Extending SSD Lifetimes with Disk-Based Write Caches

Gokul Soundararajan
University of Toronto

Vijayan Prabhakaran
Mahesh Balakrishnan
Ted Wobber
Microsoft Research
Solid-State Devices (SSDs)
Solid-State Devices (SSDs)

- Replacing Hard Disk Drives (HDDs)
  - Fast I/O reads
  - Consume low power, no moving parts, and more reliable
Solid-State Devices (SSDs)

- **Replacing Hard Disk Drives (HDDs)**
  - Fast I/O reads
  - Consume low power, no moving parts, and more reliable

- **Limited write cycles**
  - Need to **erase block before re-write**
  - High end SLC provide 100,000 erase cycles
  - Mainstream MLC provides 5,000-10,000 erase cycles
Solid-State Devices (SSDs)

- **Replacing Hard Disk Drives (HDDs)**
  - Fast I/O reads
  - Consume low power, no moving parts, and more reliable

- **Limited write cycles**
  - Need to erase block before re-write
  - High end SLC provide 100,000 erase cycles
  - Mainstream MLC provides 5,000-10,000 erase cycles

- **Starting to be used in laptops/desktops**
  - Contain write intensive workloads
SSD Write-lifetime
SSD Write-lifetime

Maximum amount of data that can be written to the SSD

Write Lifetime (TB)

300
225
150
75
0

Ideal     Micron C200     Intel X-25M
SSD Write-lifetime

Write Lifetime (TB)

- Ideal: 60GBx5000 = 300TB
- Micron C200
- Intel X-25M
SSD Write-lifetime

<table>
<thead>
<tr>
<th>Model</th>
<th>Write Lifetime (TB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>300 TB</td>
</tr>
<tr>
<td>Micron C200</td>
<td>42 TB</td>
</tr>
<tr>
<td>Intel X-25M</td>
<td>42 TB (60GB drive)</td>
</tr>
</tbody>
</table>
SSD Write-lifetime

Write Lifetime (TB)

Ideal  Micron C200  Intel X-25M

37TB (80GB drive)

20 GB/day*5 yrs
Griffin Hybrid Device

Hybrid Device Controller

Griffin

Writes

Reads

Write Cache

SSD

Thursday, February 25, 2010
Griffin Hybrid Device

Hybrid Device Controller

Griffin

MIGRATE

Write Cache

SSD

Thursday, February 25, 2010
Griffin Hybrid Device

Hybrid Device Controller

Options

Write Cache

MIGRATE

SSD

Griffin

Writes

Reads

Reads
Griffin Hybrid Device

Options
RAM

Hybrid Device Controller

Griffin

Writes
Reads

MIGRATE

Write Cache

Reads

SSD

Thursday, February 25, 2010
Griffin Hybrid Device

Options
RAM
NVRAM
SSD

Writes

Reads

Hybrid Device Controller

Griffin

Write Cache

MIGRATE

SSD
Griffin Hybrid Device

Options
RAM
NVRAM
SSD
Hard Disk

Write Cache

Hybrid Device
Controller

Griffin

MIGRATE

Reads

Writes

Reads

SSD
Griffin Hybrid Device

- **Write Cache**
- **Hybrid Device Controller**
- **Griffin**
- **MIGRATE**
- **SSD**

**Why Hard Disk?**
- Durable
- Capacity
- Cheap
- Sequential I/O
Contributions
Contributions

- Characterized I/O patterns
  - Found desktop/server traces contains many overwrites
  - Caching overwrites reduces writes to SSD by 52% ideally
Contributions

- **Characterized I/O patterns**
  - Found desktop/server traces contains many overwrites
  - Caching overwrites reduces writes to SSD by 52% ideally

- **Designed Griffin hybrid disk**
  - Uses a **disk-based write cache**
  - Cache data on hard disk, periodically move it back to SSD
  - Shows a 2x lifetime improvement ( < 5% HDD reads)
Outline

- Motivating Workload Characteristics
  - Key features
- Disk-based Write Caching
- Experimental Evaluation
- Conclusions
I/O Workload Characterization

- Examined various I/O traces
  - Desktops (Internal Microsoft traces)
  - Servers (Narayanan et. al from MSR Cambridge)
    - Use only the write-intensive traces
  - Linux (Bhadkamkar et. al. from FIU)
    - Contains desktop, SVN, and web server

- Trace descriptions
  - Block I/Os collected below the filesystem buffer cache
  - Multi-hour traces of 5 hours to 176 hours
  - Between 209K to 543M I/O events per trace
Key Metrics
Key Metrics

- Overwrites
  - Consecutive writes to a block without an intervening read
Key Metrics

- Overwrites
  - Consecutive writes to a block without an intervening read

W: 100  W: 100  W: 100  W: 100  R: 100  W: 200  R: 200  W: 200  R: 200
Key Metrics

» Overwrites

- Consecutive writes to a block without an intervening read

W: 100  W: 100  W: 100  W: 100  R: 100  W: 200  R: 200  W: 200  R: 200
Key Metrics

- **Overwrites**
  - *Consecutive writes to a block without an intervening read*

```
W: 100  W: 100  W: 100  R: 100  W: 200  R: 200  W: 200  R: 200
```
Key Metrics

- **Overwrites**
  - *Consecutive writes to a block without an intervening read*

- **Write-after-Write (WAW) Times**
  - *Time between two consecutive writes to the same block*
Key Metrics

- **Overwrites**
  - *Consecutive writes to a block without an intervening read*

- **Write-after-Write (WAW) Times**
  - *Time between two consecutive writes to the same block*

- **Read-after-Write (RAW) Times**
  - *Time between a write and a subsequent read to the same block*
Why Overwrites? Lifetime of a Block
Why Overwrites? Lifetime of a Block

R: 100

Time

∞
Why Overwrites? Lifetime of a Block

Time

Writeback to disk

R: 100  W: 100  W: 100  W: 100  W: 100

∞
Why Overwrites? Lifetime of a Block

Evicted from FS buffer cache

Time

R: 100 W: 100 W: 100 W: 100 W: 100

R: 100
Why Overwrites? Lifetime of a Block

Evicted from FS buffer cache

Coalesced write to SSD 75% write savings
# Top Overwritten Files in Desktops

<table>
<thead>
<tr>
<th>Rank</th>
<th>Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C:\Outlook.ost</td>
</tr>
<tr>
<td>2</td>
<td>C:...**Search**...\Windows.edb</td>
</tr>
<tr>
<td>3</td>
<td>C:$Bitmap</td>
</tr>
<tr>
<td>4</td>
<td>C:\Windows\Prefetch\Layout.ini</td>
</tr>
<tr>
<td>5</td>
<td>C:\Users&lt;name&gt;\NTUSER.DAT</td>
</tr>
<tr>
<td>6</td>
<td>C:$Mft</td>
</tr>
</tbody>
</table>
Ideal Write Savings: Remove Overwrites
Ideal Write Savings: Remove Overwrites

- Desksops
- Servers
- Linux

Write Savings (%)

- Desktops: 25-50%
- Servers: 75-100%
- Linux: 75-100%
Ideal Write Savings: Remove Overwrites

<table>
<thead>
<tr>
<th>Write Savings (%)</th>
<th>Desktops</th>
<th>Servers</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td>14%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thursday, February 25, 2010
Ideal Write Savings: Remove Overwrites

52% Average Write Savings

- Desktops: 14%
- Servers: 94%
- Linux: 14%

Thursday, February 25, 2010
WAW/RAW Time Intervals

Histogram Buckets (seconds)
WAW/RAW Time Intervals

- WAW
- RAW

Cumulative Time Interval (%)

Histogram Buckets (seconds)
WAW/RAW Time Intervals

Overwrites happen quickly

Cumulative Time Interval (%) vs. Histogram Buckets (seconds)
WAW/RAW Time Intervals

Cumulative Time Interval (%)

Histogram Buckets (seconds)

WAW

RAW

Written data is read late
Summary of Observations

- Large fraction of overwrites
  - Potential of 36% to 64% reduction for desktops
  - As much as 94% in server workloads (web server)
  - Linux: 62% in desktop and 81% in servers (web server)
Summary of Observations

› Large fraction of overwrites
  - *Potential of 36% to 64% reduction for desktops*
  - *As much as 94% in server workloads (web server)*
  - *Linux: 62% in desktop and 81% in servers (web server)*

› Overwrites happen quickly, reads after a long interval
  - *Over 50% of overwrites within 30 seconds*
  - *Only 21% of written data read within 15 minutes*
Outline

- Motivating Workload Characteristics
- Disk-based Write Caching
  - Basic algorithm
  - Performance tradeoffs
- Experimental Results
- Conclusions
Griffin: Handling Writes

LOG: 1

Griffin

HDD

SSD
Griffin: Handling Writes

HDD

SSD

LOG: 1

W: 100

Griffin

Thursday, February 25, 2010
Griffin: Handling Writes

LOG: 1
W: 100

Send to Hard Disk

Griffin

HDD

SSD
Griffin: Handling Writes

- **LOG: 2**
- **Blockmap**
  - Block: 100
  - Log: 1

**Griffin**

**Memory**

**HDD**

**SSD**
Griffin: Handling Writes

```
Griffin

LOG: 3

Blockmap

<table>
<thead>
<tr>
<th>Block</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
</tr>
</tbody>
</table>

Memory

HDD

SSD
```

Thursday, February 25, 2010
Griffin: Handling Reads

Blockmap

<table>
<thead>
<tr>
<th>Block</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
</tr>
</tbody>
</table>

HDD

SSD
Griffin: Handling Reads

**Log:**
- **Blockmap**
  - Block | Log
  - 100  | 1
  - 200  | 2

**Storage:**
- **HDD**
- **SSD**
Griffin: Handling Reads

**Check Blockmap**

<table>
<thead>
<tr>
<th>Block</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
</tr>
</tbody>
</table>

**Griffin**

**HDD**

**SSD**

Thursday, February 25, 2010
Griffin: Handling Reads

**Blockmap**

<table>
<thead>
<tr>
<th>Block</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
</tr>
</tbody>
</table>

HDD

SSD

**Griffin**

LOG: 3

R: 300

Thursday, February 25, 2010
Griffin: Handling Reads

LOG: 3

Blockmap

<table>
<thead>
<tr>
<th>Block</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
</tr>
</tbody>
</table>

HDD

SSD

Memory

Thursday, February 25, 2010
Griffin: Data Migration

**Griffin**

**Blockmap**

<table>
<thead>
<tr>
<th>Block</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
</tr>
</tbody>
</table>

**Memory**

**HDD**

**SSD**
Griffin: Data Migration

Migrate data to SSD

<table>
<thead>
<tr>
<th>Block</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
</tr>
</tbody>
</table>

HDD

SSD

Griffin

Memory
Griffin: Data Migration

- HDD
- SSD
- Blockmap
  - Block
  - Log

Thursday, February 25, 2010
Tradeoff: Write Savings vs. Read Penalty
Tradeoff: Write Savings vs. Read Penalty

How long to cache?

What to cache?

Long Time

All Writes
Tradeoff: Write Savings vs. Read Penalty

- What to cache?
- How long to cache?
- Long Time
- All Writes
- Write Savings

Thursday, February 25, 2010
Tradeoff: Write Savings vs. Read Penalty

- How long to cache?
- What to cache?

- Read Penalty
- Write Savings

- Long Time
- All Writes
Tradeoff: Write Savings vs. Read Penalty

What to cache?

Long Time

All Writes

What to cache?
What to cache?: Overwrite Distribution

Percentage of Overwrites vs. Percentage of Written Blocks
What to cache?: Overwrite Distribution

1% of the blocks receive 70% of overwrites.
Selective Write-Caching

LOG: 1

Griffin

HDD

SSD
Selective Write-Caching

**Blockmap**

<table>
<thead>
<tr>
<th>Block</th>
<th>Overwritten?</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

**Diagram**

- **Griffin**
  - **LOG: 1**
  - **W: 100**

- **HDD**
  - Array of blocks

- **SSD**
  - Array of blocks

- **Arrows**
  - From Griffin to HDD
  - From Griffin to SSD
Selective Write-Caching

<table>
<thead>
<tr>
<th>Blockmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block</td>
</tr>
<tr>
<td>100</td>
</tr>
</tbody>
</table>

Griffin

HDD

SSD

Is heavily overwritten?
Selective Write-Caching

Blockmap

<table>
<thead>
<tr>
<th>Block</th>
<th>Overwritten?</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>✔</td>
<td>1</td>
</tr>
</tbody>
</table>

Griffin

LOG: 2

HDD

SSD
Selective Write-Caching

**Blockmap**

<table>
<thead>
<tr>
<th>Block</th>
<th>Overwritten?</th>
<th>Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>✔</td>
<td>1</td>
</tr>
</tbody>
</table>

Griffin

HDD

SSD
Tradeoff: Write Savings vs. Read Penalty

- What to cache?
- How long to cache?

- Long Time
- All Writes

- Read Penalty
- Write Savings

Thursday, February 25, 2010
Tradeoff: Write Savings vs. Read Penalty

How long to cache?

Long Time

What to cache?

All Writes

Read Penalty

Write Savings
How long to cache?: Time Intervals

Cumulative Time Interval (%) vs. Histogram Buckets (seconds)

- **WAW**
- **RAW**

Overwrites happen quickly

Written data is read late

Thursday, February 25, 2010
How long to cache?: Time Intervals

- **WAW**
- **RAW**

- **Overwrites happen quickly**
- **Written data is read late**

Cumulative Time Interval (%) vs Histogram Buckets (seconds)
How long to cache?: Time Intervals

Cumulative Time Interval (%)

Histogram Buckets (seconds)

WAW

RAW

Overwrites happen quickly

Written data is read late

Thursday, February 25, 2010
How long to cache?: Migration Triggers
How long to cache?: Migration Triggers

- **Timeout trigger**
  - **Idea:** Fires after a certain time elapses
  - Could have high read penalty
How long to cache?: Migration Triggers

- **Timeout trigger**
  - *Idea:* Fires after a certain time elapses
  - Could have high read penalty

- **Read-threshold trigger**
  - *Idea:* Fires if HDD reads exceeds threshold since last migration
  - Bounds HDD read fraction
  - Could have no migrations for a long time
How long to cache?: Migration Triggers

- **Timeout trigger**
  - *Idea: Fires after a certain time elapses*
  - Could have high read penalty

- **Read-threshold trigger**
  - *Idea: Fires if HDD reads exceeds threshold since last migration*
  - Bounds HDD read fraction
  - Could have no migrations for a long time

- **Hybrid trigger**
  - Every 15 mins or read penalty > 5%
Outline

- Motivating Workload Characteristics
- Disk-based Write Caching
- Experimental Evaluation
  - Caching Policies: *What to cache?*
  - Migration Policies: *How long to cache?*
  - Performance: *Lifetime and Latency*
- Conclusions
What to Cache?: Write Savings

(SSD write savings (percentage of total writes))

Traces

D-1A  D-1B  D-1C  D-1D  D-2A  D-2B  D-2C  D-2D  D-3A  D-3B  D-3C  D-3D  S-EXCH  S-PRXY1  S-SRC10  S-SRC22  S-STG1  S-WDEV2

Full Caching
Selective Caching
What to Cache?: Write Savings

<table>
<thead>
<tr>
<th>Traces</th>
<th>Full Caching</th>
<th>Selective Caching</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-1A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-1B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-1C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-2A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-2B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-1D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-2C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-2D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-3A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-3B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-3C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-3D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-EXCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-PRXY1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-SRC10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-SRC22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-STG1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-WDEV2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SSD write savings (percentage of total writes)

Migrated at the end of trace

Thursday, February 25, 2010
What to Cache?: Write Savings

SSD write savings (percentage of total writes)

Migrated at the end of trace

Full Caching
Selective Caching

Traces
D-1A  D-1B  D-1C  D-1D  D-2A  D-2B  D-2C  D-2D  D-3A  D-3B  D-3C  D-3D  S-EXCH  S-PRXY1  S-SRC10  S-SRC22  S-STG1  S-WDEV2

25%

Thursday, February 25, 2010
What to Cache?: Write Savings

Migrated at the end of trace

SSD write savings
(percentage of total writes)

Traces

D-1A  D-1B  D-1C  D-1D  D-2A  D-2B  D-2C  D-2D  D-3A  D-3B  D-3C  D-3D  S-EXCH  S-PRXY1  S-SRC10  S-SRC22  S-STG1  S-WDEV2

Full Caching
Selective Caching

25%
51%

Thursday, February 25, 2010
What to Cache?: Read Penalty

HDD read penalty (percentage of total reads)

Traces
D-1A, D-1B, D-1C, D-1D, D-2A, D-2B, D-2C, D-2D, D-3A, D-3B, D-3C, D-3D, S-EXCH, S-PRXY1, S-SRC10, S-SRC22, S-STG1, S-WDEV2

Full Caching
Selective Caching
What to Cache?: Read Penalty

HDD read penalty (percentage of total reads)

Traces

D-1A, D-1B, D-1C, D-1D, D-2A, D-2B, D-2C, D-2D, D-3A, D-3B, D-3C, D-3D, S-EXCH, S-PRXY1, S-SRC10, S-SRC22, S-STG1, S-WDEV2

Legend:
- Full Caching
- Selective Caching

Migrated at the end of trace
What to Cache?: Read Penalty

Migrated at the end of trace

HDD read penalty (percentage of total reads)

Traces

D-1A  D-1B  D-1C  D-1D  D-2A  D-2B  D-2C  D-2D  D-3A  D-3B  D-3C  D-3D  S-EXCH  S-PRXY1  S-SRC10  S-SRC22  S-STG1  S-WDEV2

Full Caching
Selective Caching

19%
How Long to Cache?: Write Savings

SSD write savings (percentage of total writes)

- Blue: Migration timeout 3600 s
- Yellow: Migration timeout 1800 s
- Black: Migration timeout 900 s

Traces:
- D-1A
- D-1B
- D-1C
- D-1D
- D-2A
- D-2B
- D-2C
- D-2D
- D-3A
- D-3B
- D-3C
- D-3D
- S-EXCH
- S-PRXY1
- S-SRC10
- S-SRC22
- S-STG1
- S-WDEV2

Thursday, February 25, 2010
How Long to Cache?: Write Savings

15min gives large savings

SSD write savings (percentage of total writes)

Traces

D-1A  D-1B  D-1C  D-1D  D-2A  D-2B  D-2C  D-2D  D-3A  D-3B  D-3C  D-3D  S-EXCH  S-PRXY1  S-SRC10  S-SRC22  S-STG1  S-WDEV2

Migration timeout 3600 s
Migration timeout 1800 s
Migration timeout 900 s

Thursday, February 25, 2010
How Long to Cache?: Write Savings

15min gives large savings

1hr less additional benefit

SSD write savings (percentage of total writes)

Traces

Thursday, February 25, 2010
How Long to Cache?: Read Penalty

HDD read penalty (percentage of total reads)

- Migration timeout 3600 s
- Migration timeout 1800 s
- Migration timeout 900 s

Traces

- D-1A
- D-1B
- D-1C
- D-1D
- D-2A
- D-2B
- D-2C
- D-3A
- D-3B
- D-3C
- D-3D
- S-EXCH
- S-PRXY1
- S-SRC10
- S-SRC22
- S-STG1
- S-WDEV2

Thursday, February 25, 2010
How Long to Cache?: Read Penalty

HDD read penalty (percentage of total reads)

- Migration timeout 3600 s
- Migration timeout 1800 s
- Migration timeout 900 s

Traces:
- D-1A
- D-1B
- D-1C
- D-1D
- D-2A
- D-2B
- D-2C
- D-2D
- D-3A
- D-3B
- D-3C
- D-3D
- S-EXCH
- S-PRXY1
- S-SRC10
- S-SRC22
- S-STG1
- S-WDEV2

Migration timeout 3600 s: 7%
Performance Summary: Erasure Savings

SSD block erase savings (%)

- Savings with a simple FTL
- Savings with an ideal FTL

Traces

- D-1A
- D-1B
- D-1C
- D-1D
- D-2A
- D-2B
- D-2C
- D-2D
- D-3A
- D-3B
- D-3C
- D-3D
- S-EXCH
- S-PRXY1
- S-SRC10
- S-SRC22
- S-STG1
- S-WDEV2

Thursday, February 25, 2010
Performance Summary: Erasure Savings

Direct block-level mapping

- Savings with a simple FTL
- Savings with an ideal FTL

SSD block erasure savings (%)

Traces

D-1A D-1B D-1C D-1D D-2A D-2B D-2C D-2D D-3A D-3B D-3C D-3D S-EXCH S-PRXY1 S-SRC10 S-SRC22 S-STG1 S-WDEV2
Performance Summary: Erasure Savings

SSD block erasure savings (%)

- Savings with a simple FTL
- Savings with an ideal FTL

Traces

- D-1A
- D-1B
- D-1C
- D-1D
- D-2A
- D-2B
- D-2C
- D-2D
- D-3A
- D-3B
- D-3C
- D-3D
- S-EXCH
- S-PRXY1
- S-SRC10
- S-SRC22
- S-STG1
- S-WDEV2

Direct block-level mapping

Log structured page-level mapping
Performance Summary: Erasure Savings

- **Savings with a simple FTL**
- **Savings with an ideal FTL**

Traces:
- D-1A
- D-1B
- D-1C
- D-1D
- D-2A
- D-2B
- D-2C
- D-2D
- D-3A
- D-3B
- D-3C
- D-3D
- S-EXCH
- S-PRXY1
- S-SRC10
- S-SRC22
- S-STG1
- S-WDEV2

Erasure savings (%)

- > 2x
Performance Summary: I/O Latency

- **Workload T1** and **Workload T2**
- **HDD 7.2K**
- **HDD 10K**
- **MLC**
- **SLC**

Thursday, February 25, 2010
Performance Summary: I/O Latency

Relative I/O latency

- HDD 7.2K
- HDD 10K
- MLC
- SLC

Most I/O intensive segments (2hr) - desktop

Thursday, February 25, 2010
Performance Summary: I/O Latency

Relative I/O latency

- HDD 7.2K
- HDD 10K
- MLC
- SLC

Workload T1 vs. Workload T2

- Total Read
- Total Write
- Read T1
- Write T1
- Read T2
- Write T2

Relative I/O latency: 0.44
Performance Summary: I/O Latency

- Relative I/O latency for HDD 7.2K, HDD 10K, MLC, and SLC.
- Workload T1 and T2 are compared.
- Data points include:
  - HDD 7.2K Total Read: 0.44
  - HDD 7.2K Total Write: 0.15
  - HDD 10K Total Read: 0.44
  - HDD 10K Total Write: 0.15

Thursday, February 25, 2010
More in the Paper
More in the Paper

- Evaluation of other policies
  - Different migration triggers
More in the Paper

- Evaluation of other policies
  - Different migration triggers

- Failure handling
  - Have state on two devices
  - Recovery more intricate
    - Leverage existing journalling and recovery techniques
  - More details in the paper
Conclusions

- **SSDs starting to appear in desktops/laptops**
  - Contain more write-intensive workloads
  - Lifetimes limited due to limit of block erasures

- **Built Griffin hybrid disk**
  - Uses hard drive as a write-cache

- **Reduces writes while maintaining performance**
  - Reduces writes *by 52%* (< 5% HDD reads)
  - Improves lifetime by *factor of 2*
  - Reduces average I/O latency *by 56%*
Thank you

Griffin picture from:
http://www.e-wollmann.com/griffin.jpg
Related Work

- **SSD Lifetimes**
  - Shown to degrade over time
    - Grupp et. al [ISM’09]
    - Desyoners [HotStorage’09], Boboila et. al [FAST’10]

- **Hybrid drives**
  - Used SSD as cache for hard drive
    - Kotsidas et. al [VLDB’08], Combo drive [WISH’09]
  - Windows ReadyBoost
    - *caches data normally paged out the HDD*
  - Intel Turbo Memory and Sun ZFS+Flash
Improved Sequentiality

Sequential writes (percentage of total writes)