HEC: Improving Endurance of High Performance Flash-based Cache Devices

Jingpei Yang, Ned Plasson, Greg Gillis, Nisha Talagala, Swaminathan Sundararaman, Robert Wood
Why Flash Caching?

- Provide improved application performance – high quality data closer to the application
- Read from cache; write to primary storage
- No need to reconfigure storage or applications
- Reduce storage costs

### Diagram:

<table>
<thead>
<tr>
<th>Host Processor</th>
<th>Host Processor L1, L2 Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Memory</td>
<td>Solid State Cache</td>
</tr>
<tr>
<td>NAS (Backing Store)</td>
<td>SSD (Backing Store)</td>
</tr>
</tbody>
</table>

- Storage services (snapshots, replication, etc)
- High IOPS
- Low $/IOPS
- Local to server
- IOPS bottleneck
- Scaling IOPS is expensive

Primary Storage
Flash Caches are Different

- SSCs are generally much smaller than their storage counterparts
- **Workloads**
  - Cache workloads are more write intensive – read misses become writes yielding Cache Layer Write Amplification (CLWA)
- **Endurance of media**
  - Limited endurance and getting more limited with new NAND node geometries
- **Un-coordination at the Caching and Flash Translation Layer (FTL)**
  - Can cause increased Flash Layer Write Amplification (FLWA)
- **Techniques for reducing FLWA and CLWA must consider impact to cache hit rate**
## Example – TPC-E Polluted Workload

The table below shows the performance metrics for different TPC-E polluted workloads. The workload conditions are as follows:

<table>
<thead>
<tr>
<th>Original Reads (GiB)</th>
<th>Original Writes (GiB)</th>
<th>Cache Size (GiB)</th>
<th>Cache Writes (GiB)</th>
<th>GC Writes (GiB)</th>
<th>Total Writes (GiB)</th>
<th>CLWA</th>
<th>FLWA</th>
<th>TCWA</th>
<th>Hit Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>331.9</td>
<td>36.8</td>
<td>80</td>
<td>322.13</td>
<td>1553.98</td>
<td>1876.11</td>
<td>8.75</td>
<td>5.82</td>
<td>50.93</td>
<td>14.03</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td>300.11</td>
<td>1459.13</td>
<td>1759.24</td>
<td>8.16</td>
<td>5.86</td>
<td>47.82</td>
<td>20.67</td>
</tr>
<tr>
<td>120</td>
<td></td>
<td></td>
<td>275.83</td>
<td>1352.01</td>
<td>1627.84</td>
<td>7.5</td>
<td>5.9</td>
<td>44.25</td>
<td>27.98</td>
</tr>
</tbody>
</table>

**GC** – garbage collection  
**CLWA** – Cache Layer Write Amplification (cache writes / original writes)  
**FLWA** - Flash Layer Write Amplification (GC writes / cache writes)  
**TCWA** – Total Combined Write Amplification (CLWA * FLWA)

### Conditions of simulation
1. No admission control  
2. Tail-drop GC policy  
3. Eviction performed at cache layer

A **TCWA** of 40-50x and a low hit rate demonstrate that opportunity exists for reducing writes and improving hit rate.
Techniques for reducing CLWA and FLWA

- **CLWA**
  - How do admission policies affect device endurance?
  - Can cache eviction benefit from knowledge of the FTL?

- **FLWA**
  - Can garbage collection further reduce write amplification?
  - What if eviction was performed by the FTL?

What is the combined impact of these techniques on hit rate and endurance?
Reducing CLWA – Cache Admission Policies

<table>
<thead>
<tr>
<th>Admission Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admit All</td>
<td>All data is admitted to cache</td>
</tr>
<tr>
<td>Selective Sequential Rejection (SSEQR)</td>
<td>All non-sequential and short sequential sector ranges are admitted</td>
</tr>
<tr>
<td>Touch Count (TC)</td>
<td>Sectors or collections of sectors are admitted after they have been accessed N times</td>
</tr>
<tr>
<td>Sequential and Touch Count (SSEQR + TC)</td>
<td>Admission if data meets criteria for both SSEQR and TC</td>
</tr>
</tbody>
</table>

Admission policies can be used for improving the quality of data admitted to the cache thus improving hit rate, decreasing read misses and in turn writes to media.
Reducing CLWA - Admission Policies

- Selective admission policies
  - Reduce bytes written
  - Reduce segment erase counts
  - Improve read hit rate

Impact of Cache Admission Policy

- TPC-E Polluted Workload Trace

Patent Pending Technology
## Reducing FLWA – FTL GC Policies

<table>
<thead>
<tr>
<th>GC Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy</td>
<td>Victim segment is selected based on having the largest amount of “invalid” data</td>
</tr>
<tr>
<td>Cost Benefit</td>
<td>Segment selected based on “invalid” data percentage and age of data in the segment [Rosenblum LFS92]</td>
</tr>
<tr>
<td>Oldest (tail drop)</td>
<td>Segment selected is the oldest segment or tail of log</td>
</tr>
</tbody>
</table>

More selective GC policies can be used to reduce FLWA
Reducing FLWA – GC Policy

More selective GC policies
- Modest reductions in FLWA
- No noticeable change in hit rate and erased segments
Reducing FLWA – Cache Eviction Modes

**Cache-based Eviction**

- **Cache Controller**
  - Admission Control
  - Cache contents
  - Sector LBAs

- **Flash Controller**
  - Log Structured
  - Segment 0
  - Segment N

Eviction = LRW/LRU Replacement Policy (TRIM to Flash Controller for invalidation)

**GC-based Eviction**

- **Cache Controller**
  - Admission Control

- **Flash Controller**
  - Log Structured
  - Segment 0
  - Segment N

Garbage Collection/Eviction = more selective GC policy

TRIM → Garbage Collection = GC Policy (e.g. Tail Drop)
Reducing FLWA – Eviction Modes and Validity Distribution

Cache-based eviction mode (LRW)

GC-based eviction mode

TPC-E Workload

Improved victim segment selection and improved GC effectiveness from non-uniform distribution
Reducing FLWA – GC Policy and Eviction Mode

Cache-based eviction mode (LRW)

GC-based eviction mode

Significant reduction in FLWA and segments erased with GC-based eviction while maintaining hit rate

Patent Pending Technology
**Collective Endurance Impact**

**TPC-E Polluted Trace - Before**

<table>
<thead>
<tr>
<th>Admission Policy</th>
<th>Original Reads (GiB)</th>
<th>Original Writes (GiB)</th>
<th>Cache Size (GiB)</th>
<th>Cache Writes (GiB)</th>
<th>GC Writes (GiB)</th>
<th>Total Writes (GiB)</th>
<th>CLWA</th>
<th>FLWA</th>
<th>TCWA</th>
<th>Hit Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMIT_ALL</td>
<td>331.9</td>
<td>36.8</td>
<td>80</td>
<td>322.13</td>
<td>1553.98</td>
<td>1876.11</td>
<td>8.75</td>
<td>5.82</td>
<td>50.93</td>
<td>14.03</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>300.11</td>
<td>1459.13</td>
<td>1759.24</td>
<td>8.16</td>
<td>5.86</td>
<td>47.82</td>
<td>20.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>275.83</td>
<td>1352.01</td>
<td>1627.84</td>
<td>7.5</td>
<td>5.9</td>
<td>44.25</td>
<td>27.98</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TPC-E Polluted Trace - After**

<table>
<thead>
<tr>
<th>Admission Policy</th>
<th>Original Writes (GiB)</th>
<th>Cache Size (GiB)</th>
<th>Cache Writes (GiB)</th>
<th>GC Writes (GiB)</th>
<th>Total Writes (GiB)</th>
<th>CLWA</th>
<th>FLWA</th>
<th>TCWA</th>
<th>Hit Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSEQR+TC</td>
<td>36.8</td>
<td>80</td>
<td>70.1</td>
<td>169.19</td>
<td>239.29</td>
<td>1.9</td>
<td>3.41</td>
<td>6.5</td>
<td>46.59</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>37.65</td>
<td>169.05</td>
<td>206.7</td>
<td>1.02</td>
<td>5.49</td>
<td>5.62</td>
<td>2.24</td>
<td>59.78</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>37.65</td>
<td>44.8</td>
<td>82.45</td>
<td>1.02</td>
<td>2.19</td>
<td>2.24</td>
<td>59.78</td>
<td></td>
</tr>
</tbody>
</table>

**TCWA reduced while improving read cache hit rate using more selective admission, eviction, and GC policies**
Conclusions

• CLWA and FLWA exists and their combined effects can increase original write workloads by 10x to 50x
• We have identified techniques that provide for reduction of CLWA up to 20x and FLWA up to 10x
• We have demonstrated that combinations of such techniques reduce TCWA up to 20x yielding significant improvements in endurance and use of PBW
• We show that such techniques maintain or improve hit rate with coordinated cache and flash controller
• Future work exists in understanding such effects in write-back caching and SSC configurations with multiple cache instances