I/O Characteristics of Smartphone Applications and Their Implications for eMMC Design

The 11th IEEE International Symposium on Workload Characterization (IISWC), Atlanta, USA, 2015

Deng Zhou[1], Wen Pan[1], Wei Wang[1], and Tao Xie[2], San Diego State University, San Diego, CA 92182

Introduction

Smartphone applications' block-level I/O characteristics and their implications on eMMC design are still poorly understood. In this research, we collect and analyze block-level I/O traces from 18 common applications (e.g., Email and Twitter) on a Nexus 5 smartphone. We observe some I/O characteristics from which several implications for eMMC design are derived. Next, we conduct a case study to demonstrate how to apply the implications to optimize eMMC design. Inspired by two implications, we propose a hybrid-page-size (HPS) eMMC. Experimental results show that the HPS scheme can reduce mean response time by up to 86% while improving space utilization by up to 24.2%.

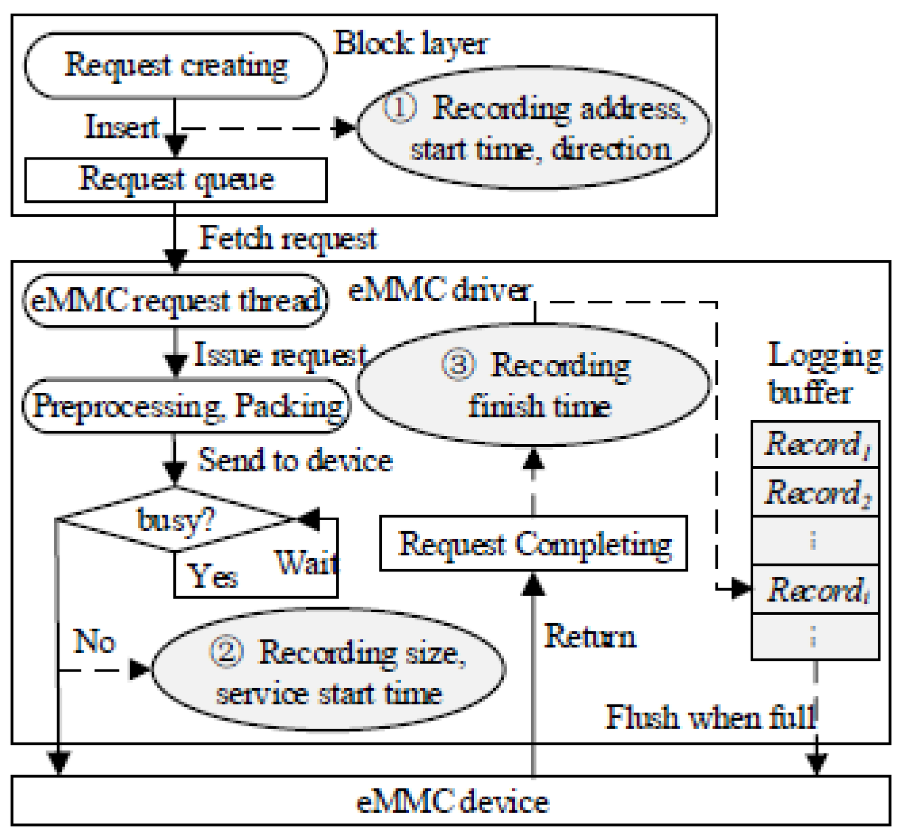

Fig. 1 The structure of BIOTracer.

Trace Collection

Table 1. Trace collection details.

Idle *(10pm - 6am)*: Idle status.
Booting *(30 seconds)*: Launching the smartphone.
CallIn,CallOut *(1 hour)*: Mimicking a phone interview including answering, talking, listening, and hanging out.
CameraVideo, AngryBrid, GoogleMaps *(0.5 - 1 hour)*: Recording a video, playing games, driving navigating.
Facebook, Tiwtter, Amazon, Email, Messaging *(10 - 20 minutes)*: Viewing comments, searching people or items, viewing pictures, and composing replies.
WebBrowsing, Youtube, Radio, Music *(1 - 1.5 hours)*: Reading news, watching online videos, listening radio, and listening music.
Movie, Installing *(10 minutes)*: Watching locally stored movie, installing game applications via WIFI connection.

Trace Analysis Results

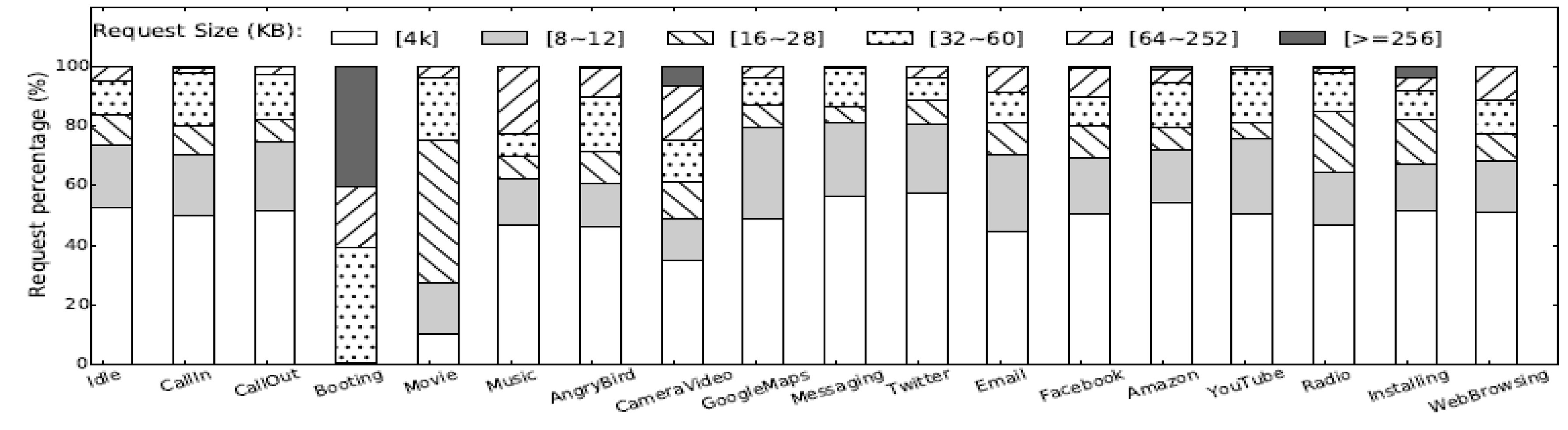

Fig. 3 Request size distribution .

★Characteristic: Small size requests take a significant percentage of the total number of requests in most applications. In 15 out of the 18 traces, majority requests (44.9%- 57.4%) are small requests (i.e., 4 KB).

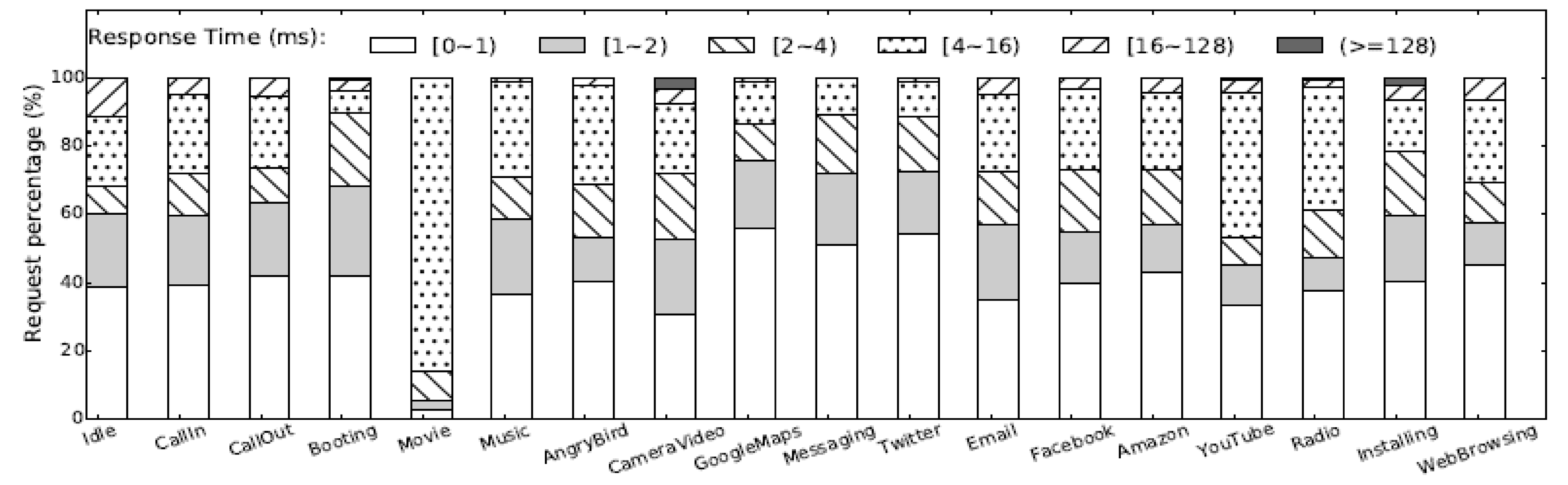

Fig. 4 Response time distribution .

★Characteristic: The eMMC device will enter into a low power mode if the inter-arrival time is longer than its power- saving threshold. Thus, in some applications periodic mode switching may happen.

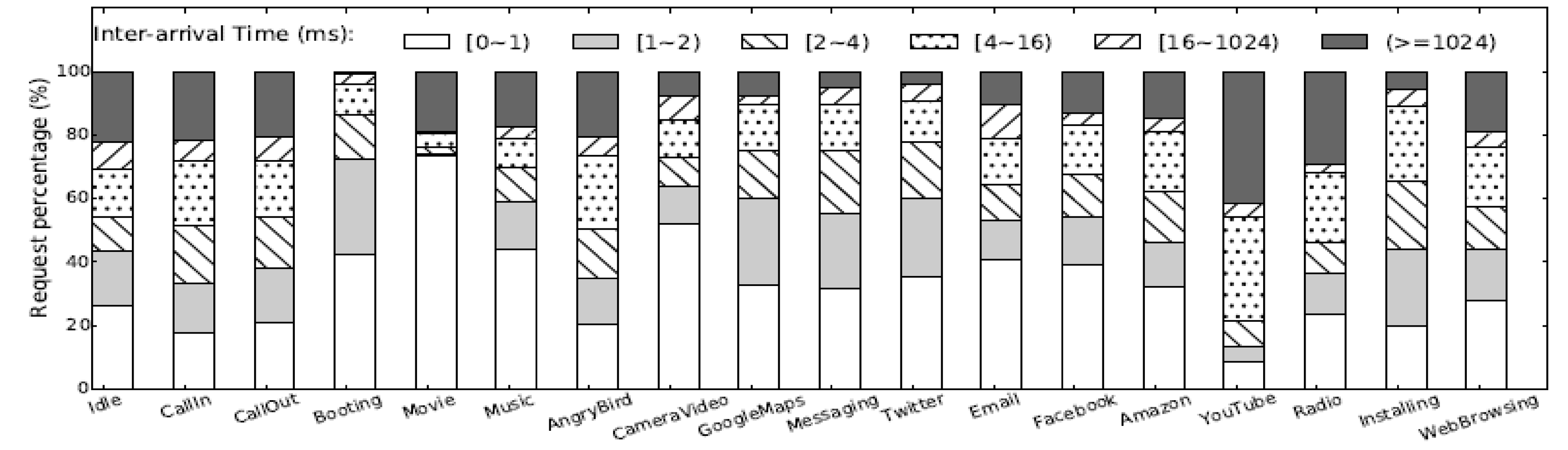

Fig. 5 Inter-arrival time distribution .

★Characteristic : The average request inter-arrival times are long in most applications. 13 out of the 18 applications have an average request inter-arrival time at least 200 ms. In 10 out of the 18 traces, more than 20% inter-arrival times are larger than 16 ms.

Implications Examples

★Implication 1: Small size requests take a significant percentage of the total number of requests in most applications. In 15 out of the 18 traces, majority requests (44.9%- 57.4%) are small requests (i.e., 4 KB). Implication 5: Characteristic 2 discloses that in 15 out of the 18 traces majority requests (44.9%-57.4%) are small single-page (4 KB) requests. The implication is that serving the large amounts of small requests quickly could improve the overall performance of an eMMC device. One feasible way to better serve these small requests is to use SLC flash, which has a better read/write performance than that of MLC flash (see Table 4). However, SLC is much more expensive than MLC. Fortunately, an MLC flash cell can work in the SLC mode by selectively using its fast pages, and thus, obtains an SLC-like [2]. Thus, the performance gain is achieved at the cost of 50% capacity loss.

★Implication 2: FTL in eMMC needs to be tailored to match the I/O characteristics of smartphones (from characteristics 3 and 6). We find that 13 of 18 traces have an average request inter-arrival time more than 200 milliseconds, which is long enough for a garbage collection process to complete. Thus, garbage collection mechanism in the FTL should be redesigned so that garbage collections are launched during the execution of these non-data-intensive applications. In this way, users cannot perceive performance degradation due to garbage collection.

Conclusions

To understand the smartphone applications' I/O patterns, we implement an I/O monitor tool called BIOtracer and integrate it into Android kernel 3.4 on a Nexus 5. Next, we conduct a comprehensive analysis on 25 traces. Six I/O characteristics have been observed. Next, 5 implications for eMMC design are derived based on the characteristics. Finally, we conduct a case study to demonstrate how to apply the implications to optimize eMMC design.

Acknowledgement

This work was supported by the US National Science Foundation under GrantCNS-1320738.

References

[1] H. Kim and D. Shin. Optimizing storage performance of android smartphone. In *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, page 95. ACM, 2013.

[2].S. Im and D. Shin. Comboftl: Improving performance and lifespan of mlc flash memory using slc flash buffer. *Journal of Systems Architecture*, 56(12):641–653, 2010.

1.Computational Science Research Center
2.Computer Science Department