Adaptive policies for balancing performance and lifetime of mixed SSD arrays through workload sampling

Sangwhan Moon
A. L. Narasimha Reddy

Texas A&M University
Outline

• Introduction
  – Mixed SSD Arrays
  – Workload distribution of mixed SSD array

• Problem Statement

• Selective caching policies

• Our approach
  – Online sampling
  – Adaptive workload distribution

• Evaluation

• Conclusion
Different classes of SSDs

Cost ($/GB)

Device Writes Per Day (DWPD, higher is better)
Mixed SSD array

• High-end SSDs cache
  – Faster: PCIe interface
  – Reliable: SLC eMLC (write endurance = 100K)
  – Expensive per gigabyte

• Low-end SSDs main storage
  – Slower: Serial ATA interface
  – Less reliable: MLC TLC (write endurance < 30K)
  – Cheap per gigabyte
Workload distribution of mixed SSD array

- LRU Caching Policy

\[ w_C = \frac{m_r r + w}{N_C \cdot C_C} \]

\[ w_S = \frac{m_w w}{N_S \cdot C_S} \]

Lifetime = \( \min \left( \frac{l_C}{w_C}, \frac{l_S}{w_S} \right) \)

\( r, w \) Read/write workload
\( w_C, w_S \) Writes per flash cell
\( m_r, m_w \) Cache read/write miss rate
\( N_C, N_S \) The number of SSDs
\( C_C, C_S \) The capacity of SSD
\( l_C, l_S \) Write endurance of cache/storage
Workload distribution of mixed SSD array

- 1 high-end SSD cache for 3 low-end SSDs

\[ w_c = \frac{0.5 \cdot 100 \text{MB/s} + 250 \text{MB/s}}{1 \cdot 100 \text{GB}} \]

\[ \text{read, write} \]

1. \( r \)

2. \( w \)

High-end SSDs

\[ \text{read miss} \]

3. \( m \cdot r \)

4. \( m \cdot r \)

5. \( (m \cdot r + m \cdot w) \cdot d \)

Low-end SSDs

\[ w_c = \frac{0.85 \cdot 250 \text{MB/s}}{1 \cdot 100 \text{GB}} \]

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-end SSD (SLC)</td>
<td>Capacity</td>
<td>100 GB</td>
</tr>
<tr>
<td></td>
<td>Write Endurance</td>
<td>100 K</td>
</tr>
<tr>
<td>Low-end SSD (MLC)</td>
<td>Capacity</td>
<td>200 GB</td>
</tr>
<tr>
<td></td>
<td>Write Endurance</td>
<td>10 K</td>
</tr>
<tr>
<td></td>
<td>Read/write (MB/s)</td>
<td>100 / 250</td>
</tr>
<tr>
<td>Workload</td>
<td>Read/write cache hit rate</td>
<td>50% / 15%</td>
</tr>
<tr>
<td></td>
<td>Read / write length</td>
<td>4KB / 64KB</td>
</tr>
</tbody>
</table>

\[ \text{Lifetime} = \min \left( \frac{1.47 \text{years}}{\text{high-end}}, \frac{6.34 \text{years}}{\text{low-end}} \right) \]
Problem statement

• High-end SSDs cache can wear out faster than low-end SSDs main storage
  – Caching less results in poor performance
  – Caching more results in poor reliability
• Static workload classifiers can be less efficient
• The characteristics of workload can change over time

• Objectives
  – Balance the performance and lifetime of cache and storage at the same time

  *metric*: Latency over Lifetime (*less is better*)
Selective caching policies

- Request Size based Caching Policy
- Hotness based Caching Policy

*Static workload classifiers cannot distribute workload across cache and storage precisely*

I/O requests whose sizes are 4KB are dominating

90% of workload is reference once and never accessed
Selective caching policies

- Control trade-offs between performance and lifetime

\[ p \text{ (threshold): the probability of caching data} \]

\[ p \text{ is more: cache wears out faster, performance enhances} \]
\[ p \text{ is less: cache wears slower, performance degrades} \]
Online sampling

- Estimate latency over lifetime for each sampling cache

Sampling Rate: 10%

- Employ best value of $p$, the proximity of caching

0.1 0.2 0.3 0.9 1.0

Sampling Cache Sampling Cache Sampling Cache ... Sampling Cache Sampling Cache Selective Cache

- $p$, 1.0 – $p$

LRU LRU LRU ... LRU LRU LRU

Main Storage
Simulation environment

• Trace-driven simulator
• Microsoft Research Cambridge I/O Block Trace
  – 13 enterprise applications trace for a week
• Cache provisioning = 5%
  – Cache size / Storage size
• Unique data size of workload / Storage Size = 0.5
• Caching policies
  – LRU, size-based (+ sampling), hotness-based (+ sampling), probabilistic (+ sampling)
Adaptive threshold

Hardware monitoring
- latency
- lifetime
- metric

Web server
- latency
- lifetime
- metric

Static threshold based analysis

Sampling based analysis
Different workload traces

- Overall, reduced latency over lifetime by 60%.
  - Very effective on some traces (mds, stg, web, prn, usr, proj, src1, src2)
  - Less effective on very skewed workload (wdev, rsrch, ts, hm, prxy)
Different sampling rates

- Higher sampling rate results in more accurate estimation (beneficial) and less space for adaptive cache (harmful)
Conclusion

• We showed that high-end SSD cache can wear out faster than low-end SSD main storage.
• We proposed sampling based selective caching to balance the performance and lifetime of cache and storage.
• Trace-based simulation showed that the proposed caching policy is effective.
Q & A