Understanding Storage I/O Behaviors of Mobile Applications

Jace Courville  
jcourv@csc.lsu.edu

Feng Chen  
fchen@csc.lsu.edu

Louisiana State University  
Department of Computer Science and Engineering
The Rise of the Smartphone

- Smart device use has steadily increased since 2007
- Users are switching to these devices for daily computing tasks

Unique Behaviors of Mobile Applications

- Flash-based storage medium
  - High read performance, poor random write performance
- Latencies have a greater impact on device usability
  - Optimizations need to be latency-oriented
- Distinct software stack and distinct app characteristics
Applications are considered "users" with their own unique ID and set of permissions.

Applications run in a protected environment and privileged operations are encapsulated in a small set of API interfaces.

Libraries such as SQLite are heavily used in nearly all mobile apps.

Prior wisdom may not apply.
Key Questions

- How much do storage I/Os impact workload performance?
- Which type of storage I/Os contribute the most to latency?
- Are there any consistent trends in application performance?
  - Are behaviors different over different categories of workloads?
- What are the systems implications of storage I/O Latency?
Experimental Setup

- Google Nexus 5, 32 GB eMMC storage, 2 GB RAM
- AOSP Android 5.1 OS / Linux kernel 3.4.0
- blktrace / blkparse used to collect and interpret I/Os
  - Traces are stored on ramfs to eliminate blktrace overhead
  - Device restarted between each test to remove variance
  - blktrace started following end of interaction
- Metrics Gathered:
  - I/O Request Size, I/O Latency
  - Information Between Successive Flushes
  - Locality
  - Percentage of I/O time
### Workloads

- **13 Workloads from 5 categories representing real-world scenarios**

<table>
<thead>
<tr>
<th>Workload Name</th>
<th>Workload Type</th>
<th>R/W Ratio</th>
<th>Read-based</th>
<th>Write-based</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angry Birds</td>
<td>Game</td>
<td>2.03/1</td>
<td>X</td>
<td></td>
<td>Load the Angry Birds Application</td>
</tr>
<tr>
<td>App Removal</td>
<td>Device Utility</td>
<td>1.35/1</td>
<td>X</td>
<td></td>
<td>Uninstall an Application</td>
</tr>
<tr>
<td>Batch Uninstall</td>
<td>Device Utility</td>
<td>1/2.79</td>
<td></td>
<td>X</td>
<td>Uninstall several Applications through ADB at once</td>
</tr>
<tr>
<td>Camera</td>
<td>Multimedia</td>
<td>1/9.12</td>
<td></td>
<td>X</td>
<td>Default Camera used to take 3 pictures in sequence</td>
</tr>
<tr>
<td>Burst Mode Camera</td>
<td>Multimedia</td>
<td>1/204.1</td>
<td></td>
<td>X</td>
<td>Burst Mode Camera app used to take 100 photos in burst</td>
</tr>
<tr>
<td>Video Recording</td>
<td>Multimedia</td>
<td>1/4.25</td>
<td></td>
<td>X</td>
<td>Uses default Camera to record a 5 second video</td>
</tr>
<tr>
<td>Video Playback</td>
<td>Multimedia</td>
<td>1.81/1</td>
<td>X</td>
<td></td>
<td>Plays back the recorded 5 second video</td>
</tr>
<tr>
<td>Add Contact</td>
<td>Productivity</td>
<td>1/2.07</td>
<td></td>
<td>X</td>
<td>New contact is added through the Contacts app</td>
</tr>
<tr>
<td>Sync Dropbox</td>
<td>Network</td>
<td>1/5.63</td>
<td></td>
<td>X</td>
<td>Links an existing DropBox account to the device and syncs</td>
</tr>
<tr>
<td>Sync E-Mail</td>
<td>Network</td>
<td>1/4.25</td>
<td></td>
<td>X</td>
<td>Links an existing E-mail account to the device and syncs</td>
</tr>
<tr>
<td>Web Request</td>
<td>Network</td>
<td>1/1.47</td>
<td></td>
<td>X</td>
<td>Load the Facebook web site through the default browser</td>
</tr>
<tr>
<td>Route Plot</td>
<td>Network</td>
<td>1/2.54</td>
<td></td>
<td>X</td>
<td>Plots a GPS route using the Google Maps app</td>
</tr>
<tr>
<td>MP3 Stream</td>
<td>Network</td>
<td>1/41.8</td>
<td></td>
<td>X</td>
<td>Streams 15 seconds of a song in the Spotify app</td>
</tr>
</tbody>
</table>
Outline of Experiments

• Basic Observations
  • Two key factors: Request Size and Latency

• Flushing Behavior
  • Directly impacts I/O speed on NAND flash-based storage
  • Requests, Total Size, Time – Between Successive Flushes

• Access Locality
  • Has strong implications to cache efficiencies

• Total Storage I/O Latency impact
  • What percentage of runtime is storage I/O latency?
Basic Observations: Angry Birds

- Average case – Small request sizes of varying latency
- Read-Heavy Workload
  - Highest number of reads of any workload (567)
- 67.8% of all I/Os are smaller than 64 KB
- Writes longer than reads
Basic Observations: Camera – Normal Mode

- Highly write-heavy – 9.12 writes to 1 read (3rd highest)
  - 2nd highest total writes (2090)
- All writes are very small – 86.9% smaller than 16 KB
Basic Observations: Camera – Burst Mode

- Most write-heavy workload (204.1 writes to every 1 read)
  - Most writes of any workload at 2246
  - Fewest reads of any workload at 11
  - Writes are more variable in size
- Only 156 more reads than the Normal Mode workload
Basic Observations: Camera

- Both Camera modes experience variable latency for I/O writes
- Normal mode workload sees smaller writes, reads
- Burst workload sees very few reads, much larger writes
Basic Observations: Dropbox Sync

- Network-based workload – Majority small writes (80% < 8 KB)
- Compared to other workloads, reads are larger
- All writes have highly variable latencies

80% < 8 KB

80% < 2.13 ms
Application Developers may wish to ensure data persistence.

Android OS uses flush operation to send buffered data to storage.

Too much flushing can be a bad thing:
  - Can result in increased latency, therefore decreased performance.

Trend of excessive flushing is common.
Application Developers may wish to ensure data persistence.

Android OS uses flush operation to send buffered data to storage.

Too much flushing can be a bad thing:
- Can result in increased latency, therefore decreased performance.

Trend of excessive flushing is common.
• Application Developers may wish to ensure data persistence
• Android OS uses flush operation to send buffered data to storage
• Too much flushing can be a bad thing
  • Can result in increased latency, therefore decreased performance
• Trend of excessive flushing is common
90% of Flushes have < 16 I/O requests between successive flush operations.

- < 80 KB of Data and < .116 sec between flushes
- Very aggressive flushing – Extremely short iterations between flushes
90% of Flushes have < 18 I/O requests between successive flush operations.
- < 180 KB of Data and < 1.10 sec between flushes
- Data persistence is desired, so we see utilization of flush operations
Video Playback

- 90% of Flushes have < 49 I/O requests between successive flush operations.
  - < 4196 KB of Data and < 3.30 sec between flushes
- I/O writes not heavily used -- not as important to make any data persistent
A common trend – Very few blocks experience multiple accesses
- Camera workload had one block re-accessed 305 times
- Only top 300 most accessed blocks shown
  - MP3 Streaming has 658 accessed block – Camera has 3293
- Nearly all workloads saw reads as single access only
The impact of Storage I/O latency varies by workload. Camera is the most affected, at nearly 70%. Asynchronous Writes and Reads were the direct contributors. Metadata Reads and Asynchronous writes had little to no impact. Storage I/O Latency impact may not be user-perceivable.
System Implications

- I/O Writes are small with varying latency
  - Small writes range from 1 ms to 10 ms of latency
  - Category independent trend – Dropbox was 5th most affected workload
- Aggressive flushing is very common
  - Data safety is a concern for developers – results in aggressive flushing
  - Resulting small writes will magnify slow write performance of flash storage
- I/O Reads happen only once in nearly all workloads
  - Confirmed by reducing available RAM to 1 GB
  - Sufficient RAM availability has the most impact
- Synchronous writes are the most common – and the biggest issue
  - By numbers, Synchronous Writes and Reads were similar
  - Metadata Reads / Asynchronous writes uncommon with minimal impact
- Storage I/O impact varies by workload
  - Camera workload much larger – next most impacted was 20%
  - May not have as much as a user perceivable impact as previously thought
Conclusions

- There is a definite space for storage I/O optimization
- Small, synchronous writes are the biggest cause for I/O latency
- Reducing flushing will negate much of the latency caused by I/Os
- Impact of I/O latency is application and workload dependent
- Any solution must be customized to the individual workload
Thank You!

Jace Courville  
jcourv@csc.lsu.edu  

Feng Chen  
fchen@csc.lsu.edu