Apps Can Quickly Destroy Your Mobile's Flash: Why They Don't, and How to Keep It That Way

Tao Zhang\textsuperscript{1}, \textbf{Aviad Zuck}\textsuperscript{2}, Donald E. Porter\textsuperscript{1}, Dan Tsafrir\textsuperscript{2,3}
We Expect Improvements Over Time

Samsung S1 (2010)  

a decade later

Samsung S10 (2019)
Flash Evolution

• **Higher density (lower cost)**
  - Smaller cells (1x nm)
  - More bits per cell

• **Easier to wear out**
  - QLC flash can’t reliably store data after < 1K write cycles

• **Poorer performance**
Problem #1: Many People Think SSD Endurance is a Non-issue

misconception also extends to operating systems designers

**don't need to be**

While horror stories prevail regarding SSD reliability, recent tests carried out suggest that consumer solid state drives (SSDs) can be subjected to high usage levels before they experience failure.
Problem #2: Compact SSD (with Compromises)

- Smaller form factor
- More power efficient
- Cost less
- High-throughput interfaces

- Lower capacity
- Limited hardware
- Less sophisticated firmware
- No replacement!
**Write Bandwidth/Capacity Ratio**

Smartphones skew toward dangerous bandwidth/capacity ratio

- Easy to issue lifetime’s worth of writes
Problem #3: False Sense of Security

- Tighter security models
Problem #3: False Sense of Security

- Misplaced trust in app marketplaces
  - “In September alone, researchers uncovered 172 infected apps with over 335 million installs on the Play Store”
  - thenextweb.com, Oct 1 2019

- Users carelessly grant permissions
• Conventional wisdom: SSD wear-out not a problem
• Our analysis: There is cause for concern, especially for mobile storage:
  1. Dangerous bandwidth/capacity skew
  2. Less sophisticated devices
  3. Users perceive mobile phones as safer (strict permissions, app stores)

• How bad could it be?
  – Let’s try attacking mobile devices and measure lifespan!
Threat Model

- Mobile storage device (eMMC/UFS)
- Long-term warranty (e.g., 2Y)
- Supports synchronous IO
- Code snippet can access storage space by default
  - Granted by default to all apps
  - E.g., app requires no special privileges
Wear-out Attack

- Prototype Android app with less than 1K lines of code
  - No special permission needed
- Stealthily rewrite small files in app’s storage space
- Current OSs provide no protection/warning

Run as background service

Only run on charging status

Pause workload on screen lit
How to Evaluate Wear-out Level

• Built-in Wear-out Indicators
  – eMMC [JESD84-B51] Extended CSD register
  – UFS [JESD220C] Device Health Descriptor
  – Value from 1 to 11

<table>
<thead>
<tr>
<th>Value</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Life Consumed</td>
<td>0% ~ 10%</td>
<td>10% ~ 20%</td>
<td>20% ~ 30%</td>
<td>30% ~ 40%</td>
<td>40% ~ 50%</td>
<td>50% ~ 60%</td>
<td>60% ~ 70%</td>
<td>70% ~ 80%</td>
<td>80% ~ 90%</td>
<td>90% ~ 100%</td>
<td>Worn out</td>
</tr>
</tbody>
</table>

- Consumed Life
Phone Wear-out Experiment Results

BLU 512MB 4GB
< 14 days

Moto E 8GB
6 days ~2 weeks

Samsung S6 (32GB)
8 days

Samsung S9 (64GB)
22 days

Phones can be worn out in weeks!
Buggy Apps Can Also Kill SSDs

Spotify has been quietly killing your SSD’s life for months

BGR Chris Smith
BGR News November 13, 2016
• Mobile flash storage can be worn out quickly
• Mobile flash storage can be worn-out quickly

Why my phone is not dead (yet)?
Mobile App I/O Characterization

- **Platform:** Samsung S6 32GB
  - ~88 TiB estimated lifetime write
  - 2Y warranty

- **1st characterization of mobile app I/O behavior:**
  - Top 150 free apps from Google Play Store*
  - 27 preloaded apps (camera, etc.)
  - I/O-intensive workloads (FTP server, file copies, backup/restore)

* 23 apps excluded due to various reasons, details in paper
Initial conclusions

• Most apps don’t consume dangerous levels of write bandwidth
  – Most apps are not used most of the time

• Minority of apps are write-intensive
  – Let’s look more closely at these “troublemakers”
Write-heavy Apps/Workloads

- Apps issue bursts of I/O
Can apps prematurely wear-out your phone?

<table>
<thead>
<tr>
<th>app</th>
<th>avg. throughput (MiB/s)</th>
<th>required daily usage (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB copy</td>
<td>29.74</td>
<td>1.18</td>
</tr>
<tr>
<td>FTP</td>
<td>6.39</td>
<td>5.50</td>
</tr>
<tr>
<td>Camera</td>
<td>4.26</td>
<td>8.24</td>
</tr>
<tr>
<td>Backup (local)</td>
<td>2.3</td>
<td>15.25</td>
</tr>
<tr>
<td>Restore (local)</td>
<td>23.29</td>
<td>1.51</td>
</tr>
<tr>
<td>Daily Horoscope</td>
<td>4.98</td>
<td>7.05</td>
</tr>
<tr>
<td>Final Fantasy</td>
<td>3.84</td>
<td>9.15</td>
</tr>
</tbody>
</table>

- Reasonable app usage won’t shorten device lifetime
  - Most write-heavy usage scenarios not long-term/frequently used
- Extreme use cases CAN prematurely wear-out phone (but not likely)
Most apps cause little to no background I/O activities
Interim Summary: device killers

- Buggy apps (unintentionally)
- Wear-out attack (intentionally)
- App users (unintentionally)
Impact Beyond Phones

Same storage devices used in TVs, medical devices, wearables, IoT, GPS, smart home devices, cars...
OS-level Wear Management

- Monitor and measure app-specific I/O behavior
  - Extend diskstats accordingly

- Let the user choose whether app behavior is normal!

- But help users make informed decision
Write Quota Regulation

- **Upper limit** (per-second) on I/O writes
- Appropriate 50% of lifetime writes as **slack** (daily)
  - Accommodates write bursts of benign apps
  - Stricter quota & threshold on background apps (i.e., hourly)
- More details in the paper
Evaluation (Write-intensive Apps)

- Video shooting with camera (foreground)
- Bursts are permitted
- ~1.2 hours daily usage without intervention

- Google Hangouts receiving messages every 5s (background)
- ~300 KiB/s background workload

Benign apps run with no/minimum disruption
Evaluation (Wear-out attack)

- Malicious wear-out attack in background
- \(~80\text{MiB/s}\) maximum throughput

Phone protection kicks in within 30s
Firmware can amplify write I/O

Effective wear management attributes app I/O to flash writes

Need to understand internal firmware behavior

80x

Done, But Not Over

storage media

indirection layer

interface (eMMD/UFS)

operating system (Android/iOS)

0 0

1 1

0 1

1 0

1 1

0 0
Conclusion

• Mobile flash storage is still in danger
  – App with no special perm can doom storage in days/weeks
• App I/O characterization
  – Mobile flash storage is safe with benign apps under reasonable usage
  – Extreme usage scenarios can still prematurely exhaust storage lifespan
• Prototype of flash wear management mechanism
  – Effectively identify & rate-limit malicious apps
  – Little to no disturbance on benign apps and user experience

• Flash storage lifespan as depletable resource needs to be managed
  – Embedded devices with flash storage (IoT devices, medical devices, etc.)

Aviad Zuck
aviadzuc@cs.technion.ac.il
Backup slides
Flash Internals

- **Floating gate (flash cell)**
  - Program (inject electrons)
  - Erase (eject electrons)
  - Electrons trapped in insulating oxide (worn out)
SLC ⇔ MLC ⇔ TLC: Evolution or Degeneration?

- Higher density (lower cost)
- Poorer performance
- Easier to wear-out
  - SLC: up to 100K P/E cycles
  - MLC: 3K ~ 10K P/E cycles
  - TLC: < 1000 P/E cycles
- “...global shipment share of client-grade SSDs using TLC Flash will exceed 75% by in 2017.”
  [DRAMeXchange]
eMMC Flash Chips Can Wear-out in Days

- 23 TiB total write, ~7 days at 40 MiB/s
- 8 TiB total write, ~6 days at 20 MiB/s