Unified Address Translation for Memory Mapped SSDs with FlashMap

Jian Huang
Anirudh Badam† Moinuddin K. Qureshi Karsten Schwan

† Georgia Tech
Microsoft
Bridging the DRAM-Disk Gap

DRAM
High Performance
Small Capacity

Disk
Low Performance
Large Capacity

Application Memory Component

Application Storage Component
Bridging the DRAM-Disk Gap

**DRAM**
- High Performance
- Small Capacity

**SSD**
- Good Performance
- Good Capacity

**Disk**
- Low Performance
- Large Capacity

Application Memory Component
Application Storage Component
Flash: Slow Memory or Fast Disk?

Flash behaves more like memory than disk
Flash: Slow Memory or Fast Disk?

Flash behaves more like memory than disk
Flash: Slow Memory or Fast Disk?

Flash behaves more like memory than disk

No Seek
Latency
Flash: Slow Memory or Fast Disk?

Flash behaves more like memory than disk

- No Seek Latency
- Internal Parallelism
Flash: Slow Memory or Fast Disk?

Flash behaves more like memory than disk

- No Seek Latency
- Internal Parallelism
- High IOPS
Flash: Slow Memory or Fast Disk?

Flash behaves more like memory than disk

No Seek LATENCY + Internal Parallelism + High IOPS

Use Flash as Memory [Badam et al., NSDI’11]

DRAM
High Performance
Small Capacity

SSD
Good Performance
Good Capacity

Disk
Low Performance
Large Capacity

Application Memory Component

Application Storage Component
Flash: Slow Memory or Fast Disk?

Flash behaves more like memory than disk

- No Seek Latency
- Internal Parallelism
- High IOPS

Use Flash as Memory [Badam et al., NSDI’11]

- DRAM
  - High Performance
  - Small Capacity
- SSD
  - Good Performance
  - Good Capacity
- Disk
  - Low Performance
  - Large Capacity

Application Memory Component

Application Storage Component
Memory Mapped SSDs
Memory Mapped SSDs

Application  Virtual Memory  Filesystem  SSD

Diagram showing the relationship between an application, virtual memory, filesystem, and SSD.
Memory Mapped SSDs

Application

Virtual Memory

Filesystem

SSD

mmap()
munmap()
msync()
Memory Mapped SSDs

Extended Memory

Application

Virtual Memory

Filesystem

SSD

mmap()
munmap()
msync()
Memory Mapped SSDs

Application

Virtual Memory

Filesystem

SSD

mmap()
munmap()
msync()

Extended Memory

Minimal Code Modifications

YAY! MORE MEM!!!
Memory Mapped SSDs

- Application
  - `mmap()`
  - `munmap()`
  - `msync()`
- Virtual Memory
- Filesystem
- SSD

Extended Memory + Minimal Code Modifications + Data Durability

YAY! MORE MEM!!!
No Free Lunch: Software Overhead

- Application
- Virtual Memory System
- File System
- Flash Translation Layer
- Flash
No Free Lunch: Software Overhead

Application

Virtual Memory System

File System

Flash Translation Layer

Flash

Virtual Address

Page Table & Memory Manager

Physical Address/File Offset

page fault
No Free Lunch: Software Overhead

Application

Virtual Memory System

File System

Flash Translation Layer

Flash

Virtual Address

Page Table & Memory Manager

Physical Address/File Offset

File Offset

File Index

Logical Block Address

Page fault
No Free Lunch: Software Overhead

Application → Virtual Memory System

Virtual Memory System → File System

File System → Flash Translation Layer

Flash Translation Layer → Flash

Virtual Address → Page Table & Memory Manager

Page Table & Memory Manager → Physical Address/File Offset

Physical Address/File Offset → File Offset

File Offset → File Index

File Index → Logical Block Address

Logical Block Address → Logical Block Address

Logical Block Address → FTL

Physical Block Address
Software Overhead Quantified

Application → Virtual Memory System → File System → Flash Translation Layer → Flash

3 address translations
+ 2 boundary checks
+ 2 permission checks

-------------------------------------------
= Latency: 15 – 20 microseconds

+ Increased Metadata Overhead
FlashMap: Unified Address Translation

1. Reduced Storage, only 1 mapping table

2. Reduced Latency, only 1 address translation + 1 permission check +1 boundary check
FlashMap: Unified Address Translation

1. Reduced Storage, only 1 mapping table

2. Reduced Latency, only 1 address translation + 1 permission check + 1 boundary check
Combining Page Table and File System

Diagram showing Process A interacting with the File System, which in turn interacts with the Flash Translation Layer and Flash storage.
Combining Page Table and File System
Combining Page Table and File System

Process A

Page Table

VM Region

File System

File

Flash Translation Layer

Flash
Combining Page Table and File System

Process A

Page Table

VM Region

File System

File

Flash Translation Layer

Flash

Process-specific, private
Combining Page Table and File System

Process A
Page Table
VM Region

Only for mapped file

File System

Flash Translation Layer
Flash
Preserving File System Permissions

Process A
VM Region

Process B
VM Region

Shared Page Table

Mapped File

File System

Flash Translation Layer

Flash
Preserving File System Permissions

Process A
VM Region
READ_ONLY

Process B
VM Region
READ_WRITE

Shared Page Table

Mapped File

File System

Flash Translation Layer

Flash
Preserving File System Permissions

Process A
VM Region
READ_ONLY

Permission Conflict !!!

Process B
VM Region
READ_WRITE

Shared Page Table
Mapped File
File System

Flash Translation Layer
Flash
Preserving File System Permissions

Only share the leaf-level page table pages!

- Process A
  - VM Region
  - READ_ONLY

- Process B
  - VM Region
  - READ_WRITE

Shared Page Table

Mapped File

File System

Flash Translation Layer

Flash
Preserving File System Permissions

Only share the leaf-level page table pages!

Process A
VM Region
READ_ONLY

Process B
VM Region
READ_WRITE

Shared Page Table

Mapped File

File System

Flash Translation Layer

Flash
Preserving File System Permissions
Preserving File System Permissions

Process A

Process B

Shared Leaf-level Page Table Pages

Mapped File

Flash Translation Layer

Flash
Page Table in FlashMap

- Process
- Page Directory
- Private Leaf-Level Page Table Pages
- Private Virtual Memory Regions
- Process’s Private Virtual Memory

Before Mapping a File
Page Table in FlashMap

Process

Page Directory

Private Leaf-Level Page Table Pages

Shared Leaf-Level Page Table Pages

Virtual Memory Regions Backed by File

Private Virtual Memory Regions

Process’s Private Virtual Memory + File Backed Memory

Only for mapped file

After Mapping a File
Preserving Memory Protection

Process A

- PGD
- PUD
- PMD
- PTE
- Offset

Process B

- PGD
- PUD
- PMD
- PTE
- Offset

Shared Leaf-level Page Table Pages

Mapped File

File System

Flash Translation Layer

Flash
Preserving Memory Protection

What if I require custom memory protection for a single page???
Preserving Memory Protection

Process A

PGD | PUD | PMD | PTE | Offset

Shared Leaf-level Page Table Pages

Mapped File

File System

Flash Translation Layer

Flash

Process B

PGD | PUD | PMD | PTE | Offset

Private Leaf-level Page Table
Combining FTL and Shared Page Table

Process A

Process B

Shared Leaf-level Page Table Pages

Mapped File

Mapping Table  GC  ECC  Wear Leveling

Flash Translation Layer

Flash
Combining FTL and Shared Page Table

Shared Leaf-level Page Table Pages → Mapped File

Overloaded PTE

Process A
PGD | PUD | PMD | PTE | Offset

Process B
PGD | PUD | PMD | PTE | Offset

Flash Translation Layer
GC  ECC  Wear Leveling

Flash
Combining FTL and Shared Page Table

Process A

Shared Leaf-level Page Table Pages

Overloaded PTE

Mapped File

Flash Translation Layer

Flash

GC  ECC  Wear Leveling
Putting It All Together

Process A

Read

DRAM

Shared Leaf-level Page Table Pages

Mapped File

FlashMap

Flash
Putting It All Together

Process A

DRAM

Mapped File

Flash

Shared Leaf-level Page Table Pages

FlashMap

PGD PUD PMD PTE Offset

Read

...
Putting It All Together

Process A

Read

DRAM

page fault

Shared Leaf-level Page Table Pages

Mapped File

Flash

FlashMap
Putting It All Together

Process A

PGD | PUD | PMD | PTE | Offset

Read

Shared Leaf-level Page Table Pages

update PTE

Mapped File

FlashMap

Flash
Putting It All Together

Process A

Write

DRAM

Shared Leaf-level Page Table Pages

Mapped File

FlashMap

Flash
Putting It All Together

Process A

Write

DRAM

DRAM hit

Shared Leaf-level Page Table Pages

Mapped File

FlashMap

Flash
Putting It All Together

Process A

Write

DRAM

DRAM miss

Mapped File

FlashMap

Flash

PGD  PUD  PMD  PTE  Offset

Shared Leaf-level Page Table Pages
Putting It All Together

Process A

<table>
<thead>
<tr>
<th>PGD</th>
<th>PUD</th>
<th>PMD</th>
<th>PTE</th>
<th>Offset</th>
</tr>
</thead>
</table>

Write

DRAM

Shared Leaf-level Page Table Pages

update PTE

Mapped File

FlashMap

Flash

FlashMap
Putting It All Together

Process A

- PGD
- PUD
- PMD
- PTE
- Offset

Write

DRAM

Shared Leaf-level Page Table Pages

Mapped File

FlashMap

Flash
Putting It All Together

Process A

PGD | PUD | PMD | PTE | Offset

DRAM

GC

FlashMap

Shared Leaf-level Page Table Pages

Mapped File

Flash
FlashMap: Implementation in Real System

Linux Memory Manager

SSD Emulator

Real SSD  RAMDisk

File System (EXT4)

Mapped File

File Index
FlashMap: Implementation in Real System

Linux Memory Manager

Mapped File

File System (EXT4)

File Index

SSD Emulator

Real SSD

RAMDisk
FlashMap: Implementation in Real System

Linux Memory Manager

File System (EXT4)

Mapped File

File Index

SSD Emulator

Real SSD

RAMDisk
FlashMap: Implementation in Real System

Linux Memory Manager

File Index

Mapped File

File System (EXT4)

SSD Emulator

Real SSD

RAMDisk
Experimental Setup

Intel Xeon processors + 64 GB DRAM + 2 TB SSD

<table>
<thead>
<tr>
<th>Baseline</th>
<th>unmodified Linux: mmap + EXT4 + FTL with page-level mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTL+FS★</td>
<td>mmap + combined FTL &amp; file system</td>
</tr>
<tr>
<td>FlashMap</td>
<td>unified address translation</td>
</tr>
</tbody>
</table>

★ similar to Nameless Writes [Zhang et al., FAST’12] and DFS [Josephson et al., FAST’10]
Real Application Workloads

<table>
<thead>
<tr>
<th>NoSQL Store</th>
<th>.redis + YCSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Database</td>
<td>Shore-MT + TPCC, TPCB, TATP</td>
</tr>
<tr>
<td>Graph Analytics</td>
<td>GraphLab + PageRank</td>
</tr>
</tbody>
</table>
Metadata Size for 2 TB SSD

- Baseline
- FTL+FS
- FlashMap

Metadata Size (GB)

50%
Benefits from Reduced Mapping Overhead

FlashMap: 1.7x performance improvement over FTL+FS
Benefits from Reduced Mapping Overhead

FlashMap: 1.7x performance improvement over FTL+FS
Benefits from Reduced Mapping Overhead

Reducing the mapping overhead improves the DRAM caching efficiency.
Latency Reduction

 Benefit (up to 53% latency reduction)
mainly comes from the combination of page table and file system
Benefits from Reduced Latency

Throughput (K TPS) vs. Device Latency (us)

- Baseline
- FTL+FS
- FlashMap

FlashMap: 1.8x more TPS than baseline and FTL+FS
Conclusion

1. Reduced Storage
   3.3x performance improvement for data-intensive applications

2. Reduced Latency
   53% latency reduction for high-end SSDs, 1.8x more TPS for latency-sensitive applications, e.g., database systems
Thanks!

Jian Huang
jian.huang@gatech.edu

Anirudh Badam†  Moinuddin K. Qureshi  Karsten Schwan

Q&A