Flash Translation Layers III

Jin-Soo Kim (jinsookim@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Contents

▪ Superblock FTL

▪ DFTL

▪ μ-FTL
Superblock FTL

Reference: Superblock FTL: A Superblock-based Flash Translation Layer with a Hybrid Address Translation Scheme
Superblock FTL

- A superblock shares log blocks
- Up to M log blocks per superblock
- Block mapping at the superblock level, Page mapping within a superblock
- Hot/cold pages separation
- Map cache
- The amount of mapping information increased
Example

- \( W = \langle \{1,2\}, \{8\}, \{1,2\}, \{12\}, \{13\}, \{9\} \rangle \)
  - Write (\(\{1,2\}\), AB)
  - Write (\(\{8\}\), C)
  - Write (\(\{1,2\}\), DE)
  - Write (\(\{12\}\), F)
  - Write (\(\{13\}\), G)
  - Write (\(\{9\}\), H)
Mapping Information

- PGD (Page Global Directory)
- PMD (Page Middle Directory)
- PTE (Page Table Entry)
# Comparison

<table>
<thead>
<tr>
<th>Data blocks</th>
<th>Terminology</th>
<th>Log block scheme</th>
<th>FAST</th>
<th>Superblock FTL</th>
<th>Page mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANAND</td>
<td>Data blocks</td>
<td>Data blocks</td>
<td>Data blocks</td>
<td>D-blocks</td>
<td>Data blocks</td>
</tr>
<tr>
<td>Management scheme</td>
<td>In-order</td>
<td>In-order</td>
<td>In-order</td>
<td>Out-of-order</td>
<td>Out-of-order</td>
</tr>
<tr>
<td>The degree of sharing</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Min(P, S)</td>
<td>P</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Update blocks</th>
<th>Terminology</th>
<th>Replacement blocks</th>
<th>Log blocks</th>
<th>Sequential / random log blocks</th>
<th>U-blocks</th>
<th>Page mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANAND</td>
<td>Replacement blocks</td>
<td>Log blocks</td>
<td>Sequential / random log blocks</td>
<td>U-blocks</td>
<td>Update blocks</td>
<td></td>
</tr>
<tr>
<td>Management scheme</td>
<td>In-order</td>
<td>Out-of-order</td>
<td>Out-of-order</td>
<td>Out-of-order</td>
<td>Out-of-order</td>
<td></td>
</tr>
<tr>
<td>The degree of sharing</td>
<td>1</td>
<td>1</td>
<td>1 (sequential) or P (random)</td>
<td>Min(P, S)</td>
<td>P</td>
<td></td>
</tr>
</tbody>
</table>
DFTL: A Flash Translation Layer Employing Demand-based Selective Caching of Page-level Address Mappings
DFTL

- Page mapping
- Keeps full mapping on flash to reduce SRAM use
- Demand-based selective caching of page-level address mappings in SRAM
- Exploits temporal locality in workloads
- Data pages vs. Translation pages
DFTL Architecture
Garbage Collection

- Current data block: updated data pages
- Current translation block: updated translation pages

- GC invoked if the number of free blocks < GC_threshold
- Selecting a victim block: simple cost-benefit policy
μ-FTL

Reference: μ-FTL: A Memory-efficient Flash Translation Layer Supporting Multiple Mapping Granularities
μ-FTL

- Minimally-updated FTL
- Supports multiple mapping granularities
  - Based on extents
  - Reduce the amount of mapping information
- Requires more sophisticated index structure
  - μ-Tree is used to store the mapping info.
- Tunable memory footprint
  - Frequently accessed mapping info. cached
Example

- \( W = \langle \{1\}, \{2\}, \{8\}, \{1\}, \{2\}, \{12,13\}, \{9\} \rangle \)
  - write (\{1\}, A)
  - write (\{2\}, B)
  - write (\{8\}, C)
  - write (\{1\}, D)
  - write (\{2\}, E)
  - write (\{12,13\}, FG)
  - write (\{9\}, H)
μ-FTL Architecture

μ-Tree Cache

Bitmap update flush

Bitmap Cache

Cache full

Extent update

Write

Data

Bitmap update

Cache miss

μ-Tree

Partition 1

Partition 2

Partition 3

Partition 4

Free block list
Conclusion

- Garbage collection issues
- Mapping information management issues
- Platform-dependent issues