Impact of File Fragmentation on Android Smartphones

Sangwook Shane Hahn, Inhyuk Yee and Jihong Kim
Department of Computer Science and Engineering, Seoul National University

1 Goals of User Study

We empirically investigate how file I/O performance is affected by file fragmentation on flash storage using 14 smartphones in use. In particular, we examine how quickly file fragmentation occurs again after defragmentation and how much I/O performance is affected by different defragmentation intervals.

2 Evaluation Study Setup

For our study, we collected 14 used Android smartphones as summarized in Table 1. 14 phones include Google Nexus 5 (N5), Nexus 6 (N6), Samsung Galaxy S3 (S3), Galaxy S5 (S5), Galaxy S6 (S6), Galaxy Note 2 (T2), Galaxy Note 3 (T3), Galaxy Note 4 (T4), Galaxy Note 5 (T5), Sony Xperia Z1 (Z1), Xperia Z3 (Z3), LG Optimus G Pro (GP), G5 (G5) and Pantech Vega Iron 2 (I2). In order to avoid possible bias, we have selected these smartphones from five different manufacturers with at least six month's real use. 14 users, like most other smartphone users, heavily used popular Android applications such as `Chrome`, `Messenger`, `Gmail`, `Facebook`, `Twitter` and `Game (Lineage 2)`. Table 2 divides 14 smartphones into 5 categories based on the file system utilization. We inspected file fragmentation on the `data` partition only because the `data` partitions occupied most of the total storage space available and most I/O operations occur in the `data` partition.

For our study, we used the degree $DoF(x)$ of fragmentation of a file `x`, which is defined as the ratio of the number of extents allocated to the file `x` to the ideal (i.e., smallest) number of extents needed for the file `x`. For example, if an 1-GB file `foo` in Ext4 were allocated to 24 extents, DoF(`foo`) would be 3 (i.e., 24/8), because `foo` would have required at least 8 extents even when `foo` was contiguously allocated. (A single extent can cover up to 128 MB in Ext4.) The large DoF value means that the file is highly fragmented.

Table 1: 14 used Android smartphones for evaluation.

	Release Year	Storage	Partition Size	Utilization
Nexus 5	2013	eMMC	26.8 GB	93%
Nexus 6	2014	eMMC	26 GB	57%
Galaxy S3	2012	eMMC	11.6 GB	63%
Galaxy S5	2014	eMMC	27.49 GB	53%
Galaxy S6	2015	UFS	25 GB	96%
Galaxy Note 2	2012	eMMC	24.16GB	81%
Galaxy Note 3	2013	eMMC	26.14 GB	74%
Galaxy Note 4	2014	eMMC	23.9 GB	94%
Galaxy Note 5	2015	UFS	24.2 GB	91%
Xperia Z1	2013	eMMC	24.64 GB	83%
Xperia Z3	2014	eMMC	11.6 GB	89%
Optimus G Pro	2013	eMMC	23 GB	51%
G5	2016	UFS	26.14 GB	80%
Vega Iron 2	2014	eMMC	24.86 GB	90%

3 Degree of File Fragmentation Analysis

We first examined DoF values of files in the `data` partition of the 14 smartphones using `e4defrag`, and Fig. 1 shows cumulative distributions of DoF values on the 14 smartphones. As reported in other investigations such as [1], our inspected smartphones exhibited similar characteristics on file fragmentation. Fragmented files accounted for between 14% and 33% of all files. In particular, on N5, 717 files among its 2,704 files were fragmented. Furthermore, 476 files were fragmented with their DoF values larger than 2. When the file system space was highly utilized, the number of fragmented files tends to be large. For example, on S6, having the highest file system utilization, 33% of its files were fragmented.

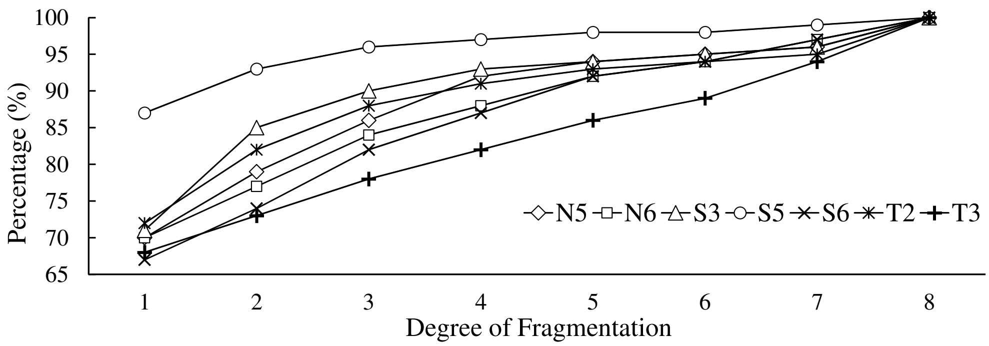
4 File Fragmentation Recurrence

Since our target smartphones have never been defragmented before, the results shown in Fig. 1 are interesting but somewhat expected. A more critical question for our study was to find out how soon file fragmentation recurs after full file defragmentation. If the recurrence interval of file fragmentation were quite large (say, several months), an existing defragmentation would be sufficient for mobile flash storage as well.

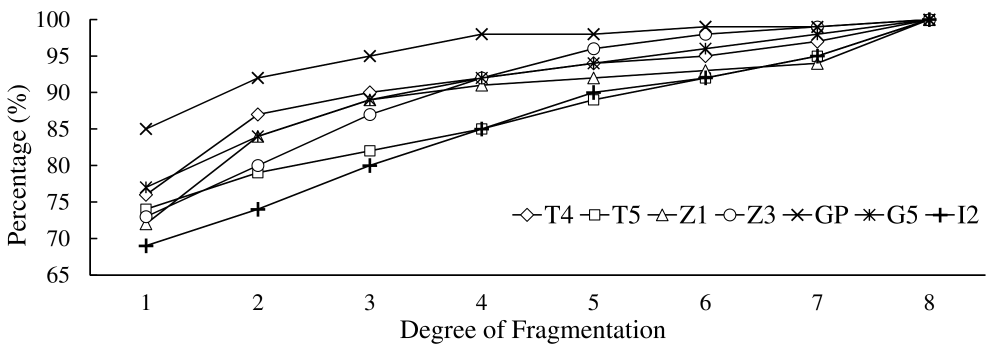
In order to understand file fragmentation recurrence (as well as others), after defragmenting all the files using `e4defrag`, we collected a daily snapshot of each smartphone for the subsequent two-week interval using a custom data collection app. Our snapshot data include DoF values of files and app launching times, Fig. 2 shows the changes in the average DoF values of the files associated with six popular applications, `Chrome`, `Messenger`, `Gmail`, `Facebook`, `Twitter and Game`, on N6. As

Table 2: File system utilizations of 14 smartphones.

50-59%	60-69%	70-79%	80-89%	90-99%
S5, GP	**S3**, G5	**N5**	**N6**, T2, T5, Z1, Z3	**S6**, I2, T3, T4



(a) N5, N6, S3, S5, S6, T2 and T3.



(b) T4, T5, Z1, Z3, GP, G5 and I2.

Fig. 1: Cumulative distributions of DoF values of files in the `data` partition.

shown in Fig. 2, file fragmentation recurred quickly after the full file system defragmentation. For most applications on N6, file fragmentation occurs again in a week since the full defragmentation. Fig. 5 shows the changes in the average DoF values of the files associated with six popular applications on the five smartphones with different file system utilizations. (In the rest of this section, we report the evaluation results on five representative smartphones, S5, G5, N5, N6 and S6, which were chosen from each utilization category.) The recurrence interval of file fragmentation was proportional to the file system utilization. For example, on the sixth day after the full file system defragmentation, the average DoF value of the `Twitter` files reached 1.56 and 2.7 for 70% and 90% of file system utilization, respectively.

Our observation strongly suggests that file fragmentation is a recurring problem in smartphones, especially when the file system utilization is high. (One of the reasons for a short recurrence interval is frequent app updates which automatically invoked in background when a smartphone is connected to a Wi-Fi environment. Since popular apps such as `Twitter` are reported to be updated, on average, every 7 days [2], when the file system utilization is high, newly installed apps are very likely to experience severe file fragmentation.) In the following sections, we shall show that file fragmentation negatively impact on user experience, but regular file defragmentation is harmful to flash storage lifetime. The proposed `janus` technique is novel in that these two conflicting phenomena are resolved in a satisfactory fashion[1].

5 Impact on User Experience

File fragmentation can negatively impact on the smartphone user experience due to degraded I/O performance. For example, the launching of an application involves reading a set of files, including executables, libraries, and data files. This procedure creates a user-perceived

[1]The impact of smaller I/O requests can be mitigated if command queueing is supported by an interface protocol for mobile storage because it allows multiple requests can be sent. However, since eMMC does not support command queueing yet [5], and the UFS protocol supports only a shallow queue (e.g., 16 entries), command queueing cannot hide the impact of smaller I/O requests on the I/O throughput.

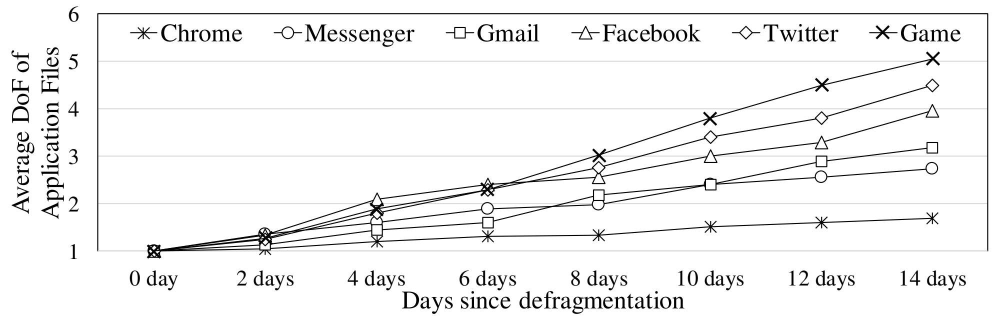


Fig. 2: The average DoF values of six application files on N6.

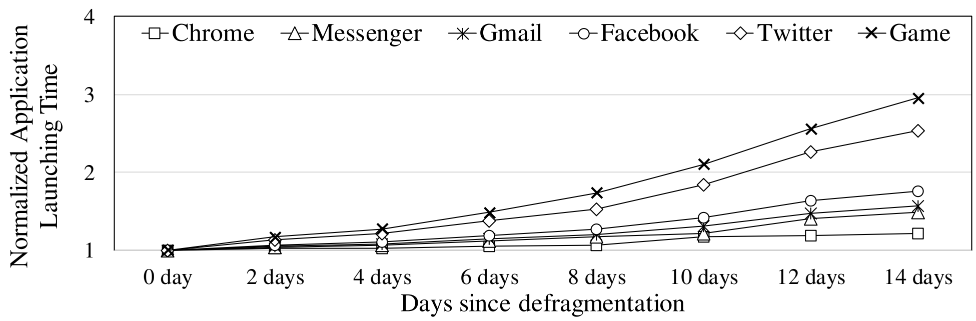


Fig. 3: Changes in six application's launching times on N6.

latency because the user has to wait until all the required files have been loaded from flash storage. We define the launching time of an application to be the time interval between the time when the application icon is touched and the time when all graphical user interface components are displayed for the next user interaction. (In order to minimize the impact of the network on app launching time, experiments were conducted with Wi-Fi connected. In addition, we removed all background applications before launching application.)

Fig. 3 shows the launching time of the six popular applications on N6 and Fig. 6 depicts the launching time of each six popular applications on five smartphones with different file system utilizations. The launching time noticeably degraded as the day count increased, especially with the high file system utilization. For example, compared to the launching time right after the full file system defragmentation, the launching time of `Twitter` on the sixth day was already 1.24 times longer when the file system utilization was 70%, and the launching time was amplified to 1.7 times longer when the file system utilization was 90%. This result indicates that the recurring file fragmentation can highly impact the quality of user experience in a short period of time.

6 Impact on Flash Memory Lifetime

Because file fragmentation is a recurring problem, regular file defragmentation might be necessary to maintain satisfiable user experience. In fact, weekly file defragmentation is recommended by many defragmentation tools [3, 4]. However, conventional file defragmentation is based on data copies, which increases the wear in flash memory. We performed full file system defragmentation with different frequencies, including a daily basis and a weekly basis, under the emulated application update behaviors. Fig. 4 shows the total write traffic contributed by file defragmentation measured by the built-in Linux kernel block I/O tracing tool `blktrace`. Surprisingly, the amount of data copies during file defragmentation was fairly large. For example, defragmenting files on the third day (after the last defragmentation) involved 1.8 GB of data copies under a 70% file system utilization, and this number increased to 5.76 GB if the file system utilization was 90%. If file defragmentation was performed in a weekly manner, the amount of data copies reached up to 9.53 GB.

The extra data copies negatively impacts on flash memory lifetime. This problem is further exaggerated by the deteriorated flash endurance due to the introduction of multilevel cells. Specifically, the program-erase cycle (PE cycle) limit of TLC NAND is as low as 300 PE cycles. The `data` partition of the S6 is 26 GB, and

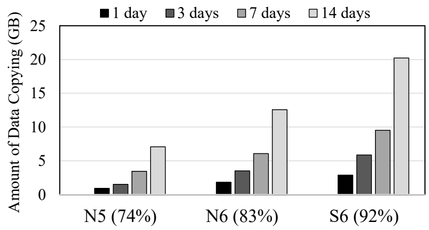


Fig. 4: The amount of data copies by file defragmentation with different defragmentation periods.

weekly file defragmentation costs every flash block (9.53 GB/week $\times$ 4 weeks)/26 GB$\approx$1.3 extra PE cycles per month. In the typical smartphone life cycle of two years, weekly file defragmentation introduces 31.2 extra PE cycles to every block, and thus the flash lifetime is degraded by more than 10%. This significant lifetime reduction highly discourages the use of conventional copy-based file defragmentation tools on flash storage.

References

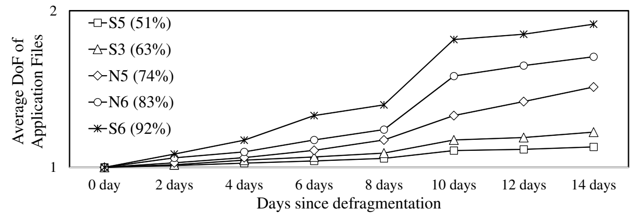
[1] JI, C., CHANG, L., SHI, L., WU, C., LI, Q., AND XUE, C. J. An Empirical Study of File-System Fragmentation in Mobile Storage Systems. In *Proceedings of the USENIX Workshop on Hot Topics in Storage and File Systems* (2016).

[2] Twitter Version History. `https://www.apk4fun.com/history/2699/`

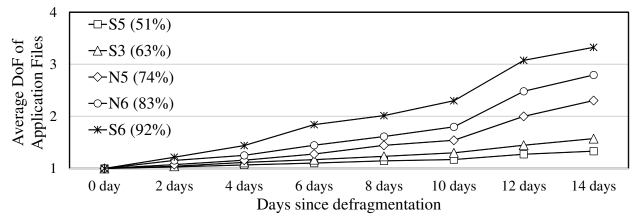
[3] KESSLER, M. Maintaining Windows 2000 Peak Performance Through Defragmentation. `https://msdn.microsoft.com/en-us/library/bb742585.aspx`

[4] SINOFSKY, S. Disk Defragmentation Background and Engineering the Windows 7 Improvements. `https://blogs.msdn.microsoft.com/e7/2009/01/25/disk-defragmentation-background-and-engineering-the-windows-7-improvements/`

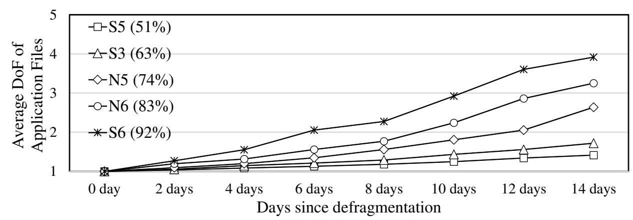
[5] Embedded MultiMediaCard (e.MMC). `http://www.jedec.org/standards-documents/technology-focus-areas/flash-memory-ssds-ufs-emmc/e-mmc`



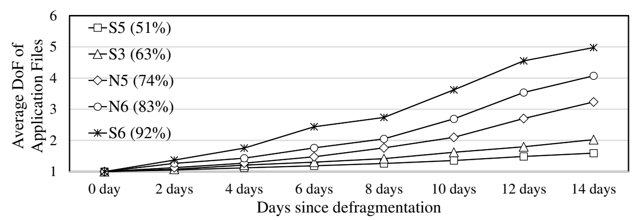
(a) `Chrome` on five smartphones.



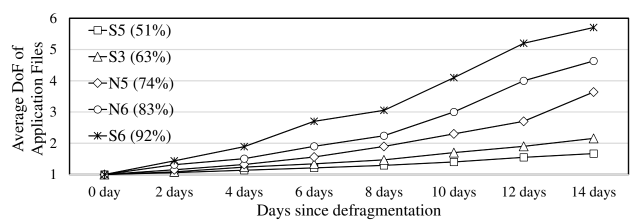
(b) `Messenger` on five smartphones.



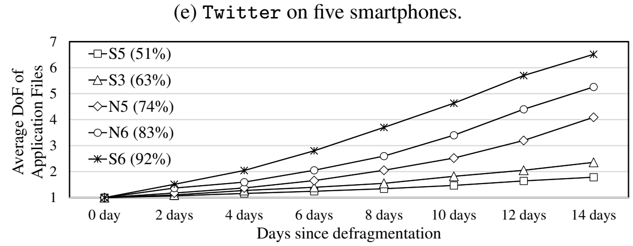
(c) `Gmail` on five smartphones.



(d) `Facebook` on five smartphones.

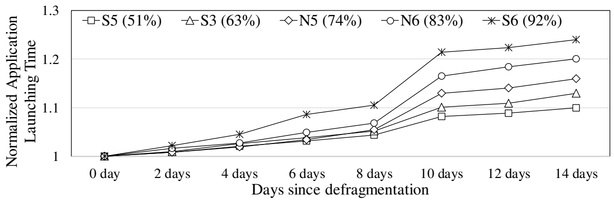


(e) `Twitter` on five smartphones.

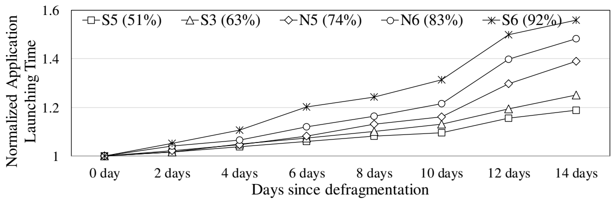


(f) `Game` on five smartphones.

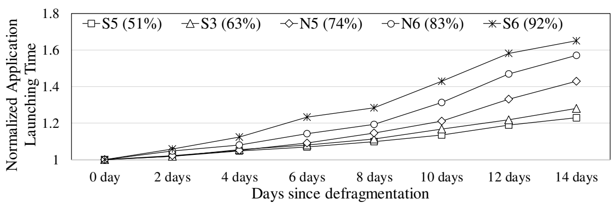
Fig. 5: The average DoF values of application files.



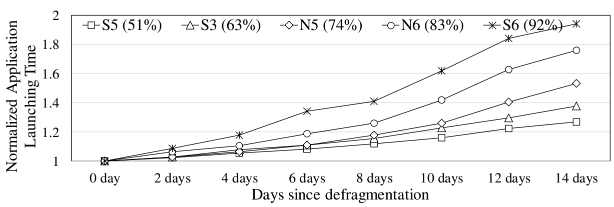
(a) `Chrome` on five smartphones.



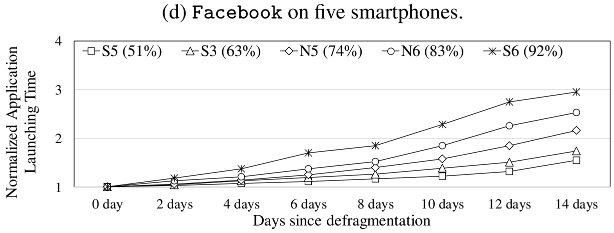
(b) `Messenger` on five smartphones.



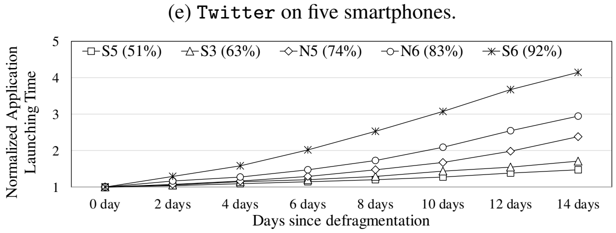
(c) `Gmail` on five smartphones.



(d) `Facebook` on five smartphones.



(e) `Twitter` on five smartphones.



(f) `Game` on five smartphones.

Fig. 6: Changes in app launching times.