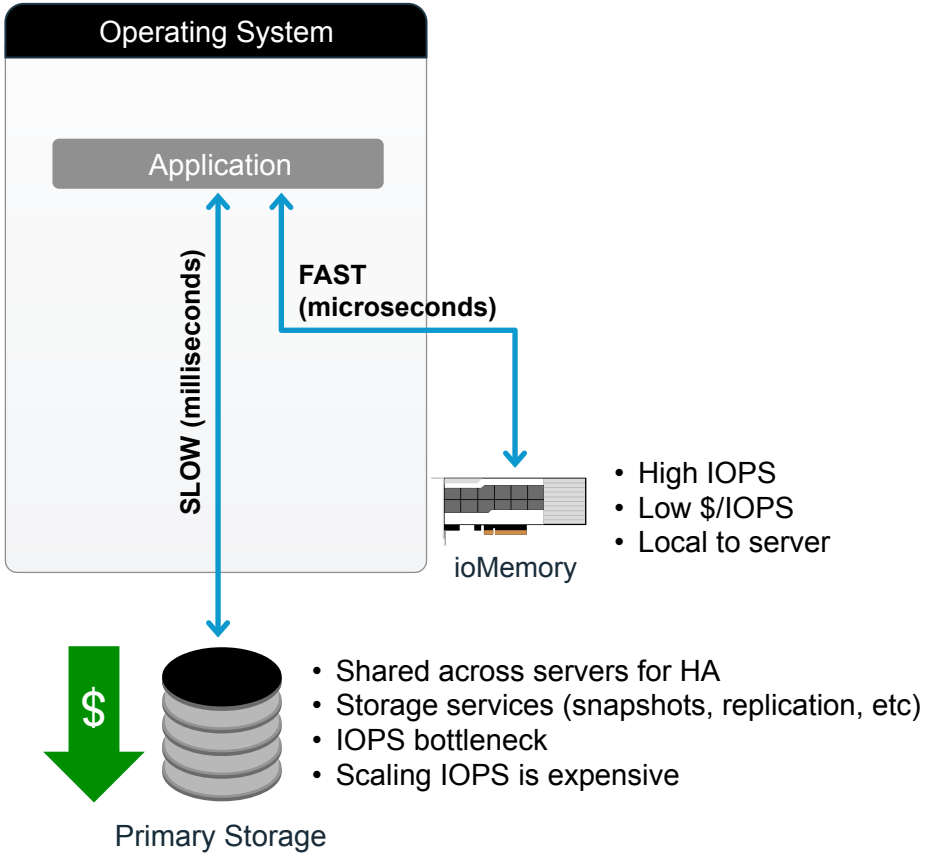
Flash based Caching: Opportunities and Challenges

Nisha Talagala, Swami Sundararaman

Why Flash Caching?



- IOPS closer to application
- Read from cache; write to primary storage
- No need to reconfigure storage or applications
- Preserve application mobility
- Reduce storage costs

Flash Caches are Different

FUSION-IO

- Flash caches are different
  - Higher write pressures than their storage counterparts
    - Admitted read-misses are writes, writes are writes
  - Writes (misses, write through) have endurance and performance cost in garbage collection
  - Cache pollution has hit rate costs; endurance and additional performance costs in flash caches
- The workload presented to flash devices are write intensive
  - Smaller cache size suffers from even higher CLWA (7x-8x)
    - CLWA: Cache Layer Write Amplification

TPC-Backup: cache layer write-amplification under ADMIT_ALL

Original Writes	Cache Size (GB)	Cache Writes (GB)	CLWA	Hit Rate
36.8	80	322.13	8.75	14.03
	100	300.11	8.16	20.67
	120	275.83	7.5	27.98

Agenda

- Cache workload impacts on flash
- Reducing cache and flash layer write amplification
- Memory efficiency in cache algorithms
- Write back caches
- Cache filters and intelligence
- Futures

Multiplicative Write Amplification

FUSION-IO

Original Writes (GiB)	Cache Size (GiB)	Cache Writes (GiB)	GC Writes (GiB	Total Writes (GiB)	CLWA	FLWA	Multiplier	Hit Rate (%)
36.8	120	275.83	1352.01	1627.84	7.5	5.9	44.25	27.98
	100	300.11	1459.13	1759.24	8.16	5.86	47.82	20.67
	80	322.13	1553.98	1876.11	8.75	5.82	50.93	14.03

more writes due to 'miss'

Even more writes due to garbage collection

Collective impact: CLWA * FLWA

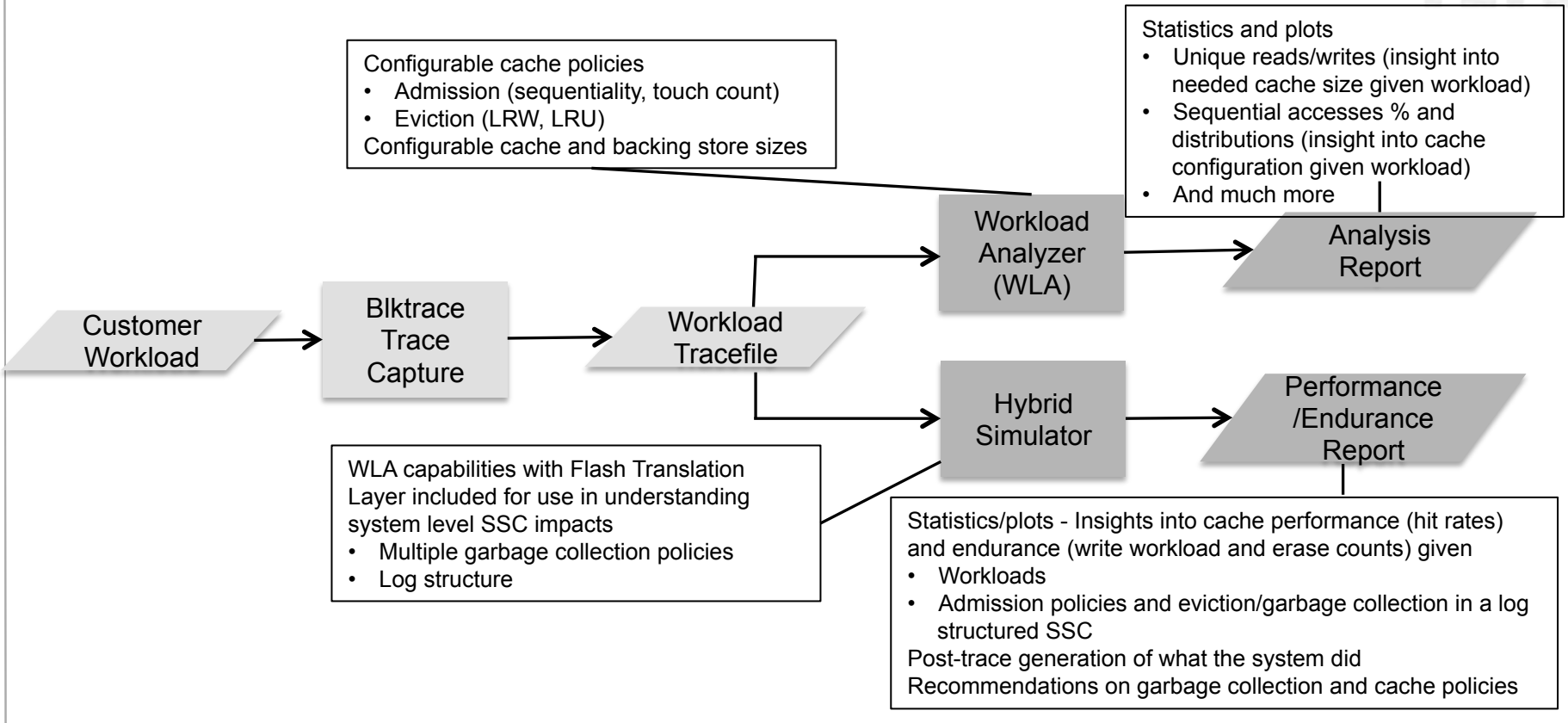
Improving Flash Cache Endurance

FUSION-IO

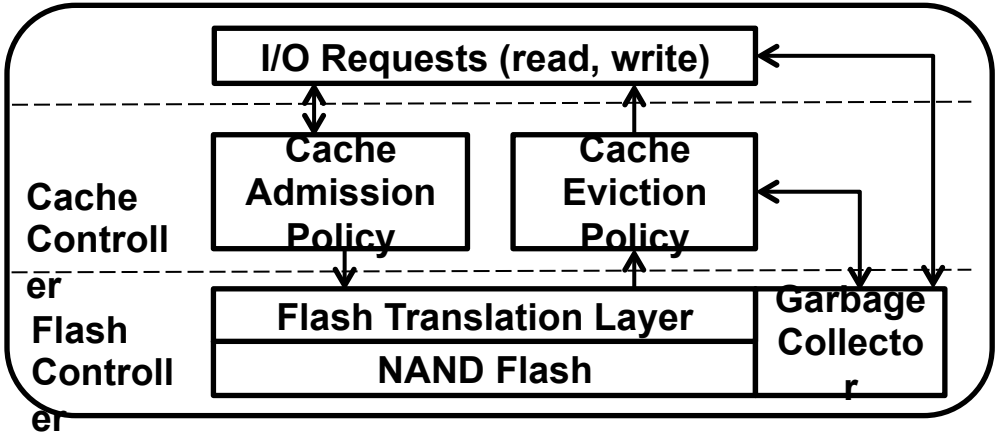
- What are the workload characteristics that a flash-based cache is likely to encounter ?
- How do different cache admission and eviction policies affect read cache hit rate and write workload ?
- How do various garbage collection strategies impact writes and erases to media ?
- What combination of admission control polices, eviction policies, and garbage collection strategies can be used to improve hit rates while reducing writes to the solid state cache (SSC) ?

Analysis tools

FUSION-IO



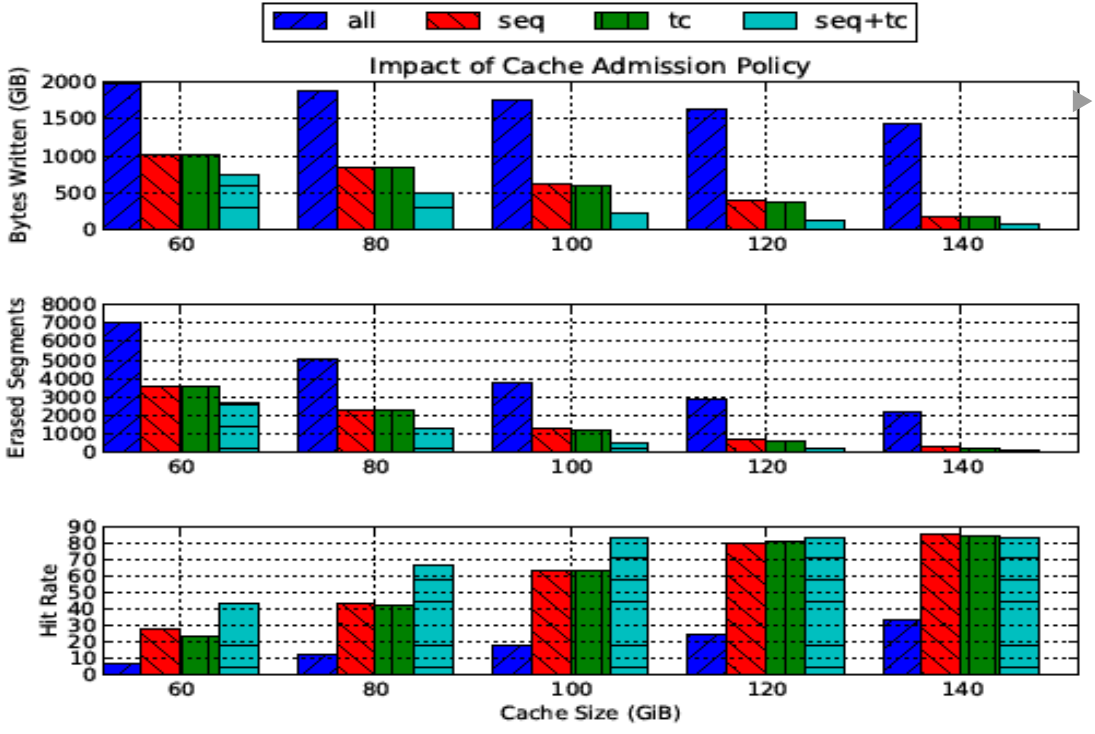
Cache Systems Effects



High Endurance flash-based cache characteristics are influenced by

- User workloads
- System reserve capacity
- Garbage collection strategies (e.g. victim segment selection policy (tail drop) runs risk of cleaning recently used data thus increased read misses and increased write load)
- Cache admission policies (e.g. directly impacts write workload when read misses)
- Cache eviction policies (e.g. can impact write workload when eviction of recently used)

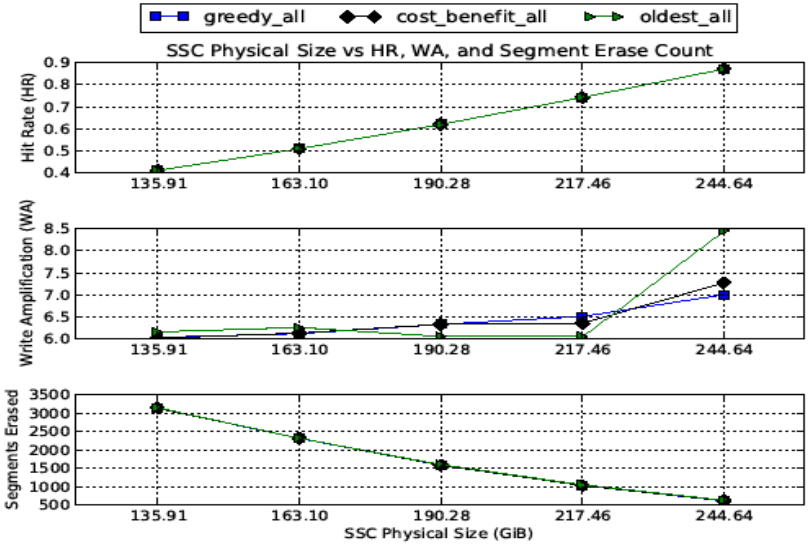
Cache Admission Control



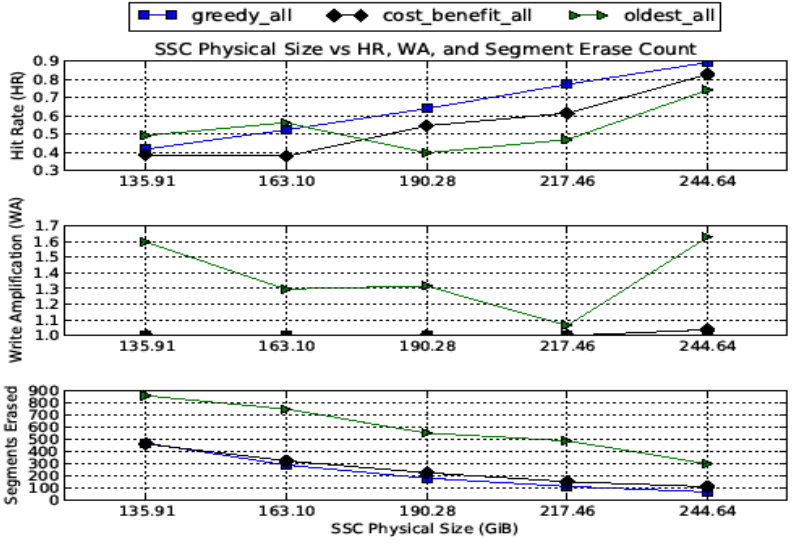
- Admission control
  - reduced bytes written
  - Reduced segment erase counts
  - and improved hit rate.

- Tpce-with backup workload

Reducing Flash Layer Write Amplification



Cache-based eviction

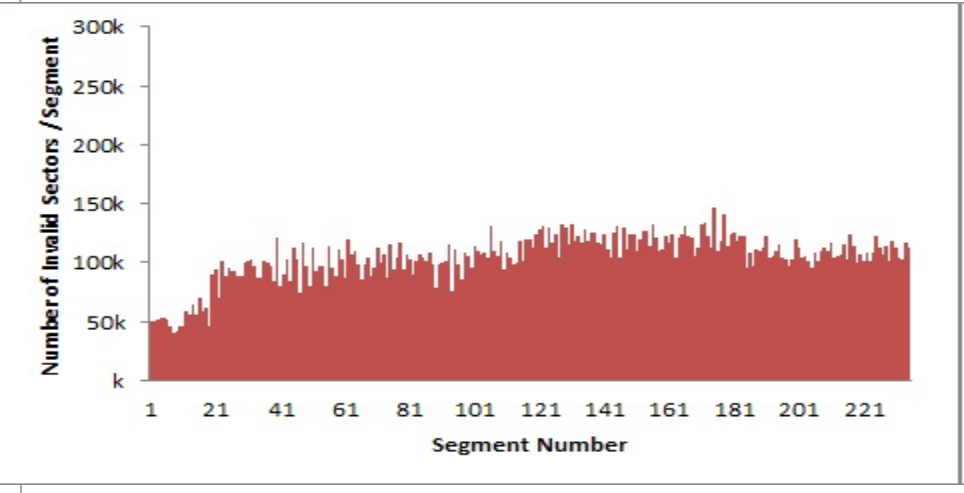


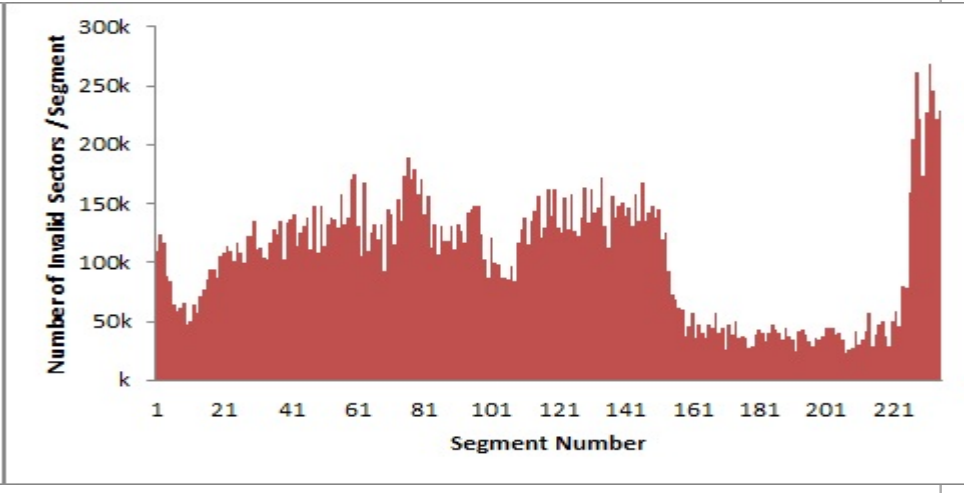
GC-based eviction

- Cache based eviction vs Garbage Collection based eviction

FTL Garbage Collection under Cache

FUSION-IO





- Invalidity distribution differs depending on cache algorithm
- Renders Flash based GC less efficient

Combined effects

FUSION-IO

Admit all
Tail drop
Cache-based
eviction

Original Writes (GiB)	Cache Size (GiB)	Cache Writes (GiB)	GC Writes (GiB	Total Writes (GiB)	CLWA	FLWA	Multiplier	Hit Rate (%)
36.8	120	275.83	1352.01	1627.84	7.5	5.9	44.25	27.98
	100	300.11	1459.13	1759.24	8.16	5.86	47.82	20.67
	80	322.13	1553.98	1876.11	8.75	5.82	50.93	14.03

sequentiality +
touch count
Cost-benefit
GC-based eviction

Original Writes (GiB)	Cache Size (GiB)	Cache Writes (GiB)	GC Writes (GiB	Total Writes (GiB)	CLWA	FLWA	Multiplier	Hit Rate (%)
36.8	120	37.65	44.8	82.45	1.02	2.19	2.24	59.78
	100	37.65	169.05	206.7	1.02	5.49	5.62	59.78
	80	70.1	169.19	239.29	1.9	3.41	6.5	46.59

- Reduced WA (CLWA, FLWA) by over 10x
- Improved hit rate for small – mid cache sizes

Observations

FUSION-IO

- More write-intensive from a flash point of view
  - Increasing cache size provides improvement but at cost
- Admission control can reduce CLWA
- Eviction control can reduce FLWA
- Cache-based eviction is not optimal
- Use of capacity reservation guarantees QoS on hit rate
  - At the expense of FLWA.

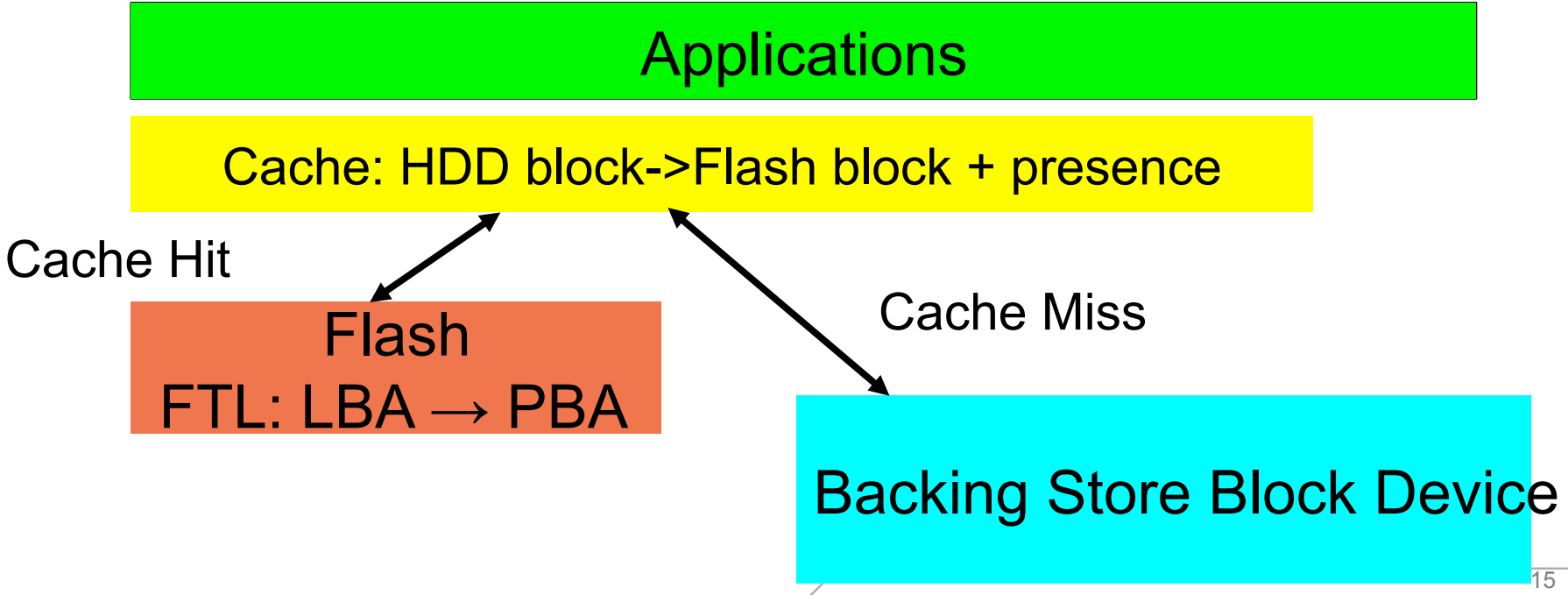
Agenda

- Cache workload impacts on flash
- Reducing cache and flash layer write amplification
- Memory efficiency in cache algorithms
- Write back caches
- Cache filters and intelligence
- App level caches
- Futures

Example - Block based caches

FUSION-IO

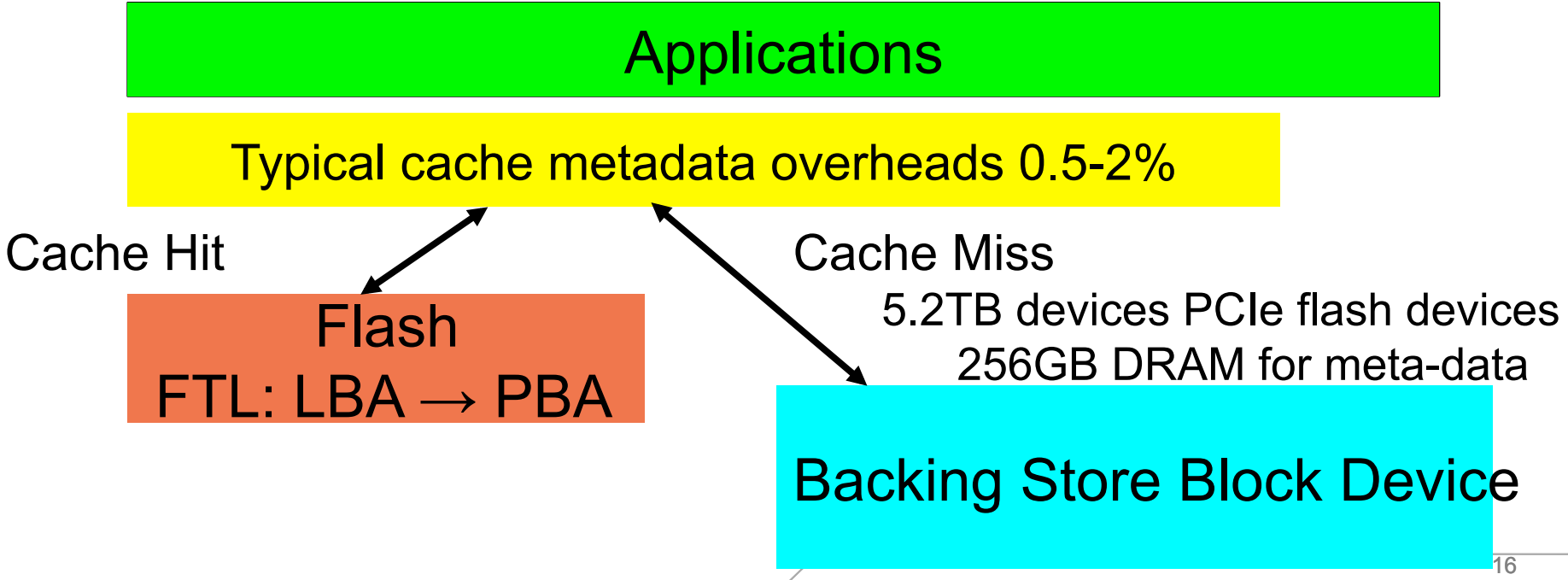
Block cache – index contains cached block location



Mapping tables and overheads

FUSION-IO

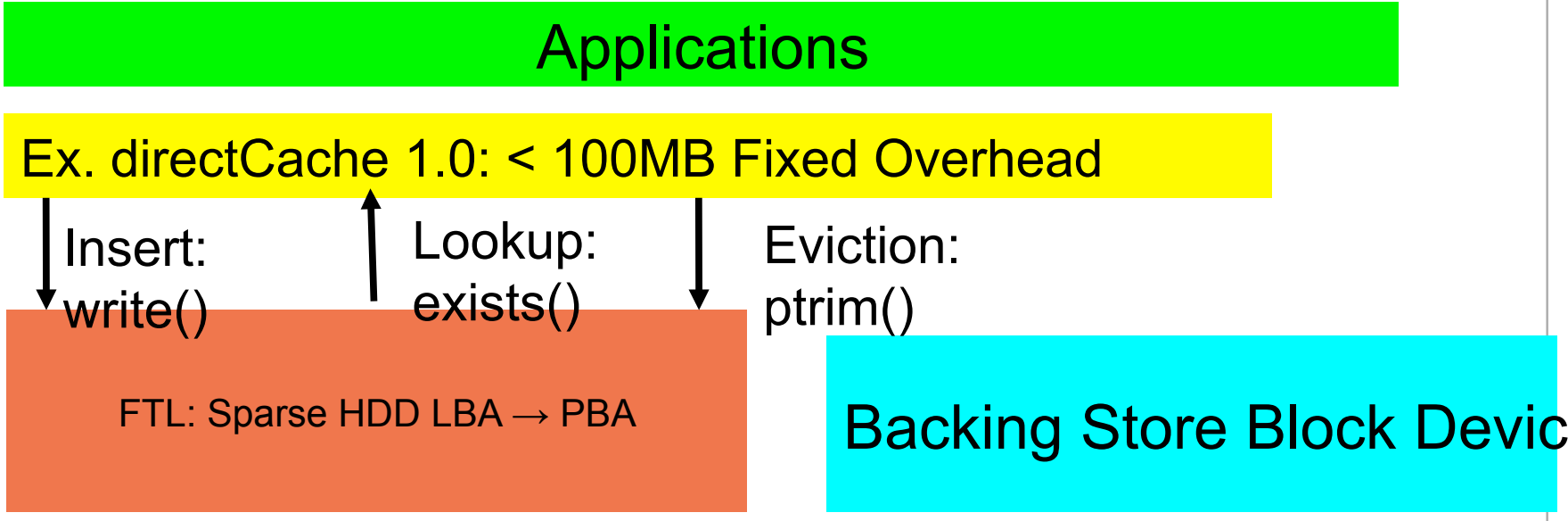
Mapping required to translate disk locations to flash locations – overhead per flash block, per disk block



Reduced Metadata Caches

FUSION-IO

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



Memory Efficient Admission Policies

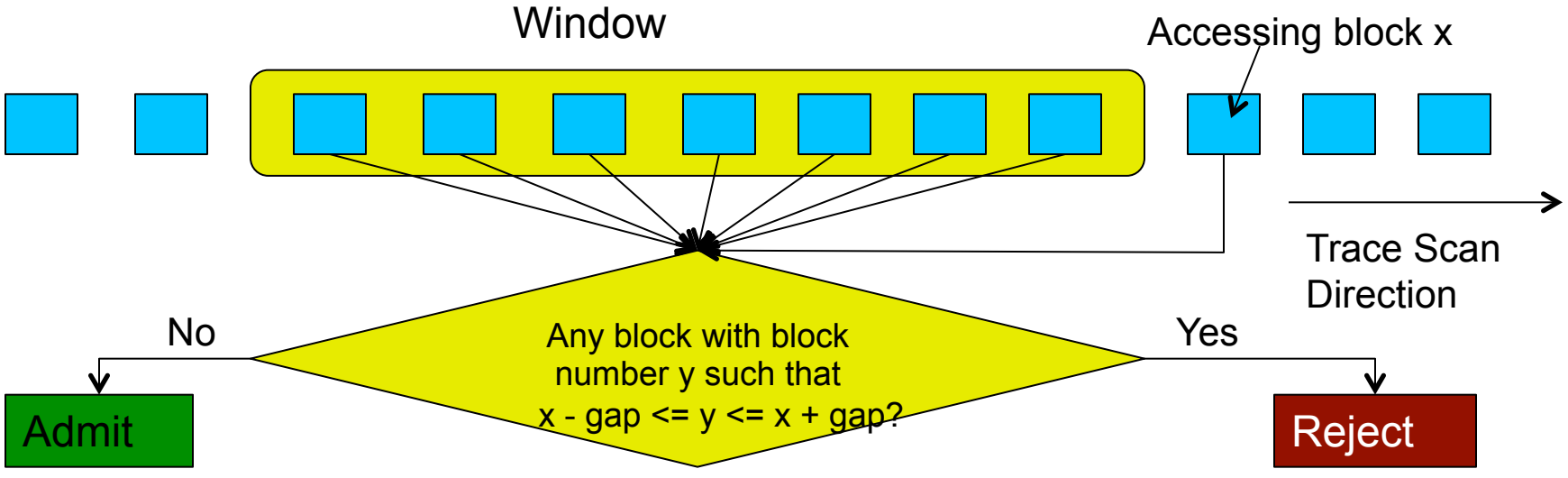
FUSION-IO

- SEQuentiality Rejection: *SEQR*
  - Parameters: window size and gap
- Selective SEQuentiality Rejection: *SSEQR(L = x)*
  - Additional parameter: admitted sequential lengths, x MB
- Touch Count: *TC*
  - Parameters: touch-count segment length, bits per segment, threshold, bitmap count, bitmap rotation mechanism.
- *SEQR + TC*
- *SSEQR + TC*

SEQUENTIALITY DETECTION ALGORITHM

FUSION-IO

Goal: Selectively reject sequential stream (*pollution avoidance*)



- **Optimal value of Window: Tunable**
  - **Number of CPU Cores**
  - **Number of database processes**
  - **Process characteristics (Accesses from one process)**
- **Optimal value of Gap: Tunable**

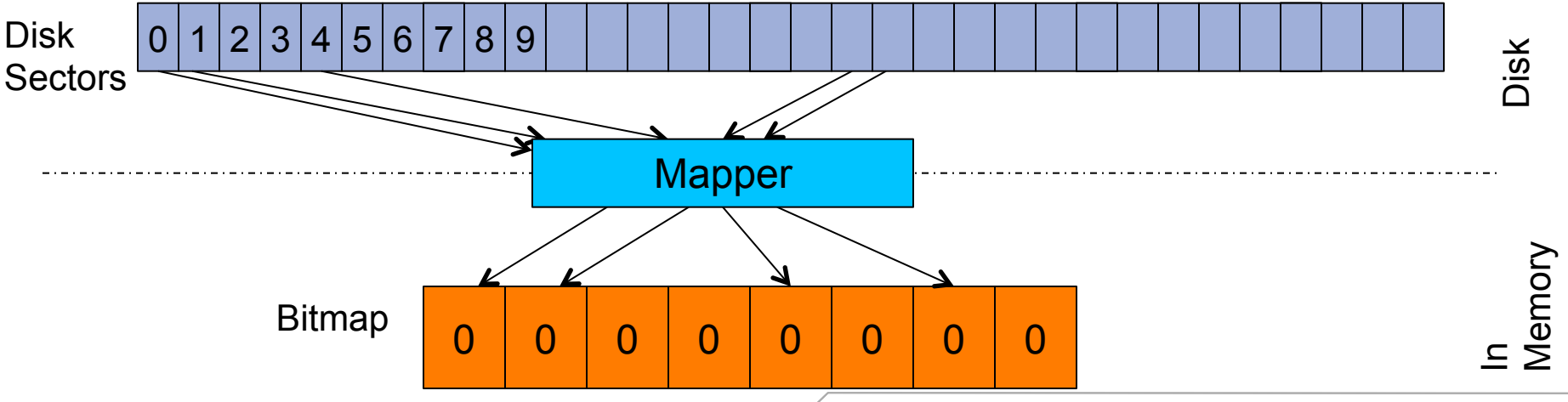
TOUCH COUNT ALGORITHMS

FUSION-IO

Goal: Selectively add "quality" blocks (*admittance control*)

Disk sectors mapped to bits in touch count bitmap

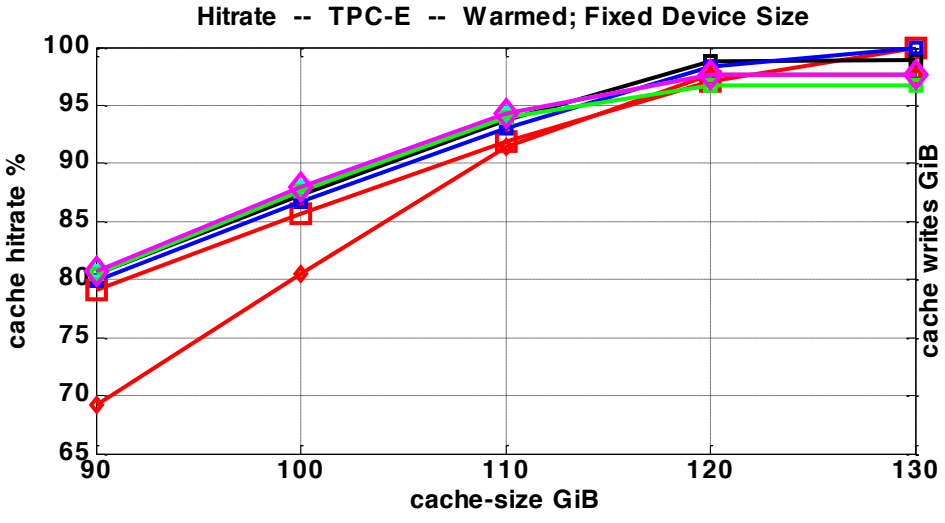
Segments mapped to bits (simple filter) to reduce memory consumption

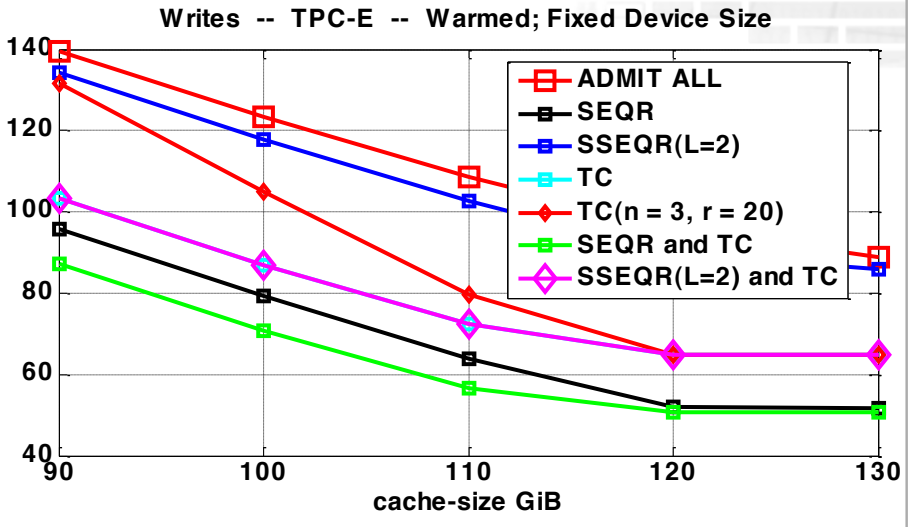


TPC-E

Unique data set (*AA*) = 127.3 GiB
Nonseq data set (*SEQR*) = 120.3 GiB
Read/write ratio = 84.5%

FUSION-IO





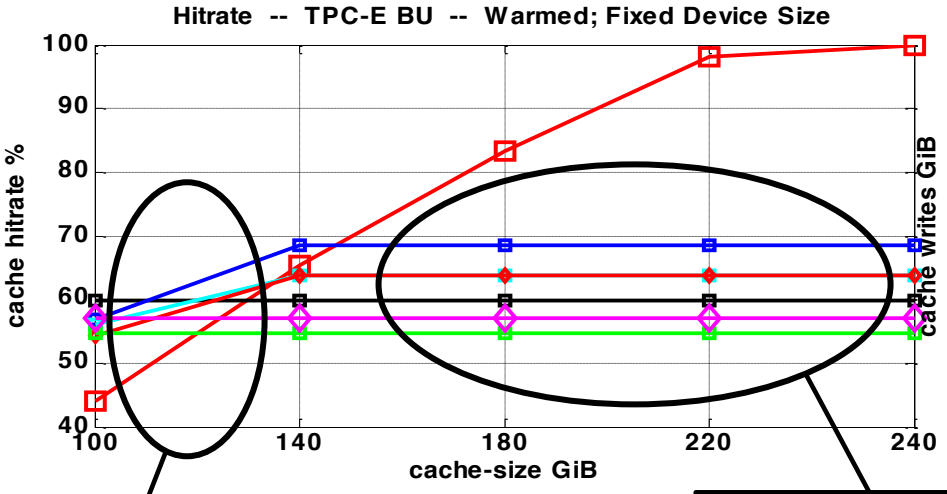
AA vs. *SEQR*: 2% HR reduction; 64% write reduction

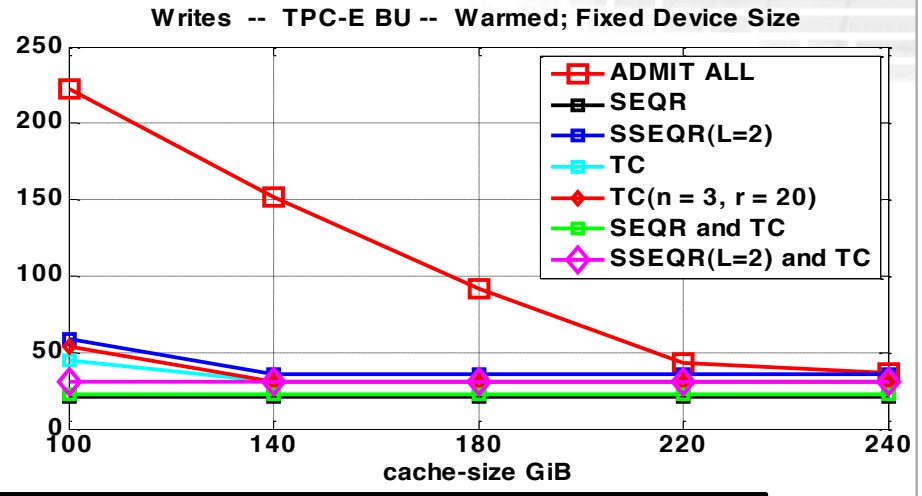
- Larger cache leads to increased hit-rate and fewer writes
  - Depends on workload and policies in effect!
- More restrictive admission policies have comparable hit-rates with greatly reduced writes
- More restrictive admission policies perform better (hit-rate) than ADMIT ALL for smaller cache sizes, without consideration to writes

TPCE-Back up

Unique data set (*AA*) = 226.5 GiB
Nonseq data set (*SEQR*) = 92.8 GiB
Read/write ratio = 96.4%

FUSION-IO





ALL other policies out-perform *AA* for smaller caches

Differences here are artifacts of the admission policies – "best" is to admit *some* sequentiality, and filter on access frequency

- Great discrepancies present for GiB > 140 are due to smaller critical cache sizes of restrictive admission policies AND great sequential scans
- More restrictive admission policies perform better (hit-rate wise) than ADMIT ALL (for sub-critical cache sizes), with 375% reduction in writes
- In presence of additional pollution, the admission policy is VERY important to reduce writes

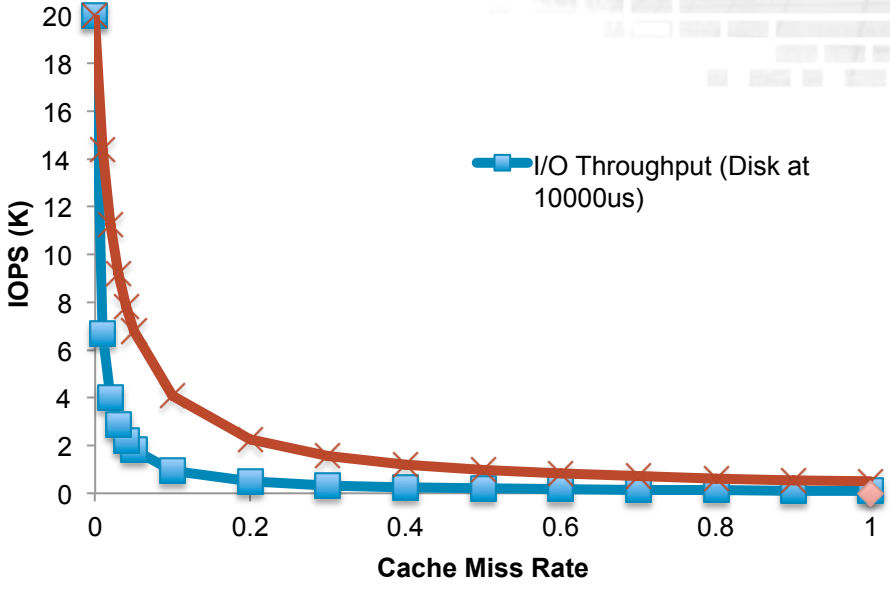
Agenda

- Cache workload impacts on flash
- Reducing cache and flash layer write amplification
- Memory efficiency in cache algorithms
- Write back caches
- Cache filters and intelligence
- Futures

Data Tiering for Non-Volatile Memories

FUSION-IO

- Flash is valuable as cache, but even small miss rates result in significant performance loss
- Write back cache policies provide high hit rate, but can result in incoherent back end storage state at failure
- Enable write back caching with coherent back end state
  - Ordered write back
  - Journaled write back
  - Journaled write back with application provided consistency hints
- Provides improved performance over write through with consistency guarantees



Write Policies for Host Side Flash Caches. R. Koller, L. Marmol*, R. Rangaswami*, S. Sundararaman+, N. Talagala+, M Zhao*. FAST 2013.*
https://www.usenix.org/conference/fast13/write-policies-host-side-flash-caches

**Florida International University, +Fusion-io*

References

- ***Write Policies for Host Side Flash Caches***. FAST 2013

  R. Koller, L. Marmol*, R. Rangaswami*, S. Sundararaman+, N. Talagala+, M Zhao**

  ** Florida International University, +Fusion-io*

  https://www.usenix.org/conference/fast13/write-policies-host-side-flash-caches

- ***Improving Endurance of High Performance Flash-based Cache Devices***. SYSTOR 2013

  J. Yang, N. Plasson, G. Gillis, N. Talagala, S. Sundararaman, R. Wood

  Fusion-io

  Camera-ready copy to be available soon

  *

- ***Admission Polices for Solid State Cache Devices***. NVM Workshop 2013

  G. Gillis, S. Sundararaman, N. Talagala, A. Mudrankit, J. Ludwig

  Fusion-io

  http://nvmw.ucsd.edu/2013/assets/abstracts/56

Intelligent Caching with Controls

FUSION-IO

- Fusion-io ioTurbine software addresses FILE, VOLUME and DISK I/O
  - Write through caching: Transparent and non-disruptive to I/O path
  - Intercepts storage calls
    - Closest to the application, in the guest OS (workload specific)
    - Caches the Active Working Set; most active data is cached
    - Allow storage to write data more efficiently
- Selective cache control
  - Control which files, volumes, or disks should be cached

Pagefile.sys ☑	Database redo ☒
Mirrored volume ❶	Kernel32.dll ☑

Futures

- QoS in cache operating well with FTLs
- Cache sharing
- New memories


THANK YOU
fusionio.com | REDEFINE WHAT'S POSSIBLE