# Comparative Performance Analysis of Magnetic Hard Disk and Solid-State Drives Using 64-bit btrfs Filesystem

Stefan Stojkov, Ina Masnikosa
School of electrical engineering, University of Belgrade
Mihajlo Pupin Institute, University of Belgrade
Belgrade, Serbia
stefan.stojkov@pupin.rs, ina.masnikosa@pupin.rs

Borislav Djordjevic
Mihajlo Pupin Institute, University of Belgrade
Belgrade, Serbia
bora@impcomputers.com

*Abstract*—**The main subject of this paper is the performance analysis of btrfs (B-tree filesystem) as a newly developed Linux filesystem, in combination with two types of storage devices: magnetic hard disk (HDD) and solid-state drive (SSD). Despite the extensive usage of the magnetic hard drives, solid-state drives tend to replace them as faster and more reliable devices. Different types of benchmark tests were performed in order to obtain representative results and see the difference in performance between these two storage units. The results clearly show the advantage of SSDs over HDDs while using modern btrfs filesystem.**

***Keywords-filesystem; btrfs; solid-state drive; magnetic hard disk drive; Linux; benchmark***

## I. INTRODUCTION

With the constant technological improvements, the necessity for better controlled and safer data storage also increases. Different filesystems are in use for controlling all kinds of data often stored on local storage devices. Filesystems depend on an operating system and represent logic for data management and organization. Ease of access to data, its reliability and time performance also depend on a storage media, which is used by the filesystem. Magnetic hard disk drive stands out as the most commonly used local storage unit, but the usage of solid-state drives steadily increases over the last couple of years due to their improved reliability and speed.

The focus of this paper is the performance characteristics of btrfs (B-tree filesystem) while utilizing magnetic hard disk drive or solid-state drive as a storage unit. Btrfs is a filesystem that works under Linux operating system. The main characteristic of this particular filesystem is that the processed data are always copied to a new location [1]. Solid-state drive works on a similar principle, since they copy data to a new location while updating them, introducing additional complexity. Nevertheless, the btrfs filesystem is supposed to be efficient in combination with solid-state drive. It would be interesting to test this hypothesis by comparing the results obtained using benchmarking tool on both drives. For the purpose of testing, *Bonnie++* [2] is used to acquire the results.

After the introductory section, brief overview of the Linux filesystems is given, with the emphasis on btrfs. Section III presents two types of storage units used as test devices. The following section includes experimental configuration, results and performance analysis of the chosen filesystem on both storage devices. Finally, conclusions are exposed in the last section.

## II. LINUX FILESYSTEMS

Linux is considered to be the most popular of all the UNIX-like operating systems. It supports a variety of filesystems [3], and some of them will be mentioned in this section. Filesystems may act differently in combination with different storage units. This comes from a fact that each of them contains unique properties which are more or less compatible with the particular storage device [4, 5]. The focus will be on btrfs filesystem which is one of the newly developed filesystems (5th generation).

Each new filesystem is created in order to deal with the limitations found in previous types of filesystems. For example, ext4 filesystem was introduced as an upgrade of ext3 by the means of scalability, performance, and reliability. This type of filesystem was included in Linux version 2.6.19, and since then it has become one of the most popular filesystems. It introduced *extents* – descriptors representing a range of contiguous physical blocks [6]. Another frequently used filesystem is xfs, which was developed in 1993, and ported to Linux kernel eight years later. It is a high performance journaling filesystem, which is divided into separate allocation groups, i.e. equally size chunks. Xfs consists of two B+ trees, one where regions of free space are ordered by size, and the other where the regions are ordered by their starting physical location on the block device [7]. Besides the ext family of filesystems and xfs, jfs and ReiserFS were also used in Linux distributions. Btrfs introduces new features in order to further improve scalability, reliability and performance in general.

### A. *Btrfs*

As mentioned before, btrfs filesystem represents one of the newly developed filesystems. The development was started in

2007, and it was intensified in the last couple of years. It was included in Linux version 2.6.29, in 2009.

This type of filesystem is based on *copy-on-write* (CoW) principle and it utilizes B+ trees as fundamental data structures. B+ tree maps index keys into internal nodes and the appropriate data is stored in the leaves. These trees are also used in xfs filesystem, but with one significant difference for btrfs. The processed data are always copied to a new location, thus in-place modification is avoided. This property gives btrfs filesystem an advantage in terms of crash recovery, since the data are written in a different location every time.

The main features of the btrfs filesystem are: dynamic *inode* allocation, writable snapshots, data checksum, fast file system checking, compression, defragmentation and mirroring [8].

## III. DISK DRIVE

### A. Magnetic Hard Disk Drive (HDD)

HDD permanently stores data using magnetic disks, called platters. These flat disks rotate and electromagnetic heads are used for reading previously stored data, or writing new ones [9, 10]. Data are organized in tracks (circular paths) which are further divided into sectors.

One of the key parameters regarding magnetic HDD performance is spindle speed. This term is used for physical rotational speed of the platters measured in RPM (revolutions per minute). It directly affects the average latency of the magnetic hard drive. The latency is defined as the time needed for the correct sector to position to the location of the heads. The average latency is calculated as half value of the "worst case" latency (full rotation). Besides latency, there are three more factors that affect the overall positioning performance: command overhead time, seek time and settle time. Command overhead time represents time required for the disk to start executing the command. Seek time usually refers to as the average time it takes for the head to move between two random tracks. This is the most common seek measurement, but two more types are used as well: *track-to-track* (seek time between two adjacent tracks) and *full stroke* (seek time for entire disk width). Settle time is the amount of time required for head stabilization before reading or writing begins. These four time intervals form total access time of the magnetic HDD. Command overhead time and settle time can be omitted, since they are small compared to seek time and rotational latency. Besides access time, two other factors influence total reading and writing performance: media speed and interface speed. Media speed is defined as the density of the track per time needed for one revolution. Basically, it is a rate at which the magnetic HDD reads data from the surface of the disk. Here, it is the same for reading and writing. Interface interval represents the time required to transfer data from the drive's controller to the host system.

Based on the previous discussion, the formula for the total time needed for reading or writing can be used:

$$T_{total} = T_{seek} + T_{rotational\_latency} + T_{media} + T_{interface} \quad (1)$$

where $T_{seek}$ refers to seek time, $T_{rotational\_latency}$ represents the average rotational latency of the magnetic HDD, whereas $T_{media}$ is the time needed for reading data from the surface of the media or to write data to the media, and $T_{interface}$ represents the amount of time for the data transfer to host system.

### B. Solid-State Drive (SSD)

In contrary to magnetic HDD, SSD permanently stores data using flash memory chips. This gives them advantage over magnetic hard drive disks regarding read/write speed since there is no need for any conversion (the information is stored in electronic form). The usage of flash memory made SSD the first competitor to magnetic disk storage [11]. In addition, SSD does not contain movable mechanical parts. Therefore, almost instantaneous access to data is possible.

The main components of SSD are: flash memory and controller. Flash memory uses NAND technology [12] which is characterized by short erasing and programming times. There are two different types of flash memory: SLC (*Single Level Cell*) and MLC (*Multi Level Cell*). As the name suggests, the difference between these two types is the number of bit values stored in one cell [13].

As mentioned before, SSD contains a controller unit, which is the fastest part of the drive. Controller groups flash memories into channels. If the buffer is present, controller is connected to the bus through buffer. Otherwise, they are connected directly.

SSD storage is divided into blocks (typical block size is 512 KB). Blocks are further divided into pages (4 KB), with 128 pages in each block. Having no mechanical delays, SSD has the advantage over HDD regarding total read/write time:

$$T_{total} = T_{media} + T_{interface} \quad (2)$$

Here, $T_{media}$ and $T_{interface}$ have the same meaning as in (1), with the difference that data are now read from or written to flash memory.

As stated before, the time required for reading and the time required for writing are basically the same when magnetic HDD is utilized. On the other hand, media rate differs for reading and writing when solid-state drive is used. In the case of reading, the time is calculated by:

$$T_{media} = T_{page\_reading} \quad (3)$$

where $T_{page\_reading}$ refers to the time needed for reading pages.

Writing process is far more complex. Pages cannot be overwritten, i.e. they need to be empty. Also, only the entire block can be deleted, i.e. it is not possible to erase pages separately. Therefore, when existing page needs to be updated, the content of the entire block is copied into a new location. Then, the block is erased, and the content of the old block, as well as the data of the updated page are written to the new block. Total time needed for writing can be expressed with:

$$T_{media} = T_{gar\_coll} + T_{block\_erasing} + T_{page\_writing} \quad (4)$$

where $T_{gar\_coll}$ represents the time needed for the garbage collection, $T_{block\_erasing}$ is the time required for the deletion of the entire block, and $T_{page\_writing}$ refers to the amount of time

required for the updating of the pages. Garbage collection refers to as process of removing the blocks of data which are not needed anymore before copying the valid ones to a new location.

Btrfs filesystem introduces additional complexity because of the CoW method, which updates data by placing them into a new location. Therefore, garbage collection processing is more time consuming.

## IV. Experimental Results and Analysis

### A. Hardware Configuration

The specification of used hardware is presented in table I. Tests were performed on CentOS Linux operating system using both magnetic HDD and SSD. The specification of HDD [14] and SSD [15] drives are shown in tables II and III, respectively.

### B. Filesystem Organization

Keeping in mind the capacities of the chosen storage devices, the operating system was installed on magnetic hard disk drive. Complete filesystem organization is given in table IV. Notice that partitions reserved for testing filesystem were chosen to be the same size, marked as /dev/sda4 for magnetic HDD testing, and /dev/sdb1 for SSD tests (120 GB each).

TABLE I. Hardware Specification

| Hardware | Specification |
|---|---|
| RAM | 8 GB |
| CPU Model | Intel(R) Core(TM) i5-4690 CPU @ 3.50GHz |
| Number of CPU Cores | 4 |
| Magnetic Hard Disk Drive | Toshiba DT01ACA050, 500GB, 3.5" |
| Solid-State Drive | Transcend, TS128GSSD370S, 128GB, 2.5" |
| Operating System | CentOS Linux 7.0.1406,<br>kernel – Linux 3.10.0-123.el7.x86_64 |

TABLE II. Magnetic Hard Disk Drive Specification

| Magnetic HDD | Specification |
|---|---|
| Model | Toshiba DT01ACA050, 500GB, 3.5" |
| Capacity | 500 GB |
| Interface | Serial ATA 3.0 / ATA-8 |
| Transfer Rate to Host | 6 Gb/s |
| Average Latency | 4.17 ms |
| Average Seek Time (read) | 0.6 ms |
| Average Seek Time (write) | 0.8 ms |
| Rotational Speed | 7,200 RPM |

TABLE III. Solid-State Drive Specification

| SSD | Specification |
|---|---|
| Model | Transcend, TS128GSSD370S, 128GB, 2.5" |
| Capacity | 128 GB |
| Interface | Serial ATA III |
| Transfer Rate to Host | 6 Gb/s |
| Storage Media | Synchronous MLC NAND Flash memory |
| Controller | Transcend TS6500 |
| Buffer | None |
| Max. Read | 550 MB/s |
| Max. Write | 170 MB/s |

TABLE IV. Filesystem Organization

| Device | Filesystem organization | |
|---|---|---|
| | Size | Partition |
| /dev/sda1 | 500 MB | /boot |
| /dev/sda2 | 10 GB | /swap |
| /dev/sda3 | 300 GB | /root |
| /dev/sda4 | 120 GB | /hdd |
| /dev/sdb1 | 120 GB | /ssd |

### C. Benchmark Tool

As stated in the Introduction, Bonnie++ software is used as benchmark tool. It is C++ software which works under UNIX-like operating systems. This tool provides means for evaluating performance of different filesystems and storage units.

### D. Results

The results are obtained using Bonnie++ benchmark program and they can be divided into two groups: random (Fig. 1) and sequential (Fig. 2) performance analysis. The first group is realized using *putchar()* and *getchar()* functions, while the other is realized using *getblk()* function. Fig. 1a depicts the random write test results (*putchar()* function) and Fig. 1b shows the random read test results (*getchar()* function). SSD outperforms magnetic HDD by 10% in random write test and 25% in random read test. The results correspond to formula (2). According to formula (2), SSD has an advantage over magnetic HDD, and the difference is clearly seen in the case of random read performance. Regarding SSD random write, performance decreases due to additional time spent on garbage collection and block erasing in accordance with formula (4).

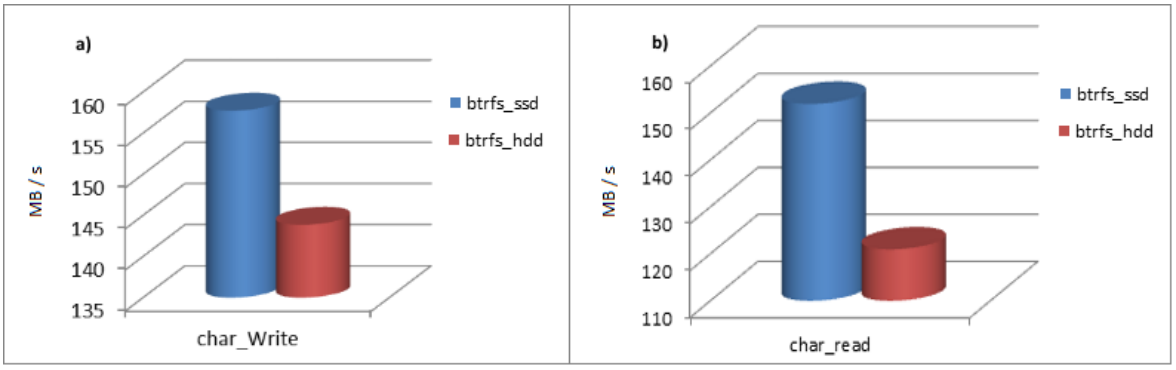


Fig. 1 – Random performance testing: a) writing; b) reading.

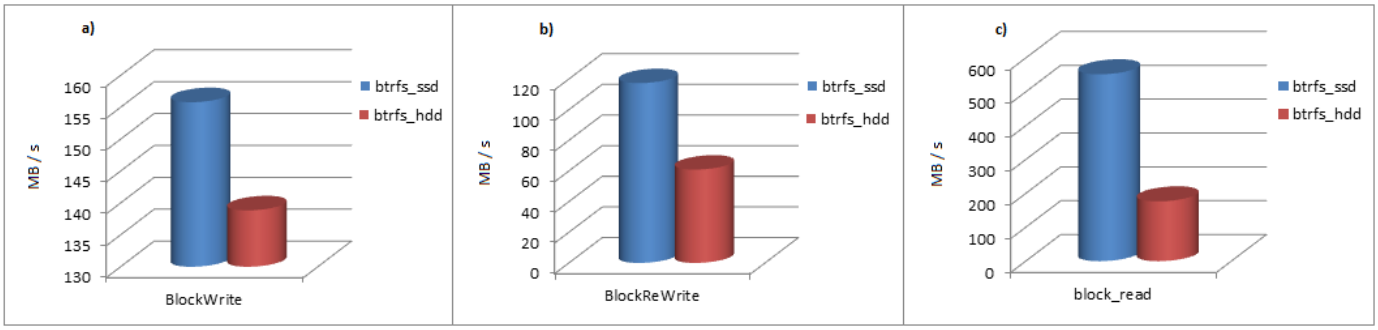


Fig. 2 – Sequential performance testing: a) writing; b) read-modify-write; c) reading.

Fig. 2 represents sequential performance of the drives, obtained using *getblock()* function. Again, SSD outperforms magnetic HDD: by 10% in the case of sequential writing, twice as fast as magnetic HDD in terms of read-modify-write performance, and three times faster than magnetic HDD in sequential reading. This group of results also fits into the given mathematical model. According to formula (2), SSD is significantly better than magnetic HDD in relation to formula (1), and the difference is not significant only for sequential writing. The reason for this lies in the fact that the large amount of data was prepared for writing, and the SSD performance decreased according to formula (4).

## V. CONCLUSION

The purpose of this paper was to compare the performance of magnetic hard disk and solid-state drives using modern 64-bit btrfs filesystem. The hypothesis was that SSD would be significantly faster, as it does not contain moving parts that introduce additional delays in form of seek time and rotational latency. It was also expected that the writing performance of the SSD would be weakened due to block erasing introduced by SSD and the garbage collection property of the btrfs filesystem. Tests confirmed these hypotheses, as well as the model given with formulas (1-4). SSD was three times as fast as magnetic HDD in the case of sequential read performance, and by about 30% better for random read tests. The difference significantly decreased when comparing the writing performance of the drives. This is especially the case for sequential writing, where large amount of data needs to be written, which results in block erasing and garbage collection overheads.

## REFERENCES


[1] O. Rodeh, J. Bacik, C. Mason "BTRFS: The Linux B-tree Filesystem," IBM Research Report, 9 July, 2012.

[2] Bonnie++ Benchmark Suite, http://www.coker.com.au/bonnie++/.

[3] L. Lu, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau, S. Lu. "A Study of Linux File System Evolution," in Proceedings of the 11th USENIX Conference on File and Storage Technologies, FAST'13, pp. 31–44, 2013.

[4] Linux 3.19 File-System Tests Of EXT4, Btrfs, XFS & F2FS, http://www.phoronix.com/scan.php?page=article&item=linux-3.19-ssd-fs&num=1.

[5] Linux 4.0 Hard Drive Comparison With EXT4/Btrfs/XFS/NTFS/NILFS2/ReiserFS, http://www.phoronix.com/scan.php?page=article&item=linux-40-hdd&num=1.

[6] A. Mathur, M. Cao, S. Bhattacharya, A. Dilger, A. Tomas, L. Vivier, "The New ext4 Filesystem: Current Status and Future Plans," in Proceedings of the Linux Symposium, Ottawa, Canada, June 2007.

[7] D. Robbins, "Common threads: Advanced filesystem implementor's guide," Part 9, http://www.ibm.com/developerworks/library/l-fs9/.

[8] B. Joksimoski, S. Loskovska, "Overview of Modern File Systems," The 7th International Conference for Informatics and Information Technology, CIIT, 2010.

[9] E. Grochowski, "Emerging Trends in Data Storage on Magnetic Hard Disk Drives," Datatech, pp. 11-16, 1998.

[10] I.R. McFadyen, E.E. Fullerton, M.J. Carey, "State-of-the-art Magnetic Hard Disk Drives," MRS Bulletin, Vol. 31, Issue 5, pp. 379-383, May 2006.

[11] S. Boboila, P. Desnoyers, "Write Endurance in Flash Drives: Measurements and Analysis," in Proceedings of FAST'10, San Jose, 2010.

[12] P. Desnoyers, "Empirical Evaluation of NAND Flash Memory Performance," in First Workshop on Hot Topics in Storage and File Systems (HotStorage'09), 2009.

[13] V. Timčenko, B. Đorđević, S. Obradović, N. Čorni, "Uticaj disk keš bafera na performanse SSD diskova," Infoteh-Jahorina Vol. 12, Jahorina, March 2013.

[14] Toshiba DT01ACA050 Datasheet, http://storage.toshiba.com/docs/product-datasheets/dt01aca.pdf.

[15] Transcend TS128GSSD370S_V10_Datasheet, http://www.transcend-info.com/products/images/modelpic/579/No3118_TSXGSSD370S_V10_Datasheet.pdf.