JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE, VOL.15, NO.2, APRIL, 2015
http://dx.doi.org/10.5573/JSTS.2015.15.2.223
ISSN(Print) 1598-1657
ISSN(Online) 2233-4866

Characterizing Memory References for Smartphone Applications and Its Implications

Soyoon Lee and Hyokyung Bahn

***Abstract*—As smartphones support a variety of applications and their memory demand keeps increasing, the design of an efficient memory management policy is becoming increasingly important. Meanwhile, as nonvolatile memory (NVM) technologies such as PCM and STT-MRAM have emerged as new memory media of smartphones, characterizing memory references for NVM-based smartphone memory systems is needed. For the deep understanding of memory access features in smartphones, this paper performs comprehensive analysis of memory references for various smartphone applications. We first analyze the temporal locality and frequency of memory reference behaviors to quantify the effects of the two properties with respect to the re-reference likelihood of pages. We also analyze the skewed popularity of memory references and model it as a Zipf-like distribution. We expect that the result of this study will be a good guidance to design an efficient memory management policy for future smartphones.**

***Index Terms*—Smartphone, NVM, write references, temporal locality, Zipf-like distribution**

I. INTRODUCTION

With the explosive dissemination of smartphones around the world, smartphones support a variety of applications and their memory demand also keeps increasing. Accordingly, design of an efficient memory management policy in smartphones is becoming increasingly imporant. In particular, as nonvolatile memory (NVM) technologies such as PCM (Phase Change Memory) and STT-MRAM (Spin Transfer Torque Magnetic RAM) have emerged and will be candidates of new memory media in smartphones, characterizing memory references for these emerging NVM-based smartphone memory systems is needed.

NVM has desirable properties, such as high density, low leakage power, and non-volatility, to be adopted as a memory medium of smartphones. However, current NVM technologies have some weaknesses in their write operations to substitute DRAM memory in its entirety. In case of PCM, a write access time is about 6-10 times slower than that of DRAM and the number of write operations allowed to each PCM cell is limited to 10^7-10^8 [1-3]. In case of STT-MRAM, write energy is 5-10 times higher than that of DRAM [4, 5]. Therefore, mitigating costly write operations on PCM and STT-MRAM is a crucial factor and write references in memory should be carefully managed. To dilute the effect of write operations in NVM-based memory systems, comprehensive analysis of memory references in smartphones is needed.

To do this, we develop memory trace collectors and extract memory reference traces from various kinds of smartphone applications. We, then, analyze the collected memory reference traces. Specifically, we analyze the memory references in terms of temporal locality and reference frequency to quantify their effects on the likelihood of future references. Specifically, we focus on

Manuscript received May. 12, 2014; accepted Jan. 29, 2015
A part of this work was presented in Korean Conference of Semiconductors, Seoul in Korea, Feb. 2014.
Department of Computer Science & Engineering, EWHA Womans University, Seoul 120-750, Korea
E-mail : bahn@ewha.ac.kr

the estimation of future write references. This result can be applied to the design of a memory management policy (e.g., replacement and allocation) in the hybrid memory architecture that uses NVM and DRAM together. For example, considering our analysis results, the policy could be designed to absorb as many write references as possible with DRAM in order to alleviate the weaknesses of NVM in writes.

We also analyze the skewed popularity of memory references in smartphone applications and model it as a Zipf-like distribution. Through this analysis, we can find the working set size of an application, and also determine an appropriate memory size of smartphones. We also expect that an efficient memory management policy for smartphones can be designed.

The remainder of this paper is organized as follows. Section II briefly summarizes how memory traces can be collected. In Section III, we capture page reference characteristics of virtual memory systems in terms of temporal locality and frequency. Section IV presents the analysis of skewed page popularity in memory references and models it as a Zipf-like distribution. We summarize the analysis and note some implications in Section V. Finally, we conclude this paper in Section VI.

II. TRACE COLLECTION PROCESS

To extract memory reference traces in smartphone environments, we implement trace extraction codes in Valgrind 3.8.1 toolset [6]. Specifically, we inject trace collector and analyzer in `cg_sim.c` of Cachegrind. The target system is ODROID-A4 Android smartphone and we filter out memory references that are accessed directly from the cache memory layers (L1, L2, and LLC) and gather only the memory references observed at the main memory system.

To explore a wide range of smartphone applications, we capture memory access traces from six smartphone applications used on Android, namely, the angrybirds a game, the facebook a social network service, the mxplayer a media player, the youtube a video-streaming service using Internet, the farmstory a network game, and the Android web browser. We also gather traces for mixed workloads in multiprogramming environments. Specifically, multi1 concurrently executes web browser, youtube, and angrybirds. Multi2 executes web browser, mxplayer, facebook, and farmstory concurrently.

Our trace analyzer shows the total memory footprint, total write footprint, total reference count, ratio of operations (read vs. write), type of references, etc. The characteristics of these traces are given in Table 1.

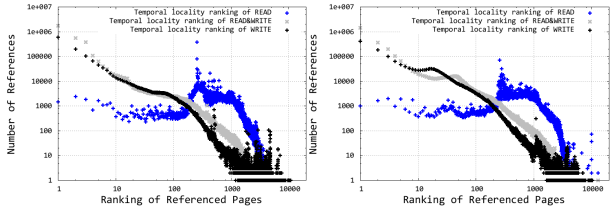
III. ANALYSIS OF TEMPORAL LOCALITY AND REFERENCE FREQUENCY

In this section, we analyze the characteristics of memory references in various smartphone applications in terms of temporal locality and reference frequency.

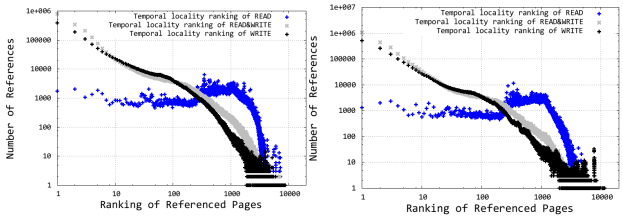
As large energy consumption (STT-MRAM) and long latency (PCM) during a write operation is an important issue in NVM-based memory systems, we separately analyze the characteristics of write references in terms of temporal locality and reference frequency. Also, it is not clearly known whether considering read and write histories together or considering write history alone is more effective in estimating future write references in memory systems. Hence, in this section, we compare the effectiveness of using write history alone and using both

Table 1. Summary of memory reference characteristics collected in Android smartphones

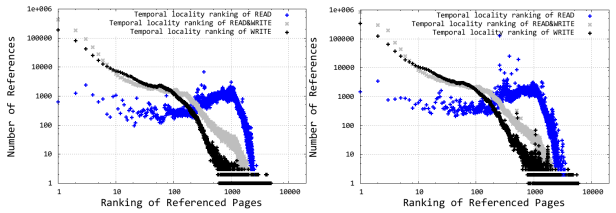
Traces	Memory footprint (KB)	Memory footprint by writes (KB)	Ratio of operations (reads: writes)	Memory access counts			
				Total	Instruction read	Data read	Data write
angrybirds	78,782	46,821	3.50 : 1	18,201,717	980,312	13,387,756	3,822,479
mxplayer	81,838	48,443	3.66 : 1	18,190,547	567,456	13,851,914	3,782,347
youtube	70,287	41,930	4.44 : 1	18,196,504	993,316	14,040,959	3,162,229
web browser	266,092	184,401	4.11 : 1	20,999,999	1,622,628	15,272,935	4,104,436
facebook	203,431	98,414	5.67 : 1	13,653,055	486,165	11,121,174	2,045,716
farmstory	55,030	30,159	6.24 : 1	15,224,670	447,297	12,675,555	2,101,818
multi1	128,974	62,980	6.61 : 1	19,199,986	738,111	15,941,686	2,520,189
multi2	206,684	106,160	6.88 : 1	35,499,985	1,370,884	29,627,038	4,502,063



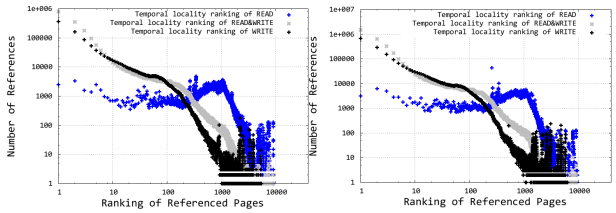
(a) angrybirds (b) mxplayer



(c) youtube (d) web browser

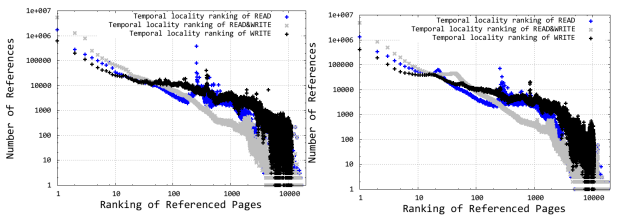


(e) facebook (f) farmstory

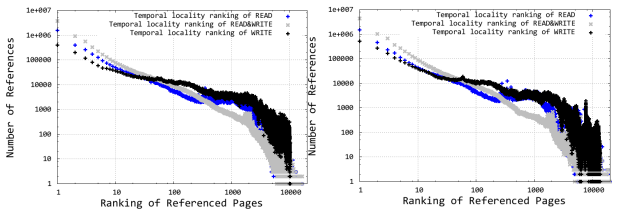


(g) multi1 (h) multi2

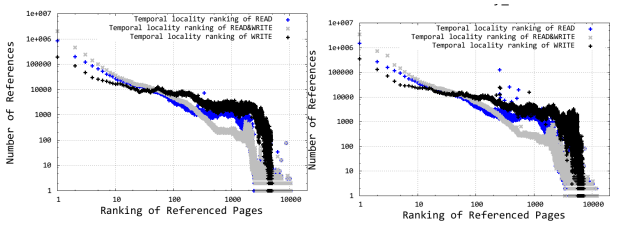
Fig. 1. Distribution of page write references according to the temporal locality ranking of pages.



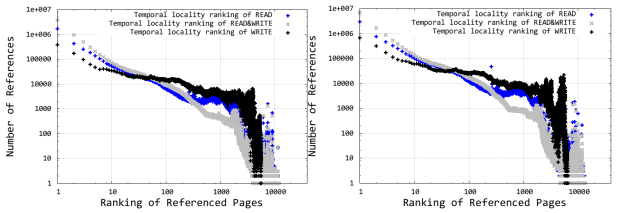
(a) angrybirds (b) mxplayer



(c) youtube (d) web browser



(e) facebook (f) farmstory



(g) multi1 (h) multi2

Fig. 2. Distribution of total references according to the temporal locality ranking of pages.

read/write histories together in terms of temporal locality and frequency to make more accurate estimations of future write references.

1. Temporal Locality

Figs. 1 and 2 show the effect of temporal locality on page references. In the figure, the *x*-axis represents page ranking (i.e., the LRU stack distance) in the LRU list. For example, rank 1 refers to the page at the most recently referenced position in the LRU list. Increase in ranking along the *x*-axis indicates an increase in the LRU stack distance, that is, longer time has passed since the pages have been referenced. The black plots, the gray plots, and the blue plots represent results based on page ranking of write history alone, both read/write history, and read history alone, respectively. The *y*-axis represents the distribution of references for the page ranking in the *x*-axis. Figs. 1 and 2 show the distribution of write references and total references (read+write), respectively.

As shown in the figures, the shape of the curves can be modeled as a monotonic decreasing function, implying that a more recently referenced page is more likely to be re-referenced in the near future.

Contrasting write temporal locality based on write history (black plot), that based on both read/write histories (gray plot), and that based on read history (blue plot) in Fig. 1, using both read and write histories (gray plot) estimates future write references better within the top 10 rankings. This implies that when a DRAM in the

hybrid memory architecture is so small to hold the highest ranking pages, using both read and write histories is more efficient than using only write history in estimating future writes. However, beyond these top rankings using write history alone and using both read and write histories show similar results. Using read history alone does not perform well in estimating write references as shown in Fig. 1.

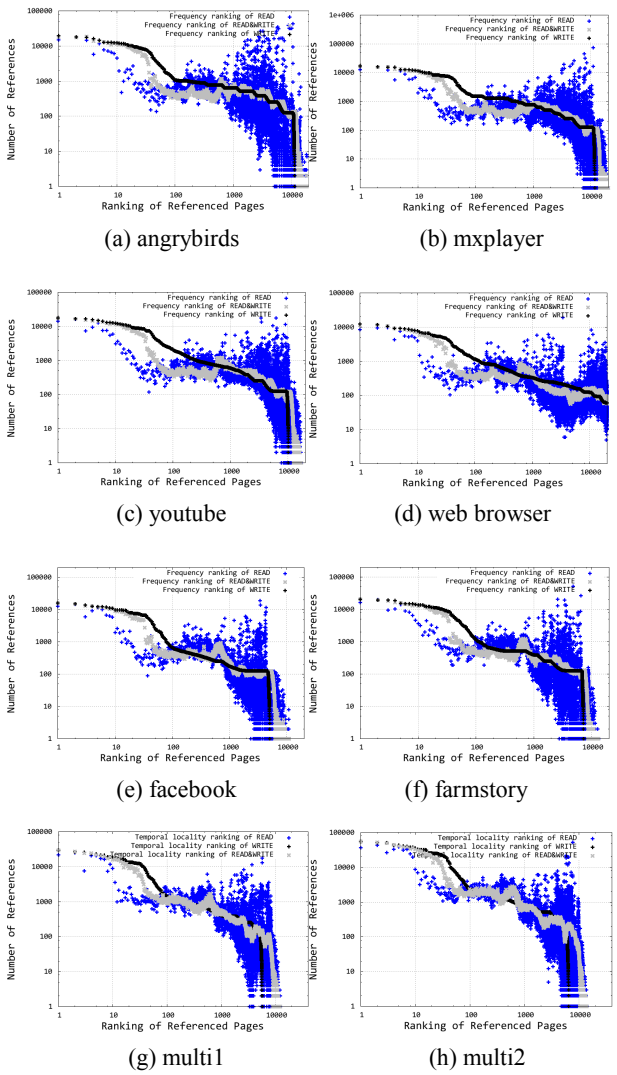
When considering the distribution of total references as shown in Fig. 2, page ranking based on write history alone (black plot), both read/write histories (gray plot), and read history alone (blue plot) show similar results. However, when we limit only top 10 rankings, using both read/write histories estimates future memory references the best, and using write history alone performs the worst.

2. Reference Frequency

Similarly to temporal locality, the effect of frequency on page references can be characterized through page ranking. Here, the page with the highest rank 1 is the page that has the highest frequency count.

We can consider two different types of frequency. The first is the total frequency, which counts the total number of references that appear in the trace, and the second is the so-far-frequency, which counts the number of references that has occurred to the current point. We use the latter in order to observe the impact of frequency on estimation of a page's re-reference likelihood each time in comparison to temporal locality. To do this, we maintain the ranking of pages according to their past frequency counts and examine the number of references that occur again for each ranking.

In Figs. 3 and 4, the *x*-axis represents the ranking of pages based on their past write counts (black plot), read/write counts (gray plot), and read counts (blue plot). The *y*-axis represents the number of references that has occurred on that ranking. To construct the curve, we maintain the ranking of the pages, and as a page in a certain ranking is referenced, we increase the value of the *y*-axis for that ranking by one, possibly resulting in a reordering of the page rankings. Figs. 3 and 4 show the distribution of write references and total (read+write) references, respectively. Note that though the range of the *y*-axis is different for Figs. 1 through 4, the total number of references will be the same for each of the corresponding workloads.



(a) angrybirds (b) mxplayer

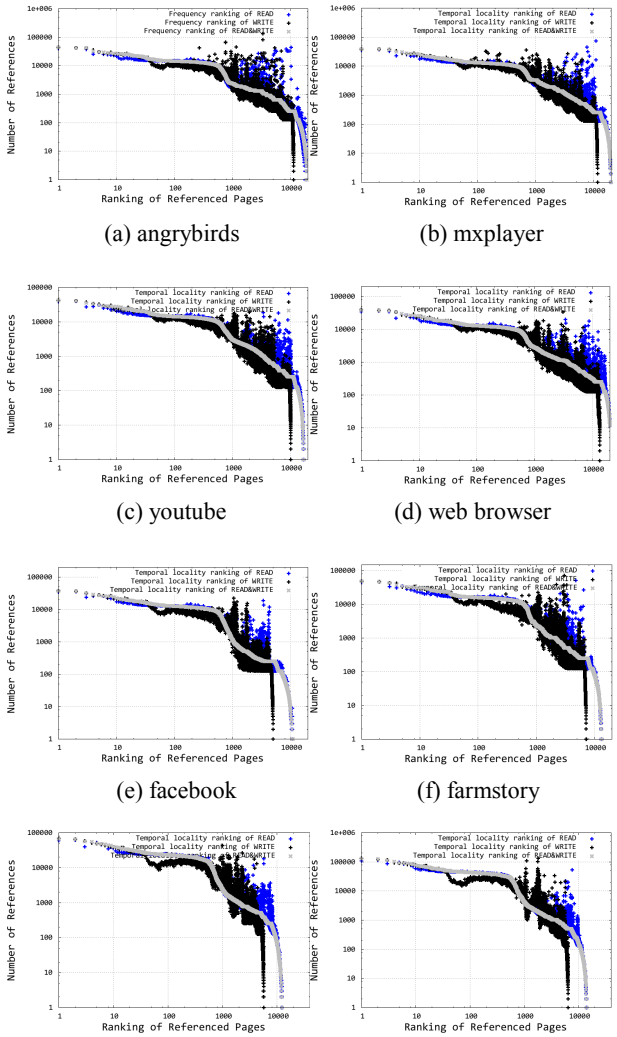
(c) youtube (d) web browser

(e) facebook (f) farmstory

(g) multi1 (h) multi2

Fig. 3. Distribution of write references according to the reference frequency ranking of pages.

As shown in Fig. 3, using write history alone (black plot) leads to a better estimation of future write references than using both read and write histories (gray plot) or using read history (blue plot). This implies that when considering the frequency property as an estimator of future write references, exploiting only write reference history would be a better choice. This is different to the temporal locality case, in which considering both read and write histories together leads to a better estimation of future write references than considering write history only. In contrast, as shown in Fig. 4, using write count only (black plot) and using both read/write counts (gray plot) show similar results when estimating total memory references including both reads and writes. However, as the ranking becomes lower, using both read and write



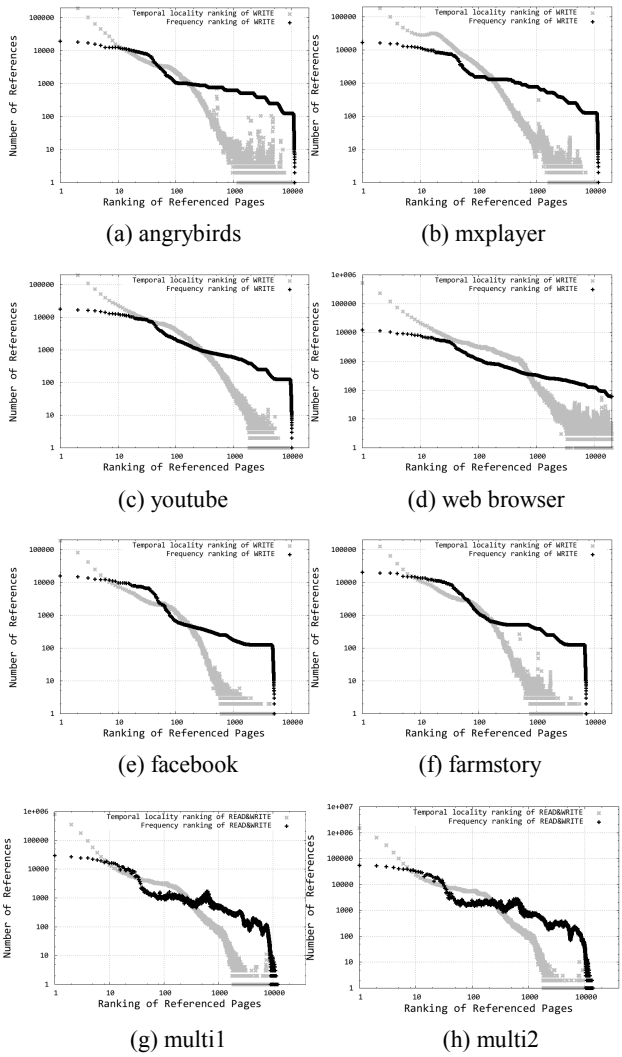
(a) angrybirds (b) mxplayer

(c) youtube (d) web browser

(e) facebook (f) farmstory

(g) multi1 (h) multi2

Fig. 4. Distribution of total references according to the reference frequency ranking of pages.



(a) angrybirds (b) mxplayer

(c) youtube (d) web browser

(e) facebook (f) farmstory

(g) multi1 (h) multi2

Fig. 5. Comparison of temporal locality and reference frequency with respect to the estimation of future write references.

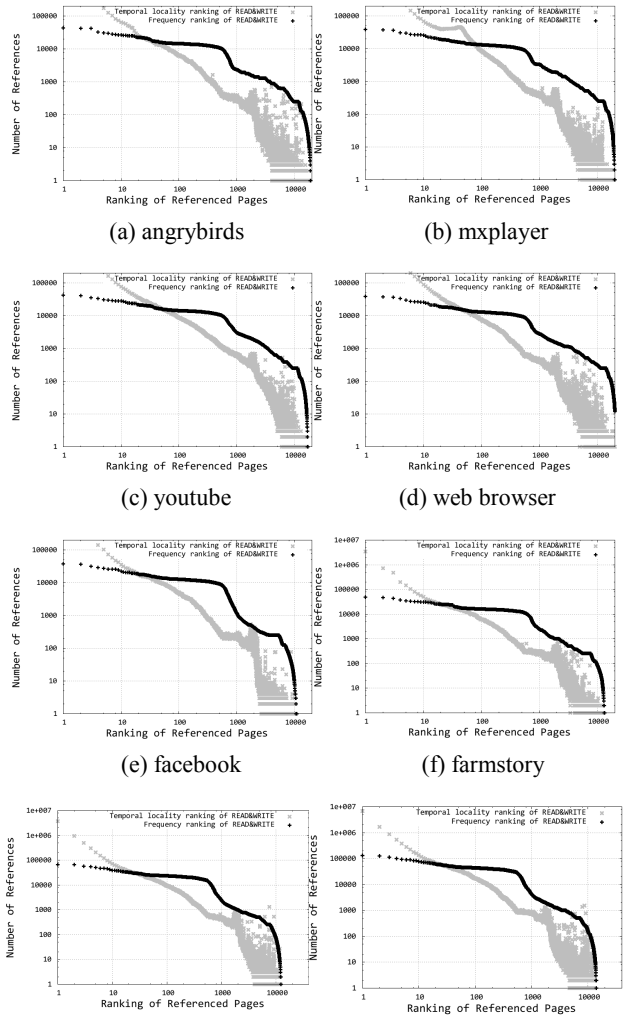
counts will be more effective.

3. Comparing Temporal Locality and Frequency

Let us now compare the temporal locality and frequency based estimators. Based on the analysis result of Figs. 1 through 4, we use write history alone in estimating future write references (Fig. 5), and both read and write histories in estimating all references including reads and writes (Fig. 6). As shown in the figure, the gray plots (temporal locality) are located above the black plots (frequency) in high rankings. This indicates that temporal locality based estimations are more accurate compared to frequency based estimations in high rankings. However, it should be noted that frequency based estimations on the pages of low ranks exhibit larger reference counts than that of temporal locality based estimations.

This indicates that if we want to maximize the expected number of hits by preserving a certain number of pages in limited memory, it would be beneficial to retain the highest ranking pages of temporal locality based estimations first, and then retain some high ranking pages of frequency based estimations. When DRAM and NVM hybrid memory is used, DRAM memory can absorb most write references by preserving the highest ranking pages of temporal locality first, and then some high ranking pages of frequency in DRAM.

This result contradicts the analysis of desktop applications, in which frequency is much stronger than



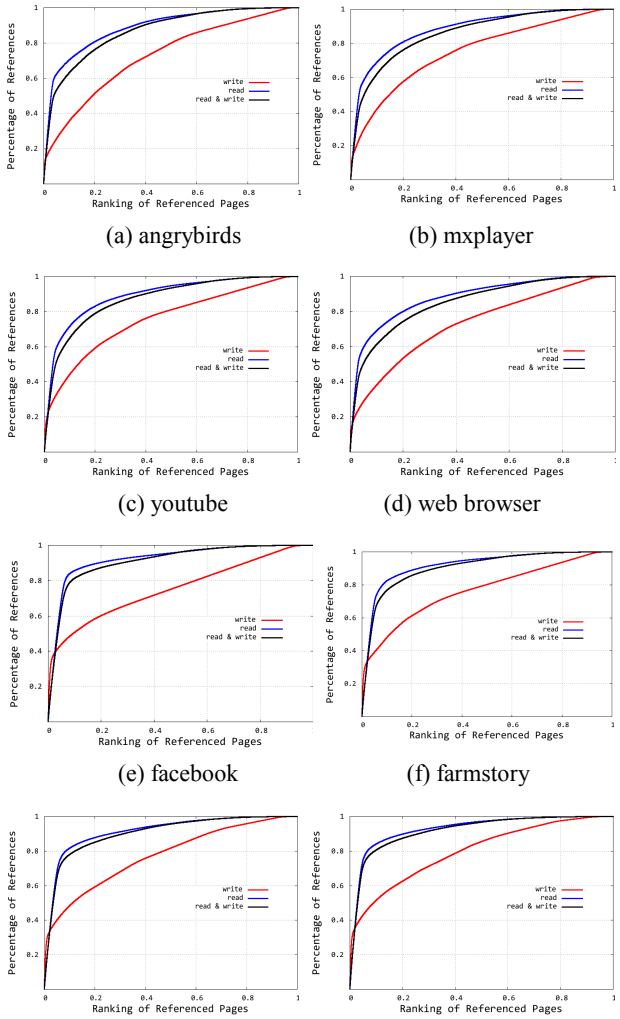
(a) angrybirds (b) mxplayer

(c) youtube (d) web browser

(e) facebook (f) farmstory

(g) multi1 (h) multi2

Fig. 6. Comparison of temporal locality and reference frequency with respect to the estimation of future references.



(a) angrybirds (b) mxplayer

(c) youtube (d) web browser

(e) facebook (f) farmstory

(g) multi1 (h) multi2

Fig. 7. Cumulative distribution of memory references sorted by page ranking.

temporal locality [7]. We cannot pinpoint the exact reason but it may be due to the read-intensive memory workload characteristics of smartphone environments.

In summary, temporal locality is generally a better estimator than frequency in predicting the re-reference likelihood of future references in smartphone memory systems, but combining the two properties appropriately leads to even better results.

IV. Analysis of Skewed Page Popularity in Memory References

1. Cumulative Reference Distribution

In this section, we analyze the skewed popularity of memory references in smartphone applications. This is a performed to find the working set size of each application, and also to determine an appropriate memory size for a smartphone when the multiprogramming degree increases. It will be also helpful to determine the ratio of DRAM and NVM that will be used for emerging smartphone architectures. Specifically, as PCM is slow in write operations and STT-MRAM requires large write energy, a certain amount of DRAM is needed and our study will give some insight to find an appropriate DRAM size. To this end, we analyze memory references for six applications, and show the cumulative distribution of pages sorted by their popularity rankings.

In Fig. 7, we illustrate the cumulative frequency of

references for the fraction of the pages referenced. In this analysis, we use "total frequency" instead of "so-far-frequency." Note that the pages shown in the *x*-axis are sorted into a decreasing order based on the reference counts. The figure shows that 20% of the top ranking pages account for 50-60% of write references, and 80-90% of total references. In case of farmstory and facebook, the popularity skew of write references is strong such that 10% of the top pages account for about 40-50% of total write references. This shows the evidence for the skewed popularity of pages. It also indicates that a certain small amount of DRAM can absorb most of write references.

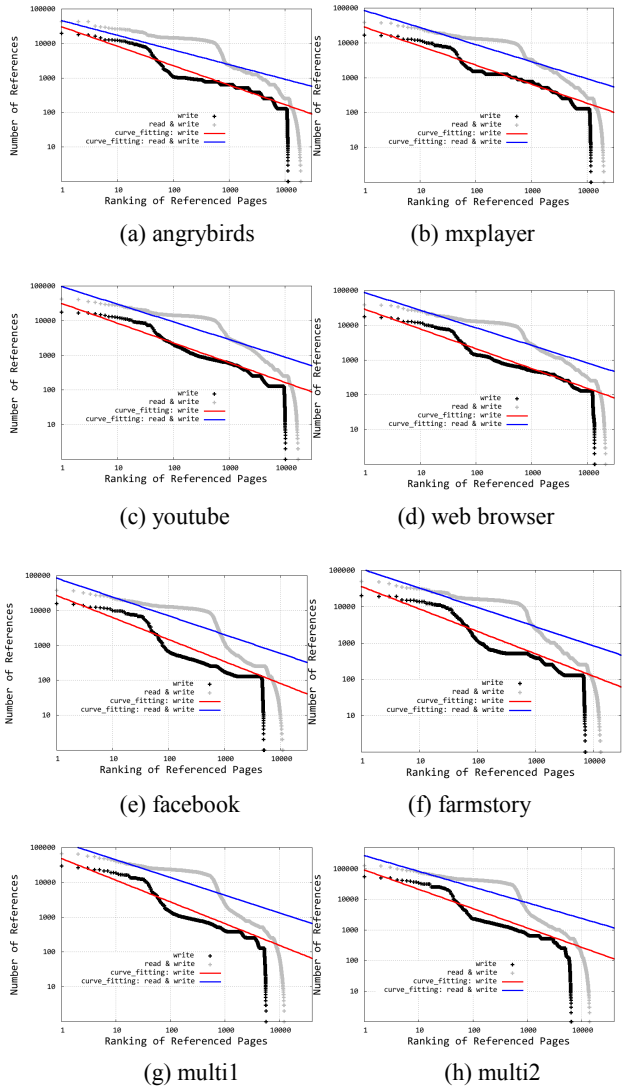
2. Modeling as a Zipf-like Distribution

Our second analysis focuses on the modeling of memory references for smartphone applications. Fig. 8 shows the number of times that a page has been referenced for the ranking of the page, where rank 1 is the most frequently referenced page. Note that both axes in the figure are in log scale, and we also use "total frequency" in this analysis.

The curve in the figure shows that references are excessively biased to some hot pages. The left part of the curves can be well modeled by a straight line (denoted as red and blue lines for write and total references), which implies that the reference frequency of the *i*-th popular page (i.e., rank *i*) is proportional to $1/i^b$, where *b* is the slope of the line. This type of distribution is called a Zipf-like distribution [8].

The value of *b*, which is known as a skew factor of a Zipf-like distribution, is given in Table 2. We obtain the values through curve-fitting analysis. When it approaches 1, the popularity of pages is heavily skewed. As shown in the table, the skew factor is in the range of 0.55 to 0.65 for write references and 0.35 to 0.53 for total references according to applications. As the skew factor of web pages is known to be about 0.8, the bias is relatively weak in our analysis but it still exhibits high skewness [9]. Moreover, as shown in Fig. 8, as the curve-fitting represents only for the left part of the curve, the popularity skew will be higher when considering the remaining part (tail) of the curve.

When comparing total references and write references, the cumulative distribution in Fig. 7 indicates that the skewness of write references is weaker than that of total or read references. However, the Zipf parameter modeled in Table 2 indicates that write references have large skew factors. This may seem strange but it is because the



(a) angrybirds (b) mxplayer

(c) youtube (d) web browser

(e) facebook (f) farmstory

(g) multi1 (h) multi2

Fig. 8. Distribution of memory references according to page ranking (Zipf-like distribution)

Table 2. Zipf parameters for each smartphone applications

	read/write references	write references
angrybirds	0.4980	0.5800
mxplayer	0.3642	0.5631
youtube	0.4953	0.5870
web browser	0.4917	0.5808
facebook	0.5283	0.6525
farmstory	0.5182	0.6346
multi1	0.5068	0.6223
multi2	0.5138	0.6322

curve-fitting is performed only for the left part of the graphs. This implies that the popularity skew of write references is much higher when we consider only top rankings. In reality, if we see only the leftmost part of the graphs in Fig. 7, the slope of write references (red plot) is much steeper than that of total references (black plot). This can be seen apparently in Figs. 7(e)-(h).

V. SUMMARY AND IMPLICATIONS

We performed comprehensive analysis of memory references for various smartphone applications. To do this, we first gathered real memory reference traces for popular smartphone applications and performed various analyses on these traces. The results can be summarized as follows.

- Smartphone memory references are all read-intensive regardless of application types. This is different from memory references of desktop applications, in which a certain type of write-intensive applications exists [7].
- Though smartphone memory references are read-intensive, more than one half of memory footprints accounts for write references. This implies that write references are scattered throughout the footprint and this makes difficult to manage NVM memory systems.

We also identify how to estimate future memory references well, especially for writes, in terms of temporal locality and reference frequency. For an accurate prediction of future memory references, we examine the effect of utilizing read history alone, write history alone, and both read/write histories, and compared them. The results can be summarized as follows.

- In case of temporal locality, using read/write histories together is more effective than using write history alone in estimating future write references, especially for the most recent reference history.
- In the case of frequency, using write frequency alone is more effective than using both read/write histories in estimating future write references.
- When we compare temporal locality and frequency, temporal locality is more effective than frequency for most cases, but combining the two properties appropriately can lead to even better results. In comparison with desktop environments, temporal locality in smartphone applications is much stronger.

The result of this analysis can be used in estimating future write references precisely, thereby absorbing as many write references as possible within a DRAM buffer through adopting hybrid memory architectures.

Finally, we analyze the distribution of memory references in smartphone environments and model it as a Zipf-like distribution. This is important in designing memory management techniques of smartphones because memory footprint is large and writes are scattered in smartphone applications. The results can be summarized as follows.

- References to memory in smartphones are excessively biased to some hot pages and can be modeled as a Zipf-like distribution.
- For write references, top 20% of pages account for 50-60% of all references. In some applications, top 10% of pages accounts for up to 50% of total write references.

We believe that the characterization and analysis study presented in this paper can be helpful to smartphone vendors as well as researchers related to smartphone memory management.

VI. CONCLUSIONS

As a DRAM main memory system has faced with limitations and challenges such as energy and scalability, nonvolatile memory has emerged as a DRAM alternative. This trend is expected to adapt in smartphones in the next few years. However, write references in nonvolatile memory systems should be managed carefully due to their high write energy and slow write access time. For the deep understanding of memory write access features in smartphones, this paper performed comprehensive analysis of memory references for representative smartphone applications. Specially, we focused on the estimation of future write references by quantifying the effects of temporal locality and frequency and investigated the bias of popularity in memory references. Through this analysis, we found which is a better

estimator for future write references in terms of temporal locality and frequency and also showed an effective guidance to estimate the future memory references. This result can be utilized in designing an efficient memory management policy for future smartphones. Our future research will include the memory management techniques in NVM-based smartphone memory systems by exploiting the results of this study.

ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (No. 2013R1A1A2060548) and the Ministry of Science, ICT, and Future Planning (No. 2011-0028825). Hyokyung Bahn is the corresponding author of this paper.

REFERENCES

[1] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "A durable and energy efficient main memory using phase change memory technology," In ISCA09, pp.14-23, 2009

[2] S. Eilert, M. Leinwander, G. Crisenza, "Phase Change Memory: A new memory technology to enable new memory usage models," In IMW09, pp.1-2, 2009.

[3] M.K. Qureshi, V. Srinivasan, and J.A. Rivers, "Scalable high performance main memory system using phase-change memory technology," In ISCA09, pp. 24-33, 2009.

[4] S. Chung et al., "Fully Integrated 54nm STT-RAM with the Smallest Bit Cell Dimension for High Density Memory Application," In IEDM, pp.1-4, 2010.

[5] H. Li, X. Wang, Z. Ong, W. Wong, Y. Zhang and Y. Chen, "Performance, Power, and Reliability Tradeoffs of STT-RAM Cell Subject to Architecture-Level Requirement," IEEE Transactions on Magnetics, Vol.47, No.10, pp.2356-2359, Oct., 2011.

[6] N. Nethercote and J. Seward, "Valgrind: a program supervision framework," Electronic Notes in Theoretical Computer Science, 2003.

[7] S. Lee, H. Bahn, and S. H. Noh, "Characterizing memory write references for efficient management of hybrid PCM and DRAM memory," In MASCOTS11, pp.168-175, 2011.

[8] G. K. Zipf, "Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology," Addison Wesley Press, 1949.

[9] Breslau Lee, Pei Cao, Li Fan, G. Phillips, and S. Shenker, "Web Caching and Zipf-like Distribution: Evidence and Implications," In INFOCOMM99, pp.126-134, 1999.

Soyoon Lee received the BS, MS, and PhD degrees in computer science from Ewha University, Korea, in 2004, 2006, and 2011 respectively. She is currently a research professor of computer science and engineering at Ewha University, Seoul, Republic of Korea. Her research interests include emerging storage systems, operating systems, and embedded systems.

Hyokyung Bahn received the BS, MS, and PhD degrees in computer science from Seoul National University, in 1997, 1999, and 2002, respectively. He is currently an associate professor of computer science and engineering at Ewha University, Seoul, Republic of Korea. His research interests include operating systems, caching algorithms, storage systems, embedded systems, system optimizations, and real-time systems. Prof. Bahn is a member of the IEEE Computer Society, the IEICE, and the Korea Information Science Society.