The SanDisk Brand

- A global leader in storage technology
- Most compelling value proposition
- Unmatched innovation and IP
- We are Data Champions
The Better Alignment of Managed Flash To System Behavior

Alex Lemberg
SW Manager
Agenda

- Flash Storage in Mobile & Embedded
  - Real Performance Requirements
  - The Gap Between Synthetic and User Activities
- Usage Case – Performance Peaks
- How to Handle Performance Peaks in Flash Management Architecture
- How it Affects the Endurance
- Driver Support

How to Get Better Performance?
Embedded Flash Memory is Everywhere
The “Real” Storage Performance Requirements

- What is the Most Important Performance Metric?
  - Synthetic Benchmarks
    - Sequential Write (MB/Sec)
    - Sequential Read (MB/Sec)
    - Random Write (IOPS)
    - Random Read (IOPS)
    - SQL Insert/Update/Delete
  - System Analysis
    - IO Latency
    - IO Flow
    - IO Stack Level
  - User Experience
    - App Launch Time
    - Boot Time
    - Multitasking
    - Etc.
Getting IO Metrics – Is the Key

24-96 Hours of intensive “managed” user activity

Wide platforms coverage

- User Experience
- Various Android & Linux Versions
- 7 OEMs
- 16GB-64GB RAM
- 1GB-4GB RAM
- High End/Mid Range
- EXT4/F2FS
- Regions

Statistics
System Analysis & Research
Simulations \ Testing
Enhanced Low Level Tracing

The Linux I/O Stack Diagram
Version 1.6, 2013-06-20
outlines the Linux I/O stack as of Kernel version 3.3

Applications (Processes)

VF

Page Cache

stackable

Block I/O Layer

I/O Scheduler
maps bio to requests

request-based
device mapper targets
dm/multipath

SCSI upper layer

sysfs

(transport attributes)

block based Fs
cdfs

Network Fs
cdrom

pseudo Fs

special

purpose Fs

BIOS (Block I/O)

Catch the Process and FS Info
Trace point

eMMC Device Driver

eMMC Host Controller
eMMC Device

The Linux I/O Stack Diagram Version 1.6, 2013-06-20
outlines the Linux I/O stack as of Kernel version 3.3

The Linux I/O Stack Diagram can be found in "The Linux I/O Stack Diagram.pdf" located in the "doc" directory: http://www.sandisk.com/LinuxStack.png
Enhanced Low Level Tracing Allows to Gather Per-Process Stat.

Operations Per Second - Mixed
Pack as Single Command

User Operation Context

- Gmail, WPS
- Instagram
- WFDirect + Gmail
- Multi-shot
- AppInstall + 4K Playback

Time (Seconds)

Night Suspend (~16 Hours)

*Peaks are shown in 1 Sec resolution

Day 2 - max Mixed logs (15sec)
Day 3 - max Mixed logs (15sec)
Agenda

- Flash Storage in Mobile & Embedded
  - Real Performance Requirements
- Usage Case
  - The Gap Between Synthetic and User Activities
  - Performance Peaks
- How to Handle Performance Peaks in Flash Management Architecture
- How it Affects the Endurance
- Driver Support
## Usage Case – Description in High Level

<table>
<thead>
<tr>
<th>High Level Category</th>
<th>Description</th>
<th>Duration (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social networking</td>
<td>Facebook, twitter, Instagram</td>
<td>67</td>
</tr>
<tr>
<td>Multimedia</td>
<td>Camera + advanced features (multi-shot, photo editing), Video recording, Video on-line, Streaming, etc</td>
<td>91</td>
</tr>
<tr>
<td>eMail</td>
<td>Gmail</td>
<td>17</td>
</tr>
<tr>
<td>Connectivity</td>
<td>USB transfers, Cloud download &amp; sync (Google Drive/drop box, Google Sync), wifi direct transfer</td>
<td>54</td>
</tr>
<tr>
<td>Audio</td>
<td>MP3 playback</td>
<td>27</td>
</tr>
<tr>
<td>Gaming</td>
<td>Minion Rush, Candy Crush</td>
<td>38</td>
</tr>
<tr>
<td>Apps</td>
<td>Playstore installs &amp; search, Frequent Apps launch</td>
<td>47</td>
</tr>
<tr>
<td>Productivity</td>
<td>Google keep, WPS office</td>
<td>16</td>
</tr>
<tr>
<td>Location based services</td>
<td>Waze, Google Maps, Google Earth</td>
<td>13</td>
</tr>
<tr>
<td>Web browsing</td>
<td>Chrome, Firefox - Browser search, open URLs</td>
<td>7</td>
</tr>
</tbody>
</table>
### Example of 2016 Flagship Mobile Phone - 32GB Storage

<table>
<thead>
<tr>
<th></th>
<th>Day1</th>
<th>Day2</th>
<th>Day3</th>
<th>Daily average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Research Time (Hours)</strong></td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td><strong>User Active Time (Hours)</strong></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td><strong>User StandBy Time (Hours)</strong></td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td><strong>Storage Busy time (Min)</strong></td>
<td>~15</td>
<td>~15.65</td>
<td>~19</td>
<td>~16.55</td>
</tr>
<tr>
<td><strong>Write (GB)</strong></td>
<td>12.57</td>
<td>11.22</td>
<td>12.86</td>
<td>12.21</td>
</tr>
<tr>
<td><strong>Read (GB)</strong></td>
<td>66.49</td>
<td>69.63</td>
<td>87.64</td>
<td>74.58</td>
</tr>
<tr>
<td><strong>Discard (GB)</strong></td>
<td>6.39</td>
<td>5.94</td>
<td>6.74</td>
<td>6.35</td>
</tr>
<tr>
<td><strong>Flush#</strong></td>
<td>103,136</td>
<td>211,371</td>
<td>228,335</td>
<td>180,947</td>
</tr>
</tbody>
</table>

**Only 15 Minutes Storage Busy Time (Out of 24 Hours)**
Multi-Tasking Apps Are Not As Stressful As Synthetic Benchmark
<2.5K Read IOPs Peaks On Highly Intensive Daily Usage Case

Operations Per Second - Read
Pack as Single Command

2.5K

~8 hours of Intensive User Activity

Night Suspend (~16 Hours)

*Peaks are shown in 1 Sec resolution
The Gap Between Real Life And Synthetic Benchmarks

- eMMC Flagship Phone: ~4K Write IOPs, ~1.3K Workload Max IOPs
- UFS Flagship Phone: ~11K Write IOPs, ~1K Workload Max IOPs
Max Write MB/S (Peaks)

Measured on Leading Android-Based Smartphone
Usage Case: 3 Days of intensive user activity
Agenda

- Flash Storage in Mobile & Embedded
  - Real Performance Requirements
- Usage Case
  - The Gap Between Synthetic and User Activities
  - Performance Peaks
- How to Handle Performance Peaks in Flash Management Architecture
- How it Affects the Endurance
- Driver Support
Handling Performance Peaks
Intelligent Peak Performance On-Demand

Peak-Awareness
Embedded Storage

Typical
Embedded Storage

Conceptual Graph- not to scale
Peak-Awareness Architecture

Application’s Storage Requests

Data

Storage Device Solution

Controller

SLC Buffer

FTL engine

Burst Storage

Main Memory Area
Agenda

- Flash Storage in Mobile & Embedded
  - Real Performance Requirements
- Usage Case
  - The Gap Between Synthetic and User Activities
  - Performance Peaks
- How to Handle Performance Peaks in Flash Management Architecture
- How it Affects the Endurance
- Driver Support
SLC Buffering is Good for Endurance

Extend product lifetime

- Host data is ‘cached’ in the SLC area
- The flash-management optimize data folding from SLC to the Main Memory Area, and by that minimize the write amplification
- Frequently accessed data (‘Hot data’) remains in SLC area and being update there
- Only ‘Cold data’ is folded and stored in main area by minimizing main-area program erase cycles
Observe Hot-data in SLC and Route Mostly Cold-data to the Main Memory Area

- Host
  - Controller
    - SLC Buffer
      - Main Memory Area

- Host
  - Controller
    - Typical Embedded Storage
      - Main Memory Area
Agenda

- Flash Storage in Mobile & Embedded
  - Real Performance Requirements
- Usage Case
  - The Gap Between Synthetic and User Activities
  - Performance Peaks
- How to Handle Performance Peaks in Flash Management Architecture
- How it Affects the Endurance
- Driver Support
Is System Ready for Peak-Awareness Architecture?

- The Storage Driver’s Power Management Flow is Not Always Adjusted to Peak-Awareness Architecture:
  - No time for Background Garbage Collection
  - Enter Sleep Mode Immediately on Suspend
  - User May Suffer from Hiccups after Resume

- Discard is Not Enabled on Some Systems
  - No Free Blocks for Internal Garbage Collection
  - Higher Write Amplification
  - Higher Latency
How to Adjust The Peak Awareness Flash Management

- Enable BKOPS Support
  - Both "Manual" and "Auto"

- Give Enough Time for BKOPS before Runtime Suspend

- Enable PowerOffNotification to Allow Background GC
  - Set PowerOffNotification ON on card init

- Enable DISCARD
  - FS Mount Flag
  - Fstrim on Android
The Problem - Need to Give Enough Time for BKOPS Before Runtime Suspend

- Hardcoded 3 Seconds Delay Before Runtime Suspend
- Stop BKOPS on Runtime Suspend
Check the BKOPS Status On Runtime Suspend
Runtime Suspend Flow with Auto BKOPS

- System Idle Time
- Call .Suspend()
- Check BKOPS Level
- BKOPS Level >=1
  - Yes: Reschedule Suspend()
  - No: Check BKOPS Level
- Recommended Flow
  - Stop BKOPS
  - Flush Cache
  - Send Sleep CMD
  - Finish eMMC Suspend
- Suspend Flow...
Patchset for Handling BKOPS Status on Runtime Suspend

- **Patchset Submitted to add BKOPS Status support in eMMC PM**
  - [http://marc.info/?l=linux-mmc&m=147274208821646&w=4](http://marc.info/?l=linux-mmc&m=147274208821646&w=4)
  - [http://marc.info/?l=linux-mmc&m=147274015121024&w=4](http://marc.info/?l=linux-mmc&m=147274015121024&w=4)
  - [http://marc.info/?l=linux-mmc&m=147274104021291&w=4](http://marc.info/?l=linux-mmc&m=147274104021291&w=4)
  - [http://marc.info/?l=linux-mmc&m=147274215821667&w=4](http://marc.info/?l=linux-mmc&m=147274215821667&w=4)

- **Check BKOPS Status & Reschedule Suspend**
BKOPS in UFS

No Proper Handling of URGENT BKOPS in System Suspend

Allows URGENT BKOPS in Runtime Suspend

Disable BKOPS in System Suspend
Q&A

Email: alex.lemberg@sandisk.com
SanDisk®, a Western Digital brand, is expanding the possibilities of storage. Our products are in the world's leading-edge data centers, advanced mobile devices and laptops, and trusted by consumers worldwide.

© 2016 Western Digital Corporation or its affiliates. All rights reserved. SanDisk and iNAND are trademarks of Western Digital Corporation or its affiliates, registered in the United States and other countries. Android is a trademark of Google Inc. The microSD, microSDHC and microSDXC marks and logos are trademarks of SD-3C, LLC. Other brand names mentioned herein are for identification purposes only and may be the trademark(s) of their respective holder(s).

Western Digital Technologies, Inc. is the seller of record and licensee in the Americas of SanDisk® products.