
  - 100 to 500 MB/s reads
- More expensive per GB (10 to 60x)
  - \$20 to \$60/GB: Depending on features

# Outline

- The Flash Promises
- **Flash Technology**
- Flash Products
- DB Workloads that are good for Flash
- Flash Results

# Flash Technology

- Two types: NAND and NOR
- Both driven by consumer technology
  - NAND: Optimized for cameras and MP3 players
    - Density, sequential access
    - Page read, Page program, Block erase
  - NOR: Optimized for device boot device
    - Single word access & XIP
    - Word read, Word program, Block erase
- Both being adopted for Enterprise use

# The Complexity of Flash

- Data loss
  - Caused by defects, erase cycles (endurance), and interference
  - SLC: Single bit per cell
    - Less density, greater endurance
  - MLC: Multiple bits per cell (2 → 3 → 4)
    - Greater density, less endurance
- Write-in-place is prohibitive
  - Read, Erase, Rewrite: Must be avoided
- Low per Chip bandwidth

Compensate for endurance, data loss, low per chip bandwidth, and write issues with **complex firmware**

# How do Flash Devices Differ?

- Sustained Write Performance
  - Flash devices tend to be Log Structured with Flash Translation Layer (FTL) to record the physical location of pages
  - Garbage Collector: Removes overwritten data, creates free space
    - Write Amplification: Physical Writes/Logical Write
    - Depends on sequentiality and drive free space
    - Can only measure after the disk is in steady state
- Efficient Page Sizes
  - Use RAID-5 to recover lost data and boost bandwidth
- Power up time
  - FTL must be rebuilt on power up
- IO Latency
  - Average and Standard Deviation

# Outline

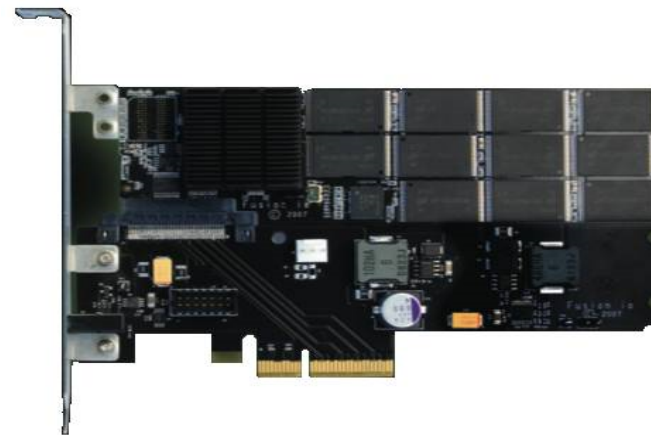
- The Flash Promises
- Flash Technology
- **Flash Products**
- DB Workloads that are good for Flash
- Flash Results

# How to use a Flash Device?

- Direct Attached vs. Networked Storage
  - Mount flash inside your server (PCIe or SATA)
  - Share device on the network (FC or 10GE)
- Cache vs. Tier 0
  - Frequently used data automatically stored in Cache
  - Carefully place data on Tier 0 storage
- Networked Cache vs. Hybrid Storage
  - Cache as shared resource
  - Cache integrated in storage unit

# Direct Attached Devices

- Memory Bus (AMD HT)
  - Virident/Spansion EcoRAM: “Reads like RAM, writes like Disk”
- I/O Bus (PCIe)
  - FusionIO
  - NVMHCI TurboMemory
- SSD (SATA/SAS)
  - Intel, Samsung, STEC, BITMICRO, ...



# Networked Devices

- Tier 0 Storage
  - TMS RamSan 500
  - EMC Symmetrix DMX-4
- Networked Cache
  - Gear6 CACHEfx
  - IBM QuickSilver SVC??
- Hybrid Storage
  - Sun Fishworks ZFS
  - NetApp Flash PAM



# Outline

- The Flash Promises
- Flash Technology
- Flash Products
- **DB Workloads that are good for Flash**
- Flash Results

# Flash Friendly DB Workloads

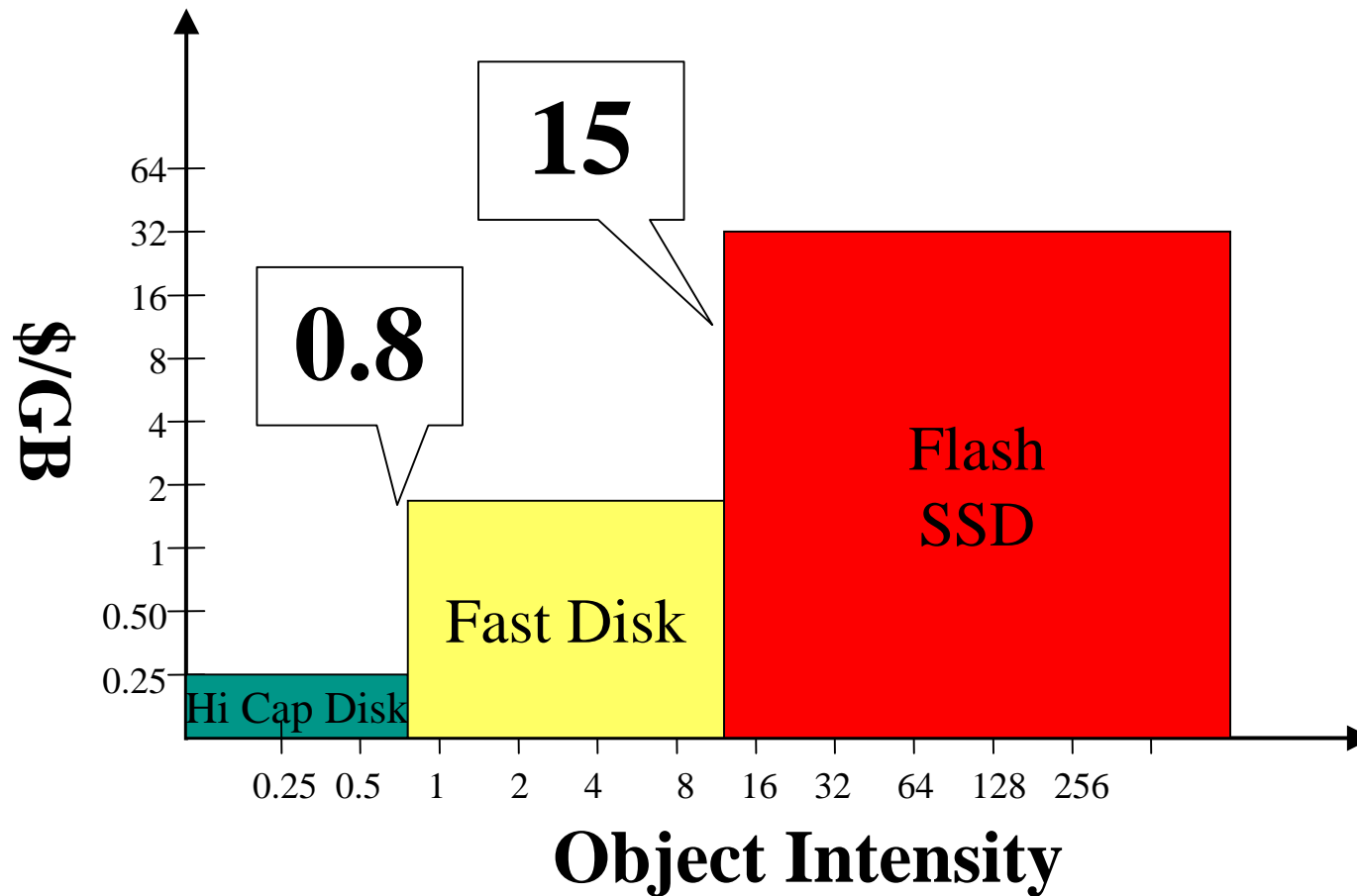
- Random Reads: ~25x better IOPS/\$
  - B-Tree leaf access
  - ROWID look up into Table
  - Access to out of line LOB
  - Access to overflowed row
  - Index scan over Unclustered Table
  - Compression: Increases IO intensity (IOPS/GB)
- Serial Reads: 50x faster
  - Buying more disks doesn't reduce latency!
- Random Writes: ~8x better IOPS/\$
  - Row update by PK
  - Index maintenance
  - Reduce checkpoint interval
- TEMP: Sort areas & Intermediate tables
  - Sequentially read and written BUT IO done in 1MB units: not enough to amortize seeks
  - Lower Latency: Get In, Get Out

# Not as Cost Effective on Flash

- Redo Log Files
  - Sequentially read and written AND commit latency already handled by NVRAM in controller
- Undo Table space
  - Sequentially written, randomly read by FlashBack. But reads are for recently written data which is likely to still be in the buffer cache
- Large Table Scans
- Buffer pools with lots of writes
  - Low latency reads followed by updates can fill up pool with dirty pages, which takes a long time to drain because Flash devices write 2-4x much slower than they read
  - Can cause “Free Buffer Waits” for readers

# Matching Object to Storage Tier

- Object Intensity = Object-IOPS / Object-GB



# Introducing Intensity Skew

- Store objects (tables, indexes, partitions) with different intensities in different Tablespaces
- Partition tables to separate high intensity data from low intensity data
  - If recent data is hot, then partition by date
  - If “Gold” customers are hot, partition by status
- Calculate intensity based on AWR reports
  - Use separate ASM disk groups per tier
  - Store all files for a Tablespace in same tier

# IO Optimization Example

- Object A has 300 GB and 10K IOPS
  - Intensity ~ 33 : Use 1 Flash SSD
- Object B has 600 GB and 4K IOPS
  - Intensity ~ 6.6: Use 14 Fast Disks

Flash Drive	Fast Disks	Price	IOPS	Capacity
s 0	47	\$28200 +	14100	14.1 TB
1	14	\$18000	34200	4.5 TB
3	0	\$28800	90000	960 GB

# Matching IO Intensity to Storage Tier

- Object Intensity = Object-IOPS / Object-GB
- Match Object Intensity with Storage Tier
  - $COST[tier] = \text{MAX}(\text{Object IOPS} * \$/IOPS[tier], \text{Object-GB} * \$/GB[tier])$
- Tier Intensity[tier] =  $\$/GB[tier] / \$/IOPS[tier]$ 
  - If Object-Intensity > Tier-Intensity then cost of object in this tier is IOPS bound
  - Otherwise cost of object in this tier is Capacity bound
- Optimize:
  - If cost is IOPS bound, compare with lower (cheaper IOPS) tier
  - if cost is Capacity bound, compare with upper (cheaper capacity) tier

# Tier Intensity Cut Offs

- Tier Intensity =  $\$/\text{GB} / \$/\text{IOPS}$ 
  - Hi Cap Disk Intensity =  $(\$250 / 1000\text{GB}) / (\$250 / 100 \text{ IOPS}) = 1/10$
  - Fast Disk Intensity =  $(\$600 / 300\text{GB}) / (\$600 / 300 \text{ IOPS}) = 1$
  - Flash Intensity =  $(\$2400 / 80\text{GB}) / (\$2400 / 30\text{K IOPS}) = 375$
- If Object Intensity is  $> 375$ : Choose Flash
- If Object Intensity is between 1 and 375
  - Break even when Intensity is  $\$/\text{GB}[\text{Flash}] / \$/\text{IOPS}[\text{Fast-Disk}] = (\$2400 / 80\text{GB}) / (\$600 / 300 \text{ IOPS}) = \mathbf{15}$
- If Object Intensity is between  $1/10$  and 1:
  - Break even when Intensity is  $\$/\text{GB}[\text{Fast-Disk}] / \$/\text{IOPS}[\text{HC-Disk}] = (\$600 / 300 \text{ GB}) / (\$250 / 100 \text{ IOPS}) = \mathbf{0.8}$
- If Object Intensity is  $< 1/10$ : Choose High Capacity Disk

# Outline

- The Flash Promises
- Flash Technology
- Flash Products
- DB Workloads that are good for Flash
- **Flash Results**
  - Storage Microbenchmarks
  - Oracle Microbenchmarks
  - OLTP Performance

# Storage Micro-Benchmarks

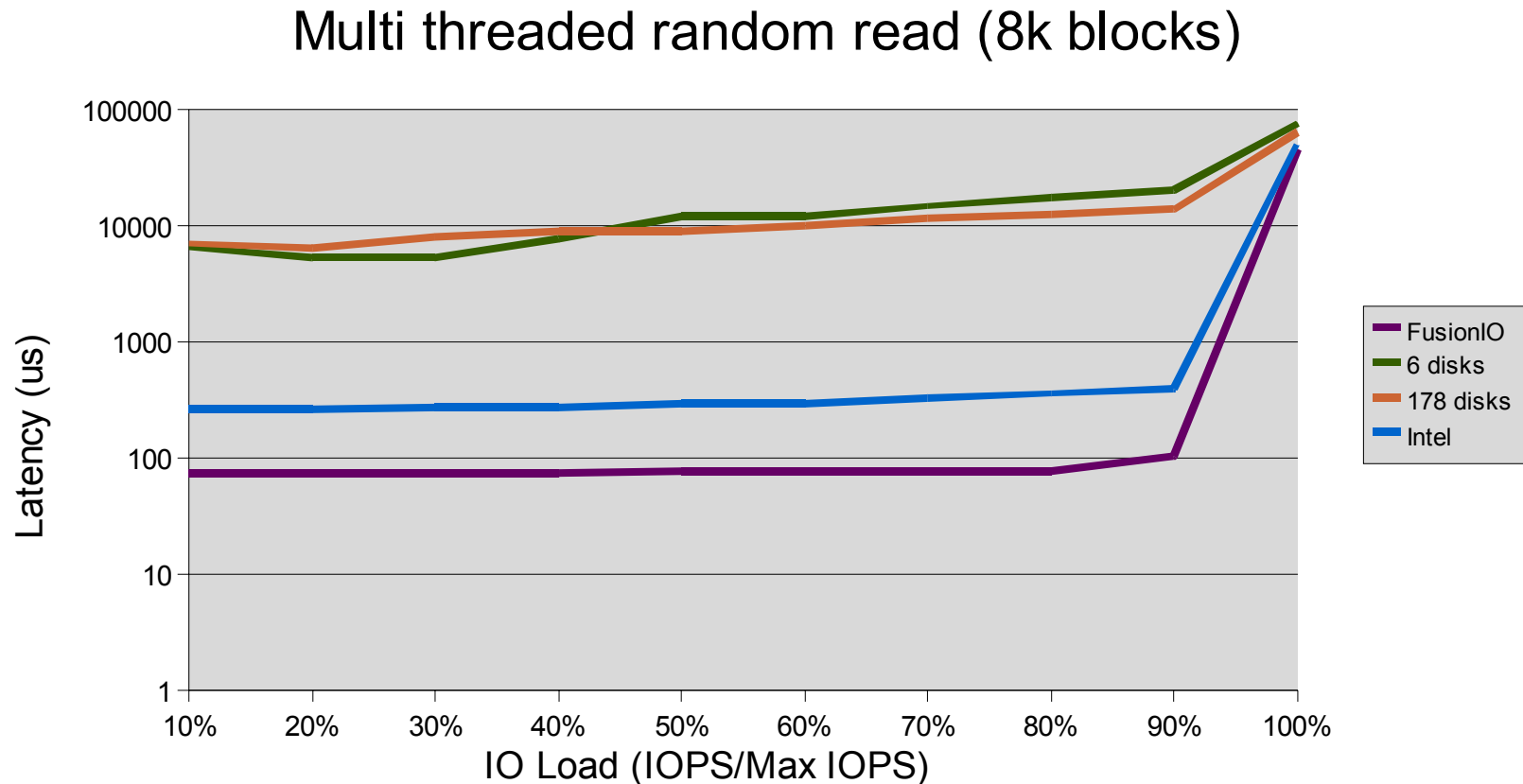
- Measurements were done using the fio tool (<http://freshmeat.net/projects/fio/>)
- We compare:
  - 6 disks in 1 RAID-5 LUN, 15k rpm, NVRAM write buffer in SAS controller
  - 178 disks in 28 (10x7+18x6) RAID-5 LUNs, 15k rpm, NVRAM write buffer in SAS controller
  - 1 Intel X25-E Extreme SATA 32GB SSD (pre-production), NVRAM write buffer in SAS controller disabled, External enclosure
  - 1 FusionIO 160 GB PCIe card formatted as 80 GB (to reduce write amplification)

# Storage IOPS

	Read IOPS 8k pages	Write IOPS 8k pages
6 SAS Disks	1637	573
178 SAS Disks	35638	16848
1 Intel SATA SSD	17030	3248
1 PCIe FusionIO	47302	25261

# Random Read Latency vs. Load

- Flash device latencies are nearly constant until 90% of maximum load

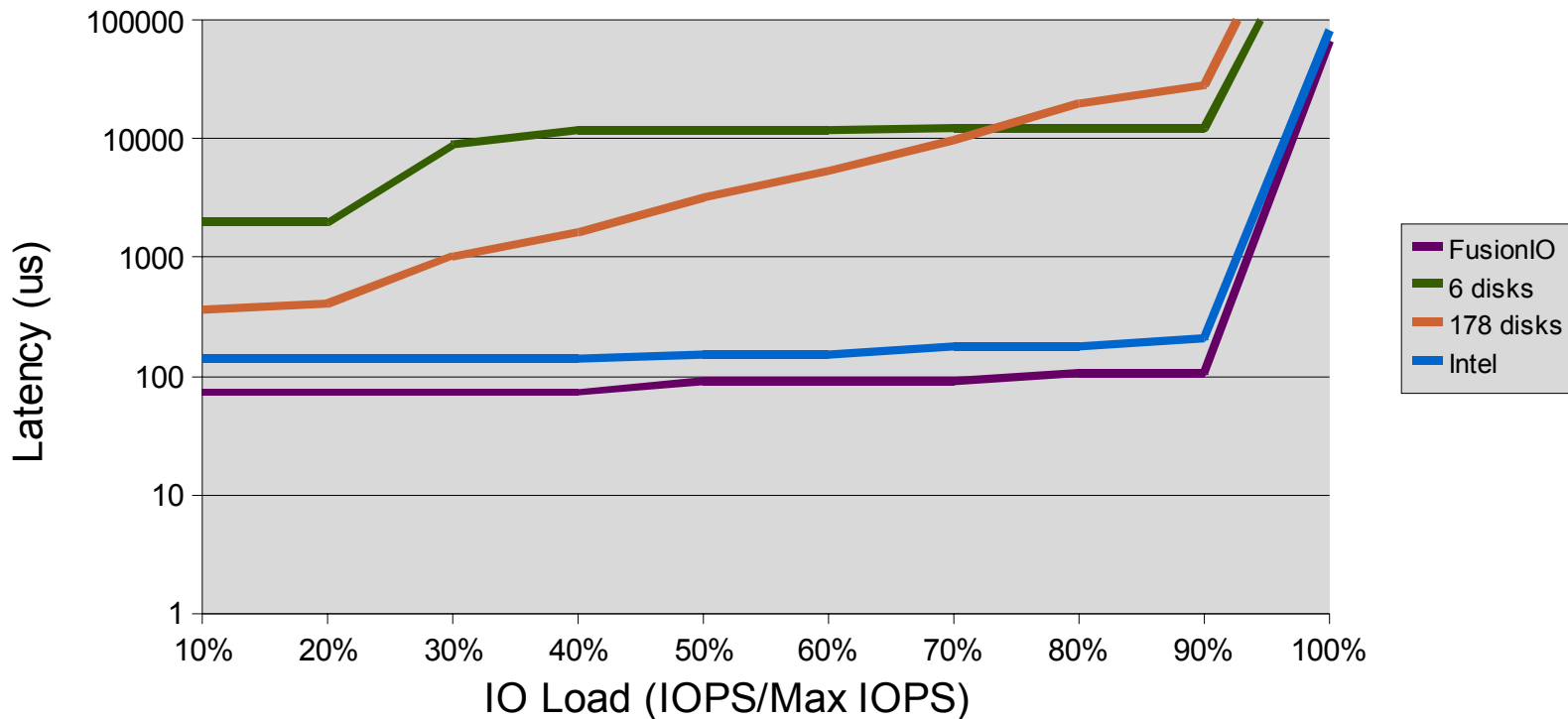


† FusionIO 80GB PCIe card, Intel 32GB SATA SSD in external storage

# Random Write Latency vs. Load

- Note the steps in the disk graphs when NVRAM write buffer fills
- Flash latencies are nearly constant until 90% of maximum load

Multi threaded random write (8k blocks)

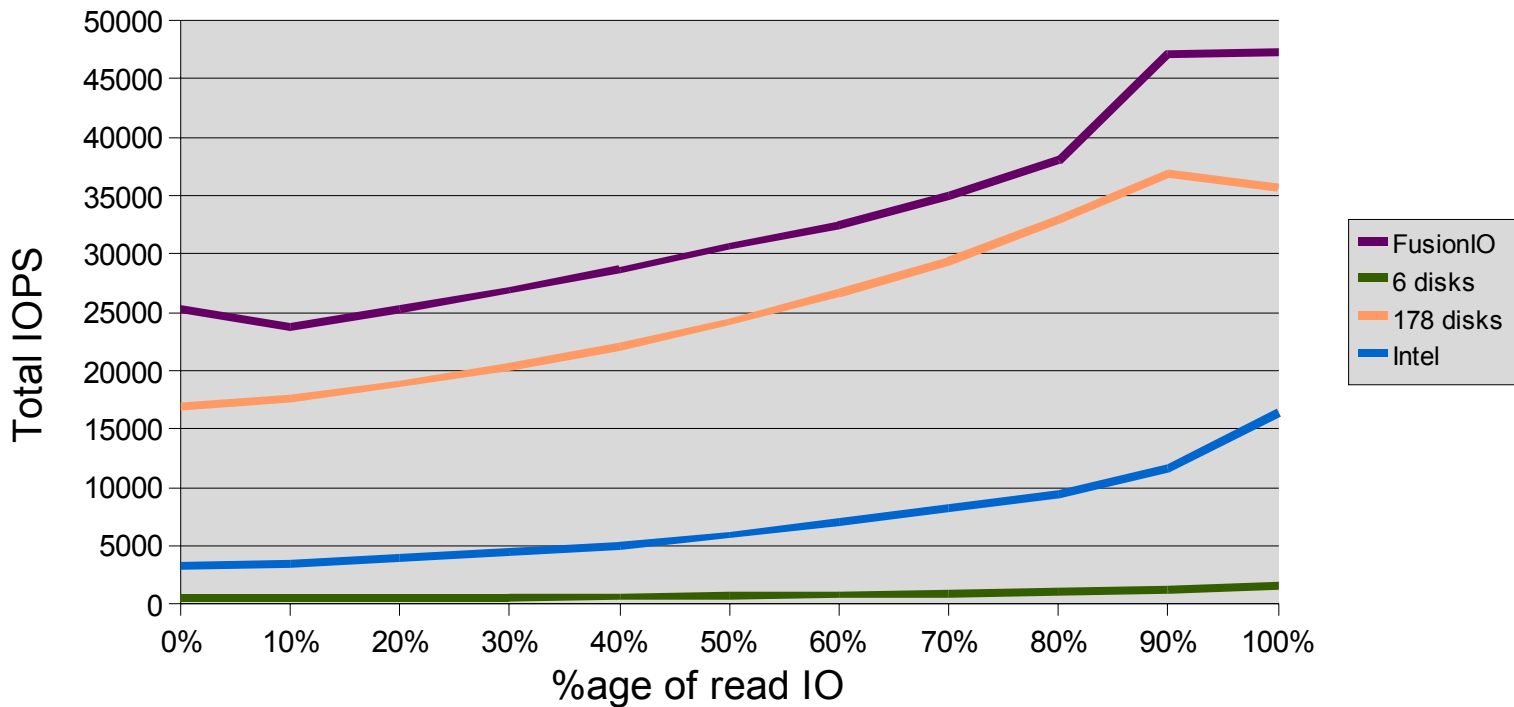


† FusionIO 80GB PCIe card, Intel 32GB SATA SSD in external storage

# Mixed Read/Write IOPS

- Measure IOPS for different read/write workload mixes
- Note interference of writes with reads

Multi threaded mixed read/write workload (8k blocks)



† FusionIO 80GB PCIe card, Intel 32GB SATA SSD in external storage

# Oracle Microbenchmarks

## External Sort on Flash

- Sorts with Temp tablespace on flash
  - 1 140 GB SAS disk vs Fusion IO 80GB PCIe card
  - ORDER BY query on 3.2 million rows
  - Sequential read/write workload
  - Sequential IO bandwidth improved by 5x
    - Improvement can be matched by using 5 disks
  - Elapsed time cut by 2.5x
    - Query switches from being mostly IO bound to mostly CPU bound

# Oracle Microbenchmarks

## Index Scan on Flash

- Text index on flash
  - 1 140GB SAS disk vs 80 GB FusionIO PCIe card
  - Workload was 10,000 actual queries issued to oracle.com
    - Benchmark corpus had 2 million documents (40GB)
    - Text index size was 7.7 GB
  - Index scans on a large B-Tree
  - Random read-only workload
  - Elapsed time for queries improved by up to 35x
  - IO Latencies improved by 56x
  - IOPS improved by 37x
  - Queries switched from being completely IO bound to 50/50 split between IO & CPU

# OTLP Performance Intel Enterprise SSDs

- 2 socket Woodcrest (Intel Xeon x5365 )
    - 4 cores
  - 64Gb memory
  - Oracle 11G
  - 2.2TB Database
    - 650 15K FC drives
- vs.
- 95 SATA SSD's (32GB SLC)
- 13% Higher TpM
  - 7x Fewer IO devices
  - Same CPU
  - 3x Lower Tx. response time
  - 2.5x Fewer Oracle foregrounds
  - 1.1x Higher IOPS

# OTLP Performance

## FusionIO/Disk Hybrid

- Single Socket Tigerton (Intel Xeon x7350)
    - 4 cores
  - 32 Gb memory
  - Oracle 11G
  - 100 GB database
    - 200 15K SAS Drives
- vs.
- 8 15K SAS Drives
  - 1 FusionIO PCIe card (80Gb SLC)
- Same TpM
  - 20x Fewer IO devices
  - 15% Less CPU
  - 3x Lower Tx. response time
  - 3x Fewer Oracle foregrounds
  - 1.3x Higher IOPS

# OTLP Performance

## Violin DRAM appliance

- DRAM Not FLASH
  - Single Socket Tigerton (Intel Xeon x7350)
    - 4 cores
  - 32 Gb memory
  - Oracle 11G
  - 100Gb Database
    - 200 15K SAS Drives
- vs.
- 1 V1010 PCIe memory appliance (120GB DRAM)
- 27% Higher TpM
  - 200x Fewer IO devices
  - Same CPU
  - 6x Lower Tx. response time
  - 5x Fewer Oracle foregrounds
  - 1.7x Higher IOPS
  - There was a lot of headroom left in the Violin device, probably enough for two times IOPS

# What can Flash Storage do for you?

- Reduce Transaction Latency
  - Improve customer satisfaction
- Remove performance bottlenecks caused by serialized reads
- Save money/power/space by storing High Intensity objects on fewer Flash devices