An Analysis on Empirical Performance of SSD-Based RAID

Chanhyun Park, Seongjin Lee and Youjip Won

Abstract In this paper, we measure the I/O performance of five filesystems—EXT4, XFS, BTRFS, NILFS2, and F2FS, with five storage configurations—single SSD, RAID 0, RAID 1, RAID 10, and RAID 5. We show that F2FS on RAID 0 and RAID 5 with eight SSDs outperforms EXT4 by 5 times and 50 times, respectively. We also make a case that RAID controller can be a significant bottleneck in building a RAID system with high speed SSDs.

Keywords RAID · SSD · Disk I/O performance · Filesystem · F2FS

1 Introduction

A Solid State Drive (SSD) is a low power storage device with high I/O bandwidth that has received much attention as a device that may replace Hard Disk Drives (HDDs) and remove I/O performance bottleneck of a computer [1–5]. As the use of services that require reliability and rapid response expands, the demand for a device that meets stringent I/O performance requirements are also increasing [6, 7]. RAID (Redundant Array of Inexpensive Disks) [8] exploits slow storage device that is HDD to improve the I/O performance of a system. One of its strength is its customizability based on the required level of reliability and performance of a computing system.

Recent studies on RAID try to use SSD as alternative to HDD and explore performance of various RAID configuration using the SSD, such as effect of stripe size

C. Park (✉) · S. Lee · Y. Won
Department of Computer and Software, Hanyang University, Seoul, Korea
e-mail: parkch0708@hanyang.ac.kr

S. Lee
e-mail: insight@hanyang.ac.kr

Y. Won
e-mail: yjwon@hanyang.ac.kr

T. Czachórski et al. (eds.), *Information Sciences and Systems 2014*,
DOI 10.1007/978-3-319-09465-6_41

[9–11] or RAID level [3, 7, 11]. However, to the best of our knowledge, the research community has not thoroughly analyzed the effect of software such as filesystem layer in the I/O hierarchy for SSD-based RAID storage system. In this research, we establish a hypothesis that log-structured filesystem is more suitable than journaling filesystem or copy-on-write filesystem for SSD-based RAID. To prove the hypothesis, we measure the I/O performance of two journaling filesystems (EXT4 [12], XFS [13]), one copy-on-write filesystem (BTRFS [14]) and two log-structured filesystems (NILFS2 [15], F2FS [16]) on SSD-based RAID. In order to provide fair experiment conditions for each filesystems, we first obtain the optimal stripe size. After obtaining the optimal stripe size, RAID organization, and the number of disks from the experiment, we analyzed I/O performances on five filesystems (EXT4, XFS, BTRFS, NILFS2, and F2FS). This experiment showed that the selection of filesystem can draw a difference in performance for RAID organization by more than 50 times.

2 Related Work

In relation to the organization of RAID with existing SSDs, many studies have been conducted [1–3, 7, 9–11].

Some studies showed the impact of stripe size on the I/O performance of SSD-based RAID [9–11]. These researches showed a correlation between the stripe size and the I/O performance. One of these study [11] in which I/O performances were measured under various stripe sizes, 16 KB, 64 KB, and 1,024 KB, with record sizes from 8 KB to 256 KB. While changing stripe size, sequential read performance showed differences of more than 250 MB/s; sequential write performance showed differences of more than 300 MB/s; random read performance showed differences of more than 7000IOPS; and random write performance showed differences of more than 2000IOPS in a particular record size.

And there are some studies that analyze the changes in I/O performance when RAID levels and the number of SSDs organizing RAID are changed [3, 7, 11]. Among them, some studies confirmed that RAID organization with SSDs can make effects different from HDD on performance of RAID 5 [3]. During the write work, read work is added to RAID 5 due to the characteristics. In this research, when organizing RAID 5 with HDD, whose read performance and write performance are symmetrical, the performance is reduced more greatly than RAID 5 such as RAID0, RAID10, etc., because of the characteristics. On the contrary, if RAID 5 consists of SSDs, which performs better at reading than at writing, it shows no performance difference from other RAID levels, such as RAID 0 or RAID 10.

For the methods embodying RAID, there is a software RAID as an operating system which organizes and manages RAID, and a hardware RAID as separate equipment, RAID controller, which organizes and manages RAID. Among previous research, there were some studies which organized software RAID and measured the performance [7, 9]. One research [7] mentioned that maximum limitation of I/O

performance can be caused by bottleneck of RAID controller during the organization of hardware RAID with SSDs, and suggested the possibility of incongruity of hardware RAID organized by SSDs.

Much research has been conducted on SSD-based RAID but there was no proper research on the effects of software class on RAID organization. We considered filesystem one of the factors that must be considered during RAID organization.

3 Background

3.1 RAID

RAID combines several disks into one logical storage so as to use it as one disk with large capacity [8]. RAID divides the requested I/Os into certain size called stripe unit [17] or chunk, and distributes them in multiple disks. It is very important to select the optimal stripe size because stripe size cannot be changed dynamically [7].

RAID 0 distributes the data to number of disks used to organize the RAID system. If a large sequential I/O is issued, the RAID controller segments the I/O to a stripe size and writes them to disks in the RAID system in parallel. The process of dividing the data and distributing the stripes to number of disks is called striping. RAID 0 is the fastest RAID system because it maximizes parallelism, and it also affords the largest capacity among RAID systems; however, the integrity of the system breaks if a disk in the RAID 0 fails.

RAID 1 is often called as mirroring which requires two disks. Identical data is stored on each disk. Since same data is stored on mirrored disk, the system can withstand the failure in any of the disk. But, benefit of using RAID 1 comes in great cost, which limits the user space to 50 % of total storage capacity.

RAID 10 creates mirror of stripe sets that is applying RAID 0 on RAID 1; to create RAID 10 system, at least four disks are required—two for striping and the other two for mirroring of the stripe set.

RAID 5 exploits striping with distributed parity. In order to configure the system, it requires at least three disks, two for the stripe unit and one for the parity. Since it keeps a parity, the system can be recovered from failure in any one of the disk; however, the total storage space available reduces to store the parity.

3.2 Filesystem Synopsis

Journaling filesystem, such as EXT4 [12] or XFS [13], saves information of all write requests in the journal area with a particular form. There are three journal modes in journaling filesystem: journal, ordered, and write back mode. Journal mode saves both data and metadata in the journal area; ordered mode records changes in metadata

to the journal area only after data is written to the storage. Write back mode writes metadata in the journal area and keeps data on the main filesystem; however, write back mode disregards the ordering of the data. Note that journal area is placed in the middle LBA of a partition to minimize the distance of movement of HDD's arm. When the arm moves back and forth to record journal data and filesystem data, the locality of the filesystem data can be broken. One way to avoid the break of the locality is to use external journaling which exploits a separate disk as a medium for storing the journal data. This research uses ordered mode to examine effect of journal on RAID system.

BTRFS [14] is a copy-on-write (COW) filesystem which is introduced in Linux Kernel 2.6.29. It is also known as next generation filesystem that provides features such as built-in volume management, per-block checksumming, self-healing redundant arrays, and atomic copy-on-write snapshots. From the perspective of I/O count and I/O volume, copy-on-write feature can be an issue. In other filesystems that does not use copy-on-write, the filesystem overwrites the existing data on the storage, whereas the copy-on-write filesystem does not overwrite the existing data, instead it writes the new data to elsewhere. This research intends to examine the overhead of managing metadata and write requests when copy-on-write filesystem is applied to RAID and repeated updates must be treated.

NILFS2 [15] and F2FS [16] are log-structured filesystem which is merged to Linux mainline in Kernel 2.6.30 and Kernel 3.8, respectively. Log-structured filesystem appends all incoming data to the end of the log, which is in units of segment. Since all write requests are treated as sequential write operations, bandwidth of random writes on log-structured filesystem exhibits the same performance as sequential write; however, a critical problem with the log-structured filesystem is its read performance. Since all writes are written sequentially regardless of its spatial locality, all data must be read randomly. This research intends to examine the performance benefits of log-structured filesystem in SSD-based RAID.

4 Environment

The objective of this study is to analyze the behavior of SSD-based RAID under various RAID configurations, and examine the effect of stripe size and filesystem on storage performance. In this paper, we use five stripe sizes—64, 128, 256, 512, and 1,024 KB, and use five storage configuration—single SSD, RAID 0, RAID 1, RAID 10, and RAID 5. Workloads used in this paper are sequential read/write and random read/write, which are tested with buffered I/O and direct I/O.

We use MobiBench [18] to measure the I/O performance of the RAID system. We use 5 % of available filesystem partition as a file size throughout the experiment, and the I/O size for sequential and random I/O is set to 2 MB and 4 KB, respectively.

To measure the performance of the system, we used a computer that consists of eight 256 GB SATA3.0 Samsung SSD 840 Pro, connected to a Dell PERC H710P (1GB Cache memory, 2 SAS ports) on PCI-E 2.0 8 lane interface in a system with

16GB of memory (Samsung DDR3 SDRAM PC3 12800 4 GB × 4) and a Intel Core i7-3770 (4 cores with clock speed of 3.5 GHz). The system operates on Ubuntu 13.04 64 bit, Kernel 3.13. The maximum performance of a SSD 840 Pro 256 GB for sequential read/write is 540/520 MB/s and for random read/write is 100K/90KIOPS.

Theoretically, the maximum bandwidth of write operation on RAID 0 with N number of disks can be calculated as the number of disks times bandwidth of a device. Suppose a system exploits a SSD that has read and write bandwidth of 540 MB/s and 520 MB/s, respectively, the performance of a RAID 0 with 8 SSDs is 4,320 MB/s and 4,160 MB/s, respectively. The maximum bandwidth of H710P connected to PCI-E 2.0 × 8 lane can be calculated as bandwidth of PCI-E 2.0, 6 Gbit/s times 8 lanes, which is 4,096 MB/s. From the fact that the maximum interface bandwidth is lower than the maximum bandwidth of RAID 0 with 8 SSDs, we can deduce that the interface can be a source of the bottleneck on a RAID system with very high bandwidth.

5 Experiment

5.1 Effect of Stripe Size

We configure RAID 0 and RAID 5 with eight Samsung SSD 840 Pro, and vary the stripe unit size to examine the I/O performance. We measure sequential I/O performance (MB/s) as well as random I/O performance (KIOPS).

Figure 1 illustrates the result of I/O bandwidth of different stripe sizes ranging from 64 KB to 1,024 KB in multiples of two for RAID 0 and RAID 5 with eight SSDs. The performance of RAID 0 is shown in Fig. 1a, b. The best sequential I/O performance in RAID 0 is observed when the stripe size is 512 KB in both read and write, except for the case of sequential buffered read. For sequential read, buffered I/O and direct I/O yield 1,585 MB/s and 2,149 MB/s, respectively; on the other hand, sequential write operation with buffered I/O and direct I/O yield 1,774 MB/s and 2,535 MB/s, respectively. We observe that the memory copy overhead of buffered I/O brings about 25–30 % performance degradation in sequential read and write, respectively. In the case of random read, buffered I/O and direct I/O yield 8.3KIOPS and 8.4KIOPS, respectively. And, random write with buffered I/O and direct I/O yield 69.3KIOPS and 21.2KIOPS, respectively.

Performance on RAID 5 is shown in Fig. 1c, d. We observe that stripe size of 512 KB also shows best performance on RAID 5, except for random buffered write. For the sequential read with buffered I/O and direct I/O yield 1,633 MB/s and 2,089 MB/s, respectively. And, the sequential write with buffered I/O and direct I/O yield 1,859 MB/s and 2,097 MB/s, respectively. For the random read, on the other hand, buffered I/O and direct I/O yield 8.2KIOPS and 8.4KIOPS, respectively. For the random write, buffered I/O and direct I/O yield 3.9KIOPS and 10.8KIOPS, respectively. Stripe size of 64 KB on random buffered write, which exhibits the best performance in the test, shows throughput of 15.3KIOPS.

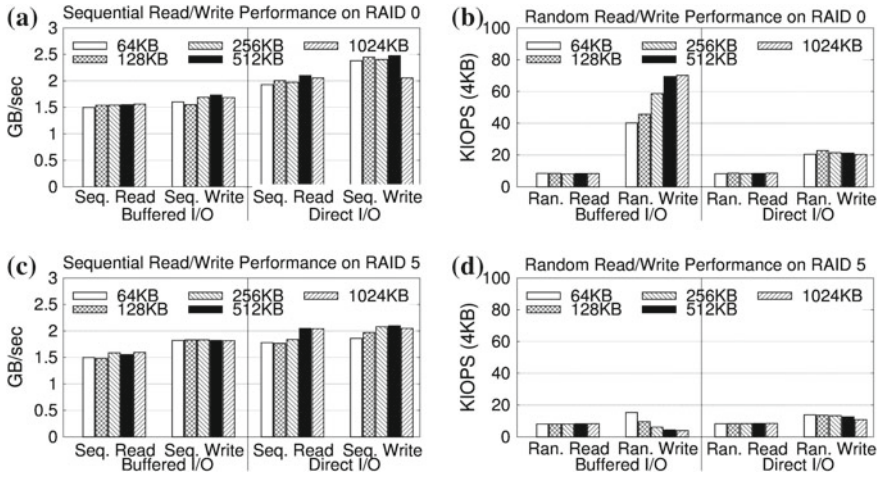

Fig. 1 **a** Seq. read/write performance on RAID 0. **b** Ran. read/write performance on RAID 0. **c** Seq. read/write performance on RAID 5. **d** Ran. read/write performance on RAID 5. I/O performance by stripe size on RAID 0 and RAID 5 (SSDx8, EXT4 filesystem)

We observe that stripe size of 512 KB exhibits either the best performance or equivalently good performance compared to the best case. Therefore, we set the optimal stripe size is 512 KB for sequential read/write and random read/write on both buffered I/O and direct I/O, except for random buffered write. The next set of experiments use stripe size of 512 KB.

5.2 Effect of RAID Level

Figure 2 illustrates the I/O performance with respect to RAID levels and the number of SSDs. Although it is not shown in the graph, we also measured the performance of RAID 1 and RAID 10. The performance of RAID 0 with one SSD and RAID 1 with two SSDs shows similar performance to sequential read/write performance with single SSD. Bandwidth of sequential write shows about 10 % lower performance than the performance of single SSD. RAID organization with additional number of disks shows better I/O performances than that of single SSD.

As the number of SSDs in RAID configuration increases, the I/O performance improves to a certain level. However, once the peak is reached, additional increase in the number of SSDs does not bring better performance. For sequential buffered read on RAID 0, the performance reached about 1,600 MB/s with three SSDs and stayed at that level even with more SSDs. In the case of sequential direct read, about 2,100 MB/s was reached with five SSDs. For sequential write on RAID 0 with four SSDs, buffered I/O shows about 1,800 MB/s and direct I/O with six SSDs shows about 2,500 MB/s. The result of RAID 0 experiment shows that the maximum

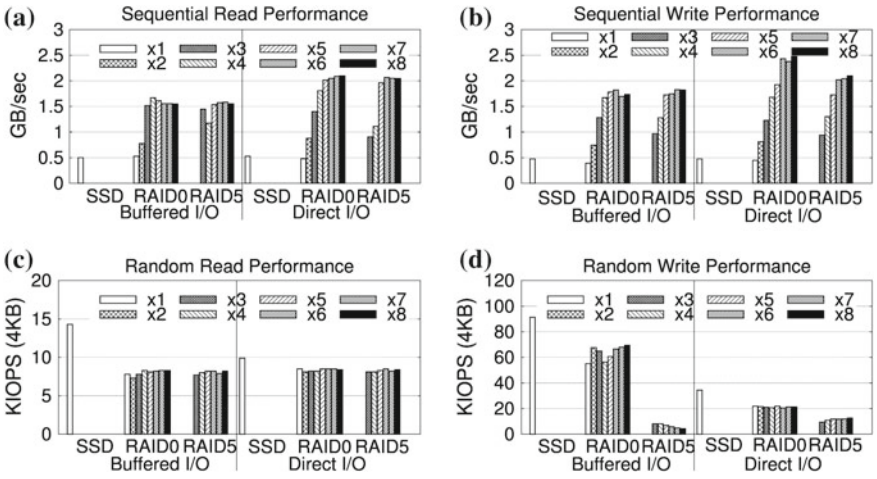

Fig. 2 **a** Sequential read. **b** Sequential write. **c** Random read. **d** Random write. I/O performance by RAID level and the number of SSDs (EXT4 filesystem)

performance of sequential read and write of RAID controller is about 2,100 MB/s and 2,500 MB/s, respectively. It is interesting to see that the bandwidth of sequential workload on RAID 5 is not far different from the performance of RAID 0. The performance result of changing the number of disks in RAID 0 and RAID 5 implies that the performance bottleneck lies in the RAID controller.

Random read and write workload on both RAID 0 and RAID 5 shows inferior performance compared to that of single SSD. In the case of random read performance, the number of disks in RAID configuration does not affect the performance. With single SSD, random read with buffered I/O and direct I/O shows 14.3KIOPS and 9.9KIOPS, respectively. We observe that the performance of RAID 0 and RAID 5 is almost the same regardless of the number of SSDs used; random buffered read exhibits minimum of 7.8KIOPS and maximum of 8.7KIOPS, and random direct read shows minimum of 8.1KIOPS and maximum of 8.5KIOPS. On all RAID configurations, random buffered read shows about 40–45 % lower performance compared to the performance of single SSD, and random direct read shows about 14–18 % lower performance to that of single SSD.

The performance of random write shows severe performance reduction in RAID 5. For random buffered write, RAID 0 shows about 20–40 % lower performance compared to single SSD; RAID 1 and RAID 10 shows performance reduction of about 55 %. The I/O performance of RAID 5 is reduced about 90–95 % compared to single SSD. In the case of random direct write on RAID 0, RAID 1, and RAID 10, about 38 % of I/O performance is reduced compared to that of single SSD; in the contrary, about 67 % of performance is reduced in RAID 5.

There are two interesting findings we can deduce from the result of experiment on the I/O performance with respect to RAID level and the number of SSDs in

RAID configuration. First, measured I/O performance does not match the theoretical performance measurements calculated with respect to RAID levels. In fact, our measurements show that the maximum I/O performance is lower than the theoretical measurements. We believe that the RAID controller is the main bottleneck in limited I/O performance. Second, we find that random read/write performance of RAID is much lower than that of single SSD—random buffered write on RAID 5 is about 95 % lower than the performance of single SSD. We believe random performance is slow because RAID cannot exploit parallelism while process the random I/O requests.

5.3 *Effect of Filesystem*

Figure 3 shows I/O performance of different filesystems on RAID 0 with eight SSDs. In the case of sequential read performance, both buffered I/O and direct I/O shows I/O performance of about 500 MB/s in all of five filesystems. It is interesting to see that sequential buffered read on BTRFS shows the best performance of about 2,500 MB/s, which is 160 % more than that of other filesystems on RAID 0. It shows that the performance of RAID 0 is about three times higher than that of single SSD. With direct I/O, the performance of RAID 0 is higher than that of single SSD by about 4.2 times in all filesystems except for BTRFS. The four filesystems, except BTRFS, show I/O performance of about 2,100 MB/s; BTRFS exhibits performance of about 1,700 MB/s.

Performance of sequential buffered write measured on single SSD shows I/O performance of about 500 MB/s in four filesystems except for NILFS2. In the case of

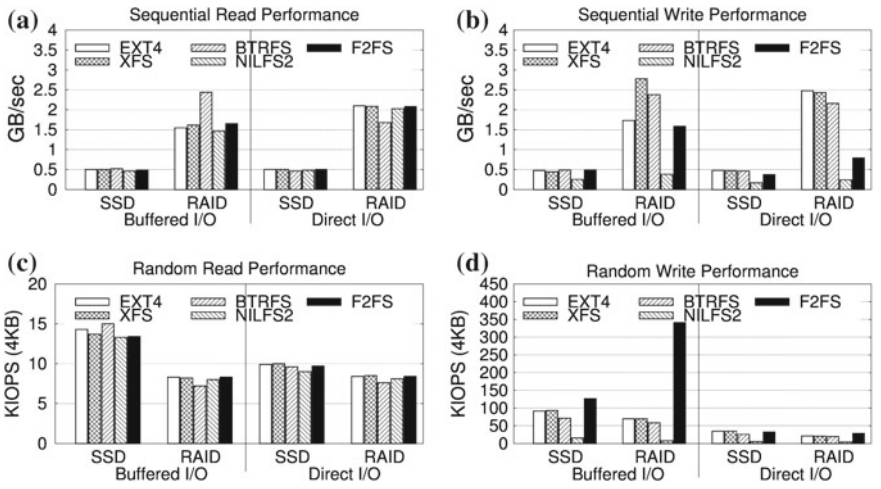

Fig. 3 **a** Sequential read. **b** Sequential write. **c** Random read. **d** Random write. I/O performance by filesystem (RAID 0, SSDx8)

direct I/O, all filesystems other than F2FS shows performance of 500 MB/s; performance of F2FS is about 380 MB/s. In the case of RAID, XFS has highest I/O performance of about 2,850 MB/s in buffered I/O, and EXT4 and XFS is the filesystem with highest I/O performance, producing performance of about 2,500 MB/s in direct I/O. BTRFS shows the second highest performance in both buffered I/O and direct I/O with 2,400 MB/s and 2,200 MB/s, respectively. In the case of F2FS, the performance of buffered I/O on RAID 0 is about 3.2 times better than that of single SSD, but the performance of direct I/O is 2.1 time better than that of single SSD.

For random read performance, the extent of I/O performance reduction is similar in all filesystems in comparison with single SSD. With RAID 0, random buffered read performance decreases by about 40 % compared to that of single SSD, and random direct read shows about 15 % drop in performance. In the case of random direct write, the performance of RAID is measured at about 60 % of single SSD in all filesystems except for NILFS2.

The performance of random buffered write shows the most interesting result. It shows that only F2FS on RAID 0 exceeds the performance of single SSD, whereas the performance of the other filesystems are lower than that of single SSD. The performance of the four other filesystems show about 20–50 % lower performance than the performance measured on single SSD. On the contrary, F2FS on RAID 0 shows 2.7 times better I/O performance than single SSD.

Figure 4 compares the I/O performance between EXT4 and F2FS in single SSD, RAID 0 (SSDx8) and RAID 5 (SSDx8). The result shows striking difference of performance on sequential direct write and random buffered write. In the case of sequential direct write, the performance of F2FS is lower than that of EXT4 on SSD, RAID 0, and RAID 5. Although sequential write with direct I/O of EXT4 on RAID 5 is about 20 % lower than that of RAID 0, the performance on RAID 0 and RAID 5 is about 3.1 and 2.6 times better than that of F2FS, respectively.

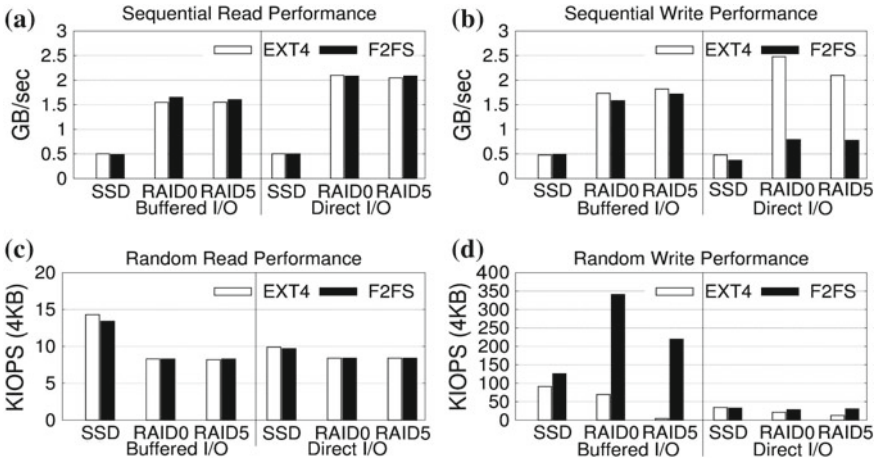

Fig. 4 **a** Sequential read. **b** Sequential write. **c** Random read. **d** Random write. I/O performance on EXT4 and F2FS on RAID 0 (SSDx8) and RAID 5 (SSDx8)

We observe that the performance of random write with buffered I/O on EXT4 does not excel as much as in the sequential write experiment. In fact, the performance of F2FS on RAID 0 and RAID 5 is about 5 times and 50 times better than that of EXT4, respectively. It is interesting to see that the performance of EXT4 on RAID 5 is very poor; the throughput of EXT4 on random write with buffered I/O shows about 40 times lower compared to the performance on single SSD. Although F2FS shows lower I/O performance on RAID 5 compared to that of RAID 0, it still shows about 1.7 times better I/O performance than performance of single SSD.

The result of this section conveys that the filesystem plays a key role in defining the performance of RAID with SSDs. It also shows insight on decision making for choosing right filesystem for different workloads. The most interesting result shown in the experiments is that F2FS is the choice for the random write with buffered I/O workload, where all other filesystems fail to exhibit better performance than single SSD.

6 Conclusion

In this paper, we used a DELL PERC H710P RAID controller and eight Samsung SSD 840 pro to measure the performance of sequential read/write and random read/write with buffered I/O and direct I/O on various RAID configurations. We find the optimal stripe size to conduct the experiment on given workload, which is found to be 512 KB in our experiments on RAID 0 and RAID 5 with eight SSDs. To analyze the effect of the number of SSDs on the RAID system, we varied the number of the SSDs, and find that the performance of sequential read/write is limited by the performance of RAID controller not by number of SSDs used in the RAID organization. After analyzing the effect of different filesystems on the RAID system, we find that F2FS, the log-structured filesystem, shows the best performance on random write with buffered I/O on RAID 0 and RAID 5 with eight SSDs. The performance of F2FS on random write with buffered I/O on RAID 0 and RAID 5 shows about 5 times and 50 times, respectively.

Acknowledgments This work is supported by IT R&D program MKE/KEIT (No. 10041608, Embedded System Software for New-memory based Smart Device), and supported by a grant from Samsung Electronics Co., Ltd.

References

1. M. Balakrishnan, A. Kadav, V. Prabhakaran, D. Malkhi, Differential raid: rethinking raid for ssd reliability. Trans. Storage **6**(2), 4:1–4:22 (2010)
2. H. Lee, K. Lee, S. Noh, Augmenting raid with an ssd for energy relief. in: *Proceedings of the 2008 Conference on Power Aware Computing and Systems*. pp. 12–12. HotPower'08, USENIX Association (2008)

3. S. Moon, S. Kim, S.W. Lee, Revisiting raid configuration on the flash ssd. Korean Soc. Internet Inf. **9**, 237–242 (2008)
4. D. Narayanan, E. Thereska, A. Donnelly, S. Elnikety, A. Rowstron, Migrating server storage to ssds: analysis of tradeoffs. in: *Proceedings of the 4th ACM European Conference on Computer Systems*. pp. 145–158. ACM (2009)
5. N. Nishikawa, M. Nakano, M. Kitsuregawa, Energy aware raid configuration for large storage systems. in: *Proceedings of the 2011 Green Computing Conference and Workshops*. pp. 1–5 (2011)
6. D. Iacono, M. Shirer, Enterprises look to accelerate applications with all-solid state storage arrays, according to idc's first all flash array forecast (2013), http://www.idc.com/getdoc.jsp?containerId=prUS24072313
7. N. Jeremic, G. Mühl, A. Busse, J. Richling, The pitfalls of deploying solid-state drive raids. in: *Proceedings of the 4th Annual International Conference on Systems and Storage.* pp. 14:1–14:13. SYSTOR '11, ACM (2011)
8. DA. Patterson, G. Gibson, RH. Katz, A case for redundant arrays of inexpensive disks (raid). in: *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*. pp. 109–116. ACM (1988)
9. J. He, J. Bennett, A. Snavely, Dash-io: an empirical study of flash-based io for hpc. in: *Proceedings of the 2010 TeraGrid Conference*. pp. 10:1–10:8. ACM (2010)
10. S. Lee, D. Shin, Performance analysis on ssd raid device. Korea Inf. Sci. Soc. **39**, 367–369 (2012)
11. I. Petrov, G. Almeida, A. Buchmann, U. Gräf, Building large storage based on flash disks. in: ADMS 2010 (2010)
12. A. Mathur, M. Cao, S. Bhattacharya, A. Dilger, A. Tomas, L. Vivier, The new ext4 filesystem: current status and future plans. in: Linux Symposium. vol. 2, pp. 21–33. Citeseer (2007)
13. Wang, R., Anderson, T.: xfs: a wide area mass storage file system. in: 4th Workstation Operating Systems. pp. 71–78 (1993)
14. O. Rodeh, J. Bacik, C. Mason, Btrfs: the linux b-tree filesystem. Trans. Storage **9**(3), 9:1–9:32 (2013)
15. R. Konishi, Y. Amagai, K. Sato, H. Hifumi, S. Kihara, S. Moriai, The linux implementation of a log-structured file system. SIGOPS Oper. Syst. Rev. **40**(3), 102–107 (2006)
16. J. Kim, F2fs (2012), http://www.kernel.org/doc/Documentation/filesystems/f2fs.txt
17. P.M. Chen, E.K. Lee, G.A. Gibson, R.H. Katz, D.A. Patterson, Raid: high-performance, reliable secondary storage. ACM Comput. Surv. **26**(2), 145–185 (1994)
18. S. Jeong, K. Lee, J. Hwang, S. Lee, Y. Won, Framework for analyzing android i/o stack behavior: from generating the workload to analyzing the trace. Future Internet **5**(4), 591–610 (2013)