*International Journal of Advanced Smart Convergence Vol.10 No.4 52-58 (2021)*
*http://dx.doi.org/10.7236/IJASC.2021.10.4.52*

IJASC 21-4-7

# Characterization Studies on Data Access Bias in Mobile Platforms

Hyokyung Bahn

*Professor, Department of Computer Engineering, Ewha University, Korea*
*bahn@ewha.ac.kr*

***Abstract***

*Data access bias can be observed in various types of computing systems. In this paper, we characterize the data access bias in modern mobile computing platforms. In particular, we focus on the access bias of data observed at three different subsystems based on our experiences. First, we show the access bias of file data in mobile platforms. Second, we show the access bias of memory data in mobile platforms. Third, we show the access bias of web data and web servers. We expect that the characterization study in this paper will be helpful in the efficient management of mobile computing systems.*

**Keywords:** *Data access, bias; mobile data; web access; characterization study; file access; memory access.*

## 1. Introduction

With the explosive growth in mobile applications and the advances in software platform technologies, mobile data accesses are now become essential in our lives [1-5]. People are increasingly working with their mobile devices, and a variety of mobile applications emerge day by day [6]. In reality, the hardware specification of a mobile device is similar to that of a PC [7-9]. For example, Google Pixel 5, the 2020 version of the Android reference phone, consists of Qualcomm Snapdragon 765G, Octa-core CPU (1 × 2.4 GHz Kryo 475 Prime & 1 × 2.2 GHz Kryo 475 Gold & 6 × 1.8 GHz Kryo 475 Silver), Adreno 620 GPU, 8 GB LPDDR4X memory, and 128GB UFS 2.1 storage, which is sufficient to execute traditional desktop applications. Thus, we can do official works like social broadcasting, video conferencing, and stock trading with our mobile phones.

As a mobile device increasingly absorbs the functionality of desktop computers, we need to characterize its applications and data access characteristics. Meanwhile, when we investigate data access characteristics in computing systems, the bias can be observed in any kind of system components [10-12]. In this paper, we characterize the data access bias that appears in mobile platforms. In particular, we focus on three subsystems based on our experiences. First, we characterize the access bias in file data. Our observations show that the top 5-10% file data account for about 80% of total data accesses. Second, we characterize the access bias in memory data. Our observations show that the top 10% memory data account for 80% of total memory data accesses. Third, we characterize the access bias in web data and web servers. Our observations show that the

Manuscript Received: October. 3, 2021 / Revised: October. 7, 2021 / Accepted: October. 10, 2021
Corresponding Author: bahn@ewha.ac.kr
Tel: +82-2-3277-2368, Fax: +82-2-3277-2306
Professor, Department of Computer Engineering, Ewha University, Korea

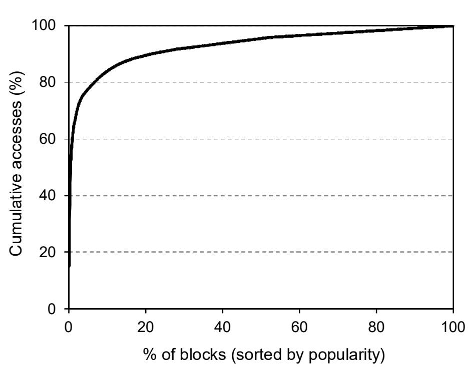
top 30% web data account for 80% of total web accesses. Similarly, the top 10% of web servers account for 80% of total web accesses. We expect that the characterization in this study will be helpful in the efficient management of mobile platforms.

The remainder of this paper is organized as follows. Section 2 describes the characterization result of file data access bias in mobile platforms. Section 3 describes the memory access bias in mobile platforms, and Section 4 characterizes the bias in web data accesses. Finally, Section 5 concludes this paper.

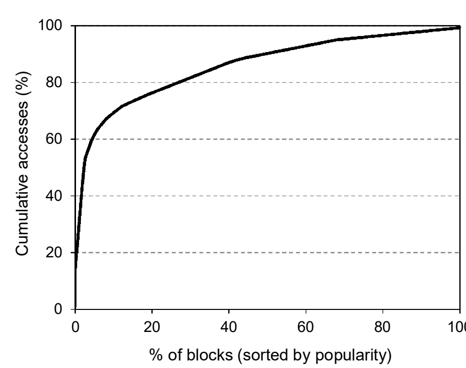
## 2. Access Bias in File Data

In this section, we characterize the file data access bias in mobile platforms. To collect file access traces in mobile phones, we utilize the strace utility, which has the ability of tracing the system calls of a process [20]. We also capture the file access trace of desktop PCs and compare the data access bias in mobile phones and PCs. For applications, we select Facebook, Angrybird, Twitter, and Web browser for mobile applications, and LibreOffice, Cscope, Firefox, and Gnuplot for PC applications. The duration of the trace collection period was in the range of 15-20 minutes for each application. We ran the four applications sequentially and then repeated them three times to see the effect of multitasking. Note that a graduate student in our research group participated in our trace collection process.

Figure 1 depicts the cumulative references for file data in mobile devices in comparison with those in PC systems. In the figure, the $x$-axis represents the percentage of file blocks sorted by their popularity rankings and the $y$-axis shows the cumulative accesses made by the given fraction of file blocks. As file systems store files by the same size unit called blocks, the meaning of "block" in Figure 1 is a file block belonging to the files in storage. As shown in Figure 1(a), in mobile devices, the top 5-10% of file data account for 80% of total accesses, which is highly biased. However, in PC environments, we can see a certain biased popularity in file accesses, but it is not stronger than mobile systems. In particular, about top 30% of file data account for 80% of total accesses in PC systems.




(a) mobile




(b) PC

**Figure 1. Cumulative file access distributions for mobile and PC systems.**

Figure 2 shows the file access ratio of mobile applications in terms of the access operations in comparison with PC environments. As shown in the figure, all mobile applications show write-intensive reference characteristics. In particular, the ratio of write operations is over 60%. Such high percentage of write operations happens as mobile applications utilize the SQLite database, which is a library used in file manipulations. This is different from traditional PC programs where read operations are dominant in file references [16]. As can be seen from the figure, the ratio of write operations in PC environments is at most 15%.

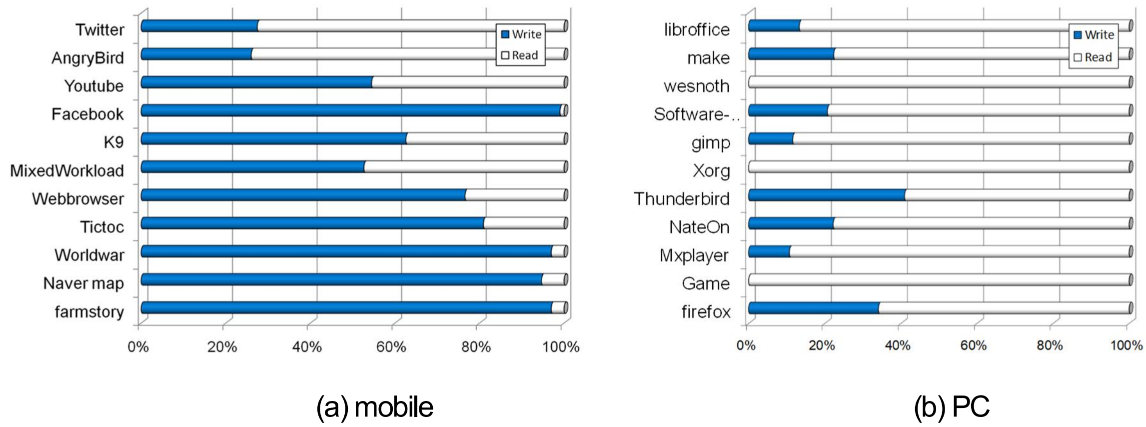



**Figure 2. File access distributions for mobile and PC systems.**

Figure 3 shows the access distributions of storage in mobile devices. Storage access traces in our experiment were extracted by the ftrace utility supported in Google Android. In this experiment, we executed 5 applications consisting of Angry Bird, Candy Crush, File Browser, Temple Run2, and Balance 3D sequentially and repeated them three times. The duration of each execution was in the range of 15-20 minutes. As storage accesses consist of swap accesses as well as file accesses, we included all storage accesses in this graph. In particular, the graph plots the cumulative number of accesses in the storage of mobile devices. In the figure, the *x*-axis is the ratio of accessed data sorted by their reference count and the *y*-axis is the percentage of references for the given ratio of data. For example, 10% in the *x*-axis refers to the top 20% data, and the corresponding point in the *y*-axis is the percentage of storage accesses they made. As can be seen from this figure, the top 50% data account for 80% of total storage accesses. This means that the bias of accesses in mobile storage is relatively weak.

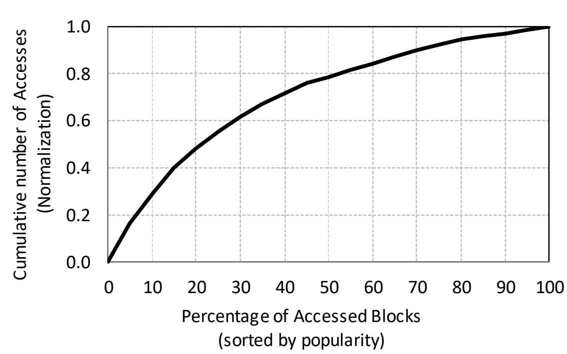



**Figure 3. Cumulative distribution of storage accesses in mobile platform.**

## 3. Access Bias in Memory Data

In this section, we characterize the memory references in mobile platforms. To collect memory access traces of Android applications, we used the Cachegrind utility of the Valgrind toolset [21]. In our experiment, we executed 4 mobile applications, Facebook, Game, Youtube, and Web browser. Specifically, we ran these applications sequentially and repeated them three times. The duration of each execution was in the range of 15-20 minutes. Similar to file data collection, a graduate student in our research group participated in the memory trace collection process. Figure 4 shows the number of memory references that happens on each memory data. In the figure, the red plot represents the write operation while the blue plot represents the read operation. As we see in the figure, a limited number of memory data account for a large fraction of the memory references.

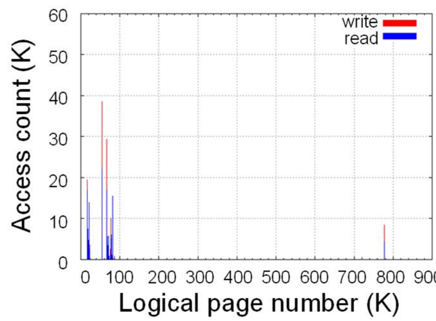
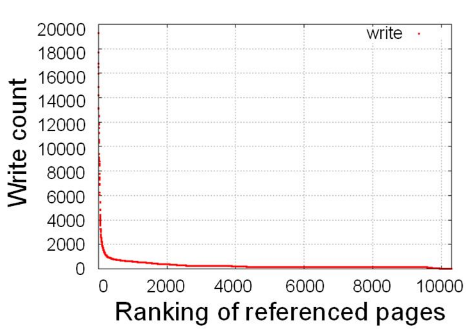


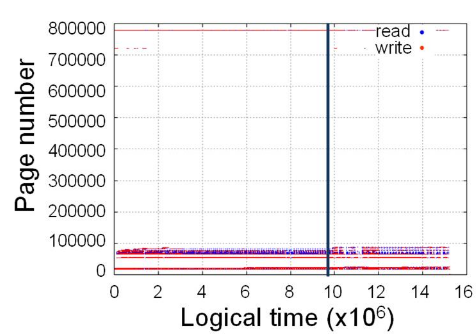

**Figure 4. Memory access count for each memory location in mobile platforms.**

To investigate the characteristics of write operations in memory of mobile systems, we plot the distributions of write operations as the data ranks increase in Figure 5(a). In the figure, the *x*-axis is the rank of data sorted by the number of write operations that occurred on that data and the *y*-axis is the number of write operations on that ranking. As we see from this figure, the write operations made by mobile systems are excessively biased. In particular, the top 10% data account for 80% of total write operations, which implies that write operations in the memory of mobile systems are mostly generated by a certain hot data. Note that this is different to PC environments where the top 50% data typically account for 80% of write operations [17].

Figure 5(b) shows the memory locations that have been accessed as time progresses. In this figure, the *x*-axis is the logical time, which is increased by 1 for each memory access and the *y*-axis is the memory location, which is represented as page numbers. In this figure, the blue plot represents the read operation and the red plot represents the write operation. As shown in the figure, write operations are biased to a certain hot memory locations and they are consistently accessed as time progresses. The vertical line in the figure is the time where the application finishes its launch. As shown in the figure, the characteristics of memory data accesses do not change significantly even after the application finishes its launch.




(a) Write count distributions




(b) Accessed pages over time

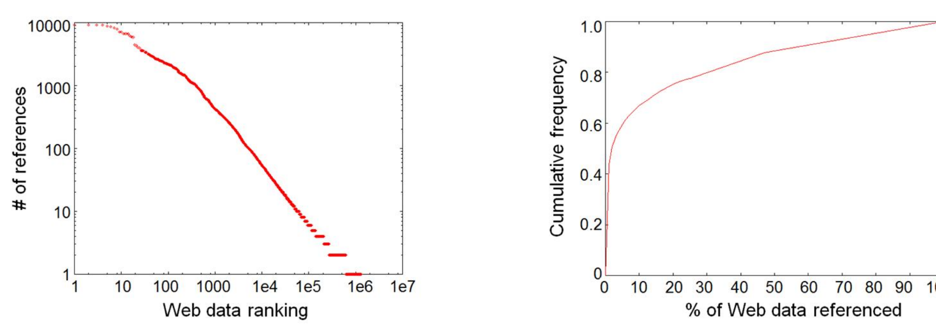
**Figure 5. Access bias in smartphone memory data.**

## 4. Access Bias in Web Data

In this section, we characterize the access bias in web data accesses [13, 14]. We used the trace captured by the proxy server at KREN (Korea Education Network) [19], which acts as agents on behalf of users to send HTTP requests to web servers. The trace collection period was 6 days and the total number of web pages and web servers in the trace were 1,308,730 and 988,036, respectively.

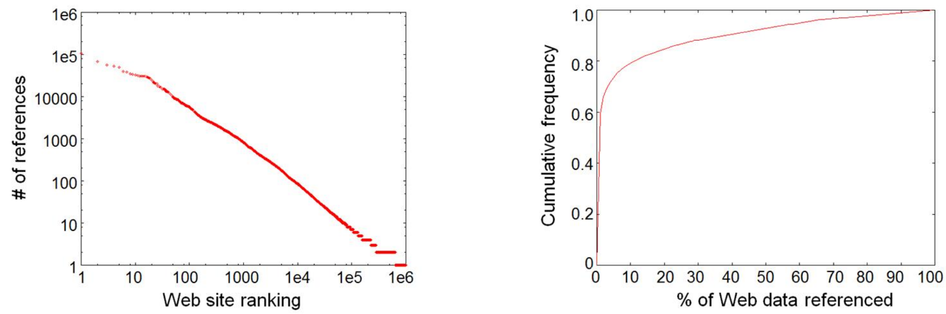
Apparently, not all web data are evenly accessed. Figure 6(a) shows the frequency of a web data that has been referenced versus the popularity rank of the web data. In this figure, the $x$ and $y$ axes are all in the log scale. As shown in the figure, accesses of web data are excessively biased to a certain hot web data. In particular, the graph seems to be a straight line, which implies that the reference probability of a web data whose popularity rank is $i$ is proportional to $1/i^a$, where $a$ is the slope of the line. We call this distribution a Zipf distribution, which is excessively biased [15, 18]. In Figure 6(b), we plot the cumulative references versus the ratio of the web data referenced. Web data in the $x$-axis are sorted by their reference count. This graph shows that the top 10% web data account for 70% of web accesses and the top 30% web data account for 80% of web accesses. That is, references for web data are significantly biased to a certain hot data, which can be modeled by a skewed distribution of Zipf.

Let us now consider the bias in web sites. Figure 7(a) shows the reference count of web sites versus the rank of the web sites, where rank 1 is the most popular web site. In this figure, both $x$ and $y$ axes are in log scale. Similar to the web data characterization, references are significantly biased to a certain hot web sites. The distribution can also be modeled by a Zipf distribution. Figure 7(b) shows the cumulative references of web sites versus the ratio of the web sites visited. The web sites in the $x$-axis are sorted by the reference counts. This figure shows that the top 10% web sites account for 80% of web references and the top 30% web sites account for 90% of web references.




(a) Frequency versus ranking (b) Cumulative distributions

**Figure 6. Access bias in Web data.**




(a) Frequency versus ranking (b) Cumulative distributions

**Figure 7. Access bias in Web sites.**

## 5. Conclusion

In this paper, we characterized the data access bias in mobile platforms. In particular, we analyzed the bias in web data references, file data references, and memory data references. Our analysis showed that the top 5-10% file data account for 80% of references, the top 10% memory data account for 80% of references, and top 30% web data account for 80% of references. We expect that the characterization result in this paper will provide implications and guide for the efficient management of mobile systems.

## Acknowledgement

This work was supported in by the ICT R&D program of MSIP/IITP (2018-0-00549, extremely scalable order preserving OS for many core and non-volatile memory) and (2019-0-00074, developing system software technologies for emerging new memory that adaptively learn workload characteristics).

## References


[1] S. Bae, H. Song, C. Min, J. Kim, Y. Eom, “EIMOS: enhancing interactivity in mobile operating systems,” Lecture Notes in Computer Science, vol. 7335, no. 1, pp. 238-247, 2012.
DOI: https://doi.org/10.1007/978-3-642-31137-6_18

[2] S. Hyun, H. Bahn, K. Koh, “LeCramFS: an efficient compressed file system for flash-based portable consumer devices,” IEEE Trans. Consumer Electronics, vol. 53, no. 2, pp. 481-488, 2007.
DOI: https://doi.org/10.1109/TCE.2007.381719

[3] E. Lee, H. Bahn, “Electricity usage scheduling in smart building environments using smart devices,” The Scientific World Journal, vol. 2013, no. 1, 2013.
DOI: https://doi.org/10.1155/2013/468097

[4] J. Park, H. Lee, S. Hyun, K. Koh, H. Bahn, “A cost-aware page replacement algorithm for nand flash based mobile embedded systems,” in Proc. the seventh ACM international conference on Embedded software (EMSOFT), 2009
DOI: https://doi.org/10.1145/1629335.1629377

[5] D. Kim, E. Lee, S. Ahn, H. Bahn, “Improving the storage performance of smartphones through journaling in non-volatile memory,” IEEE Trans. Consumer Electronics, vol. 59, no. 3, pp. 556-561, 2013.
DOI: https://doi.org/10.1109/TCE.2013.6626238

[6] F. Huang, X. Li, S. Zhang, J. Zhang, J. Chen, and Z. Zhai, “Overlapping community detection for multimedia social networks,” IEEE Trans. Multimedia, vol.19, no. 8, pp. 1881-1893, 2017.
DOI: https://doi.org/10.1109/TMM.2017.2692650

[7] I. Nayeem and R. Want, "Smartphones: past, present, and future," IEEE Pervasive Computing, vol. 13, no. 4, pp. 89-92, 2014.
DOI: https://doi.org/10.1109/MPRV.2014.74
[8] J. Kim, H. Bahn, "Analysis of smartphone I/O characteristics—Toward efficient swap in a smartphone," IEEE Access, vol. 7, no. 1, pp. 129930-129941, 2019.
DOI: https://doi.org/10.1109/ACCESS.2019.2937852
[9] J. Kim, H. Bahn, "Maintaining Application Context of Smartphones by Selectively Supporting Swap and Kill," IEEE Access, vol. 8, no. 1, pp. 85140-85153, 2020.
DOI: https://doi.org/10.1109/ACCESS.2020.2992072
[10] E. Lee, H. Kang, H. Bahn, K.G. Shin, "Eliminating periodic flush overhead of file I/O with non-volatile buffer cache," IEEE Trans. Computers, vol. 65, no. 4, pp. 1145-1157, 2014.
DOI: https://doi.org/10.1109/TC.2014.2349525
[11] E. Lee, J. Kim, H. Bahn, S. Lee, S.H. Noh, "Reducing write amplification of flash storage through cooperative data management with NVM," ACM Trans. Storage, vol. 13, no. 2, pp. 1-13, 2017.
DOI: https://doi.org/10.1145/3060146
[12] E. Lee, H. Bahn, "Caching strategies for high-performance storage media," ACM Trans. Storage, vol. 10, no. 3, pp. 1-22, 2014.
DOI: https://doi.org/10.1145/2633691
[13] H. Bahn, Y.H. Shin, K. Koh, "Analysis of Internet reference behaviors in the Korean Education Network," Lecture Notes in Computer Science, vol. 2105, no. 1, pp. 114-127, 2001.
DOI: https://doi.org/10.1007/3-540-47749-7_9
[14] H. Bahn, H. Lee, S.H. Noh, S.L. Min, K. Koh, "Replica-aware caching for web proxies," Computer Communications, vol. 25, no. 3, pp. 183-188, 2002.
DOI: https://doi.org/10.1016/S0140-3664(01)00365-6
[15] G. K. Zipf, Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology, Addison Wesley Press, 1949.
[16] D. Kim, H. Bahn, "Exploiting write-only-once characteristics of file data in smartphone buffer cache management." Pervasive and Mobile Computing, vol. 40, no. 1, pp. 528-540, 2017.
DOI: https://doi.org/10.1016/j.pmcj.2017.01.004
[17] S. Lee, H. Bahn, and S. Noh, "CLOCK-DWF: A write-history-aware page replacement algorithm for hybrid PCM and DRAM memory architectures," IEEE Trans. Computers, vol. 63, no. 9, pp. 2187-2200, 2014.
DOI: https://doi.org/10.1109/TC.2013.98
[18] E. Lee, J. Whang, U. Oh, K. Koh, H. Bahn, "Popular channel concentration schemes for efficient channel navigation in internet protocol televisions," IEEE Trans. Consumer Electronics, vol. 55, no. 4, pp. 1945-1949, 2009.
DOI: https://doi.org/10.1109/TCE.2009.5373754
[19] The Archives of KREN, https://web.archive.org/web/20180809051356/http://kren.ne.kr/
[20] W. E. Loewe, R. M. Hedges, T. T. McLarty, and C. J. Morrone, "LLNL's parallel I/O testing tools and techniques for ASC parallel file systems," in Proc. IEEE Cluster Computing Conference, pp.1-4, 2004.
[21] N. Nethercote and J. Seward, "Valgrind: a program supervision framework," Electronic Notes in Theoretical Computer Science, vol. 89, no. 2, pp. 44-69, 2003.
DOI: https://doi.org/10.1016/S1571-0661(04)81042-9