INFORMATION
SYSTEMS IN
MANAGEMENT

Information Systems in Management (2014) Vol. 3 (4) 273–283

MODERN FILESYSTEM PERFORMANCE IN LOCAL MULTI-DISK STORAGE SPACE CONFIGURATION

MATEUSZ SMOLIŃSKI

Institute of Information Technology, Technical University of Lodz

This paper includes analysis of modern filesystems performance in multi-disk storage space configuration. In performance testing only popular open source filesystem types were used in GNU/Linux operating system: BTRFS, EXT4, XFS. Base file operations were tested in various local multi-disk storage configurations using Logical Volume Manager, that differentiated due to disk number, allocation policy and block allocation unit. In multi-disk storage configurations managed by LVM many allocation policies were used like various RAID levels and thin provisioning. The obtained filesystem performance characteristics allow to choose parameters of multi-disk storage space configuration, which provides the best performance for file operations. Research result show also which filesystem type is the efficient in storage space configuration with locally connected disks.

Keywords: Multi-disk storage configuration, filesystem performance, logical volumes, disk managers, data allocation policy

1. Introduction

In operating system local storage configuration has significant impact on file operations performance. For security and reliability reasons file storage space is divided into zones. Storage zone configuration should be accordant to stored data characteristic. Simple zone of data storage configuration includes filesystem and single block device. Block devices can be locally connected or remote accessed by storage area network. Advanced storage zone configuration can use many block devices. Parallel I/O disk operation can provide better performance of filesystem that uses multi-disk storage space configuration. Disk manager provides data distribution between disks according to multi-disk volume configuration and can be software or hardware implemented. In advanced storage zone configuration managed volume has fixed number of disks, allocation policy and base block unit size. Volume policy sets the block allocation algorithm, which is responsible for block addressing and its localization on disks. In volume configuration any disk I/O operation is performed on data chunk, that sets allocation unit size for volume allocation policy. Operating systems can support many block device managers and filesystem types. Therefore multi-disk storage zone configuration can offer various file operation performance. Selection of the zone storage configuration is more difficult, because software disk manager can offer various allocation policies.

In all realized multi-disk storage zone configuration Logical Volume Manager was used as external disk manager, independent on used filesystem type. LVM is supported by Linux operating system and bases on Device Mapper kernel implementation. Logical volume created by LVM can distribute data between disks configured as physical volumes. Each logical volume has allocation policy defined as Device Mapper target, that is a kernel module with allocation algorithm implementation. Among other algorithms, LVM supports level 0, 5, 6 of RAID (Redundant Array of Independent Disks), that base on data striping where next data chunk are stored on separate disks [1].

Second main element of multi-disk storage zone configuration was filesystem. Filesystem types differ in physical and logical layer. Filesystem physical layer defines structures to data and metadata localization on block device. Files namespace and attributes are specified by logical filesystem layer, which main role is data organization (i.e. naming space, directories). For research purposes three filesystem types were chosen: BTRFS, EXT4 and XFS. Many performance testing include filesystem type comparison in simple storage zone configurations [3].

File operation performance tests in advanced storage zone configuration requires a specifiation of multi-disk storage zone scenarios and mearurement method. For the analysis and comparisons of the file operation performance a uniform environment is necessary for multi-disk storage space configuation scenarios.

2. Scenarios of multi-disk storage configuration

In multi-disk storage zone performance testing each configuration scenario has fixed number of disks, allocation policy and chunk size. All multi-disk configurations are created with LVM software in 2.02.106 version and default 4MB extent size. Environment equipment limits the range of disk number up to 5 locally connected hard drives. In case of scenario with RAID5 or RAID6 allocation policy occurs a additional disk synchronization phase. Size of chunk in storage space configuration with striping was selected from set: 8KB, 32KB, 128KB, 512KB.

Additionally LVM supports thin provisioning, in which block allocation is delayed to its first access. Thin provisioning also changes block addressing order, next blocks don't have to be localized contiguously in physical block device. Created thin provisioned volume does not require full coverage in the available block device storage space according to volume size. Thin provisioning has its own allocation unit, which defines block allocation form used storage pool.

Before each performance test run a new multi-disk storage configuration was created and synchronized, then one of filesystem type was created and mounted in selected directory. Created filesystems always were mounted in read-write and asynchronous mode. Identical mounting options for filesystems unifies file access. When filesystem performance test was completed the filesystem was unmonted and next filesystem type was formatted in multi-disk device created for storage scenario.

3. Uniform environment for storage scenarios testing

All multi-disk storage space configuration scenarios for filesystem performance testing was created in a uniform environment. The environment includes hardware and software elements. Computer hardware used for file operation performance testing was equipped with CPU i5-2400 3.10GHz, 8GB RAM and five identical SATA3 Western Digital disks, model WD5000AZRX with 64MB cache. All files of Linux operating system from Fedora 20 distribution were installed on external USB disk (used Linux kernel version: kernel-3.17.2-200.fc20.x86_64). In Linux operating system all disks have configured default CFQ elevator algorithm. In system additional packages are installed: btrfs-progs in version 3.17-1, bonnie++ in version 1.96-6, e2fsprogs in version 1.42.8, xfsprogs in version 3.2.1. Installed packages include used filesystem tools, and filesystem performance benchmark.

The Bonnie++ program tool was used for filesystems benchmarking in multi-disk storage configuration scenarios. Benchmarking program was configured to generate workload, that includes sequential and random creation of 1024 regular files in each of 1024 directories, write and read of 16GB data from regular file. Single run of benchmarking program provides statistic per second for every type of performed file operation: number of created and deleted regular files, number of files that attributes were read and sequential regular file data write and read rate.

4. Filesystem types

The filesystems performance testing was realized only for BTRFS, EXT4 and XFS filesystem types, each one is supported by Linux kernel via virtual filesystem interface implementation. All filesystems were created with 4KB allocation unit. Other filesystem parameters were set according to default configuration convention. Chosen filesystems types for performance testing are often used in various server configurations operated by Linux. Each of them has other design, especially different data structures.

BTRFS is a transactional filesystem, with bases on binary tree structures and write on copy rule update method. BTRFS does not have journal and uses 64 bit addressing. It has internal block device manager with own allocation policies separately for data and metadata, but in BTRFS performance testing only external disk manager software were used. BTRFS supports subvolumes, snapshoting and file structures cloning. If has also online defragmentation and resizing capability. This filesystem supports data checksumming and recovery according to used internal allocation policy [5]. For storage performance testing purposes BTRFS was configured with the same binary tree node and leaf sizes as 4KB block size.

EXT4 is a journaled filesystem with many improvements, that increase performance in comparison to earlier versions. Implemented in EXT4 extent feature provide continuous block allocation and pre-allocation of storage space for file. Actually EXT4 uses 48 bit addressing and journal checksumming [2]. All EXT4 metadata structures are prepared in creation time, therefore this filesystem limits number of stored files.

XFS is popular journaled filesystem with 64 bit addressing. Like EXT4 a XFS limits file fragmentation using separate allocation resource groups. Part of its internal structures are binary tree as BTRFS, it uses also delayed and sequential block allocation with various size of extent [4].

5. Filesystem performance in multi-disk storage scenarios

Performance of each filesystem was tested in single disk storage configuration and obtained results are a reference point to results of filesystem performance in multi-disk storage configuration scenarios. Figure 1 presents filesystem performance depending on number of disk managed by LVM with RAID0 striping policy with chunk size 128KB. In this multi-disk storage configuration increasing number of disks provides higher file data read and write speed. However number of disks does not have impact on base file operations like file creation or stat performed sequentially or random on files (fig. 1, 2).

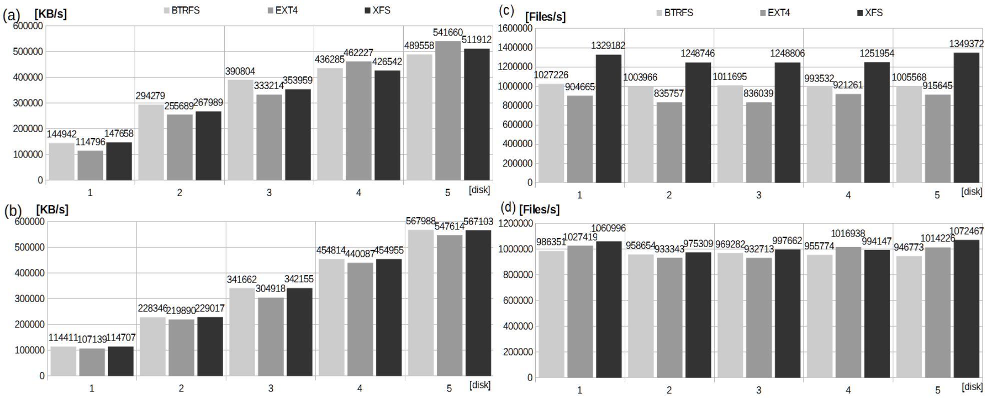

Figure 1. Filesystems performance depending on number of disks managed by LVM with RAID0 allocation policy and 128KB chunk size: (a) data read speed from regular file, (b) data write speed to regular file, (c) sequential file stat operations, (d) random file stat operations

Significant differences in file deletion efficiency are present according to number of disk. For BTRFS and XFS localized in multi-disk volume managed by LVM with RAID0 allocation policy the number of random deleted files is lower than for single disk storage.

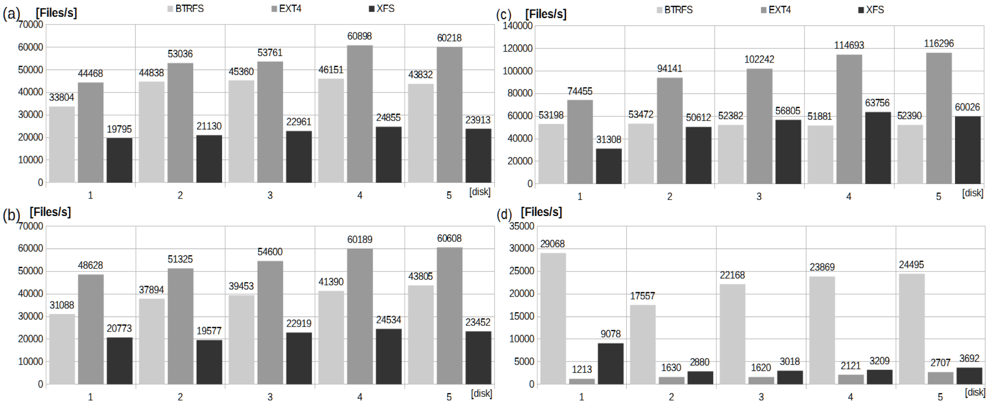

Figure 2. Filesystems performance depending on number of disks managed by LVM with RAID0 allocation policy and 128KB chunk size: number of (a) sequentially created files, (b) random created files, (c) sequentially deleted files, (d) random deleted files

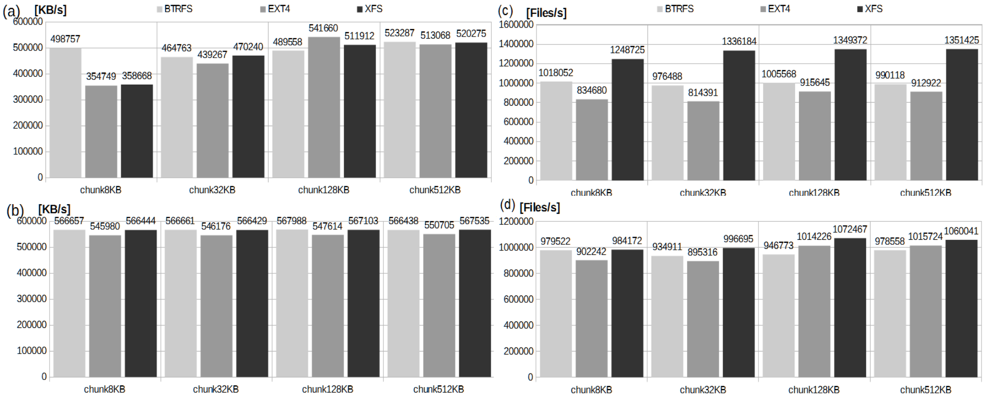

Figure 3. Filesystems performance depending on chunk size of RAID0 allocation policy for five disk volume managed by LVM: (a) data read speed from regular file, (b) data write speed to regular file, (c) sequential file stat operations, (d) random file stat operations

Further filesystems performance is presented according to chunk size in RAID0 striping policy shows that for all tested filesystems chunk size impact more on data read speed form file than data write to file. Chunk size has minimal impact on base file operation performance, whether are realized sequentially or random in volume storage space (fig. 3, 4).

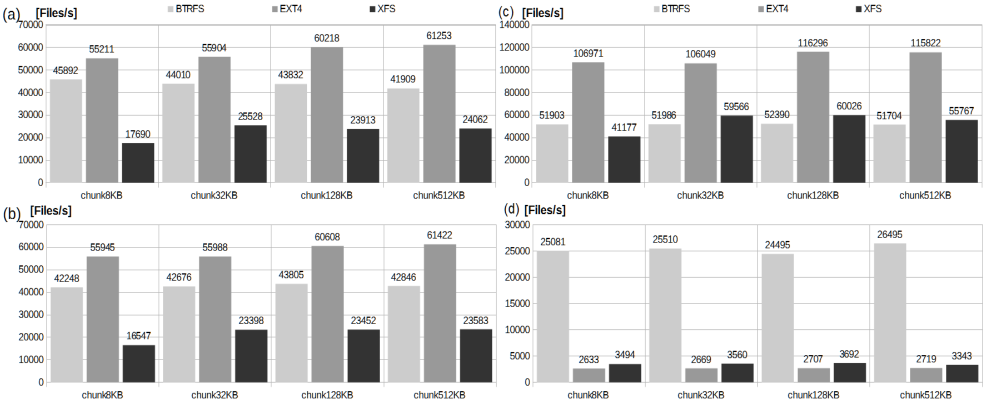

Figure 4. Filesystems performance depending on chunk size of RAID0 allocation policy for five disks volume managed by LVM: number of (a) sequentially created files, (b) random created files , (c) sequentially deleted files, (d) random deleted files

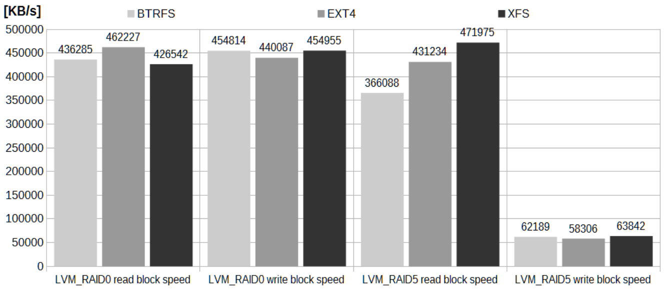

Figure 5. File data read and write speed comparison in LVM storage scenarios: RAID0 allocation policy with 4 disks and RAID5 allocation policy with 5 disks with 128KB chunk size

The comparison of filesystem performance between multi-disk storage space configuration managed by LVM with RAID0 allocation policy with 4 disks and RAID5 allocation policy with 5 disks shows drop in filesystem performance, in example for data write to file speed up to 86% performance decrease for all filesystems. This comparison shows also that drop of base file operations performance is present for tested filesystems (fig 5, 6).

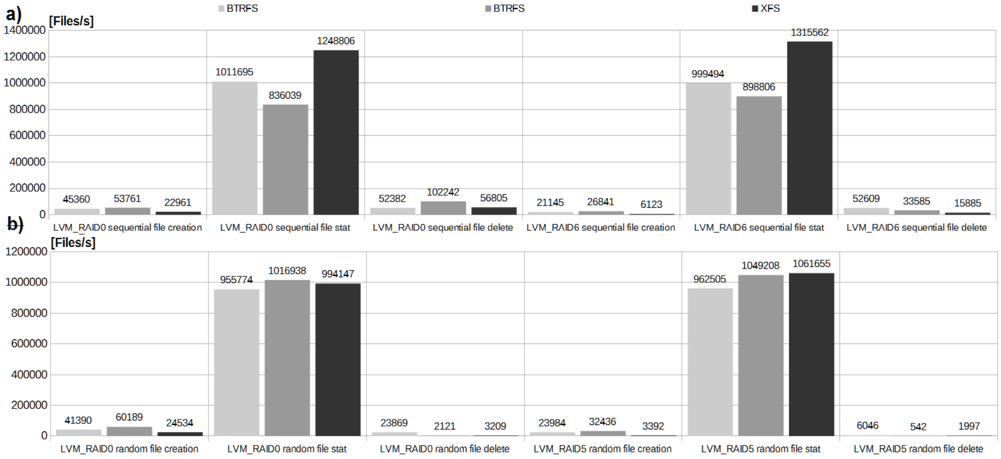

Figure 6. Filesystems performance comparison in LVM storage scenarios: RAID0 allocation policy with 4 disks and RAID5 allocation policy with 5 disks with 128KB chunk size (a) number of sequential file operations, (b) random file operations

Decreasing filesystem performance was also observed in comparison between RAID0 allocation policy with 3 disks and RAID6 allocation policy with 5 disks (fig. 7, 8).

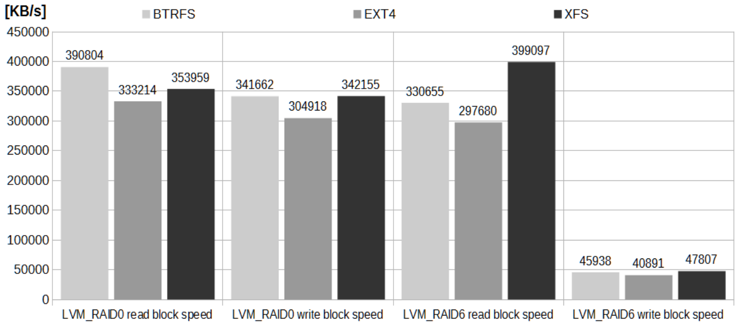

Figure 7. File data read and write speed comparison in LVM storage scenarios: RAID0 allocation policy with 3 disks and RAID6 allocation policy with 5 disks with 128KB chunk size

Filesystems performance characteristic also changes when it is localized on thin provisioned volume, especially for XFS. Using the same 128KB allocation unit size and RAID0 policy with 5 disks the XFS has the 67% drop in sequentially performed stat operation on regular file while random files deletion is performed over 8 times faster. Using thin provisioned volume in multi-disk storage configuration filesystem performance characteristic can change significantly (fig. 9, 10).

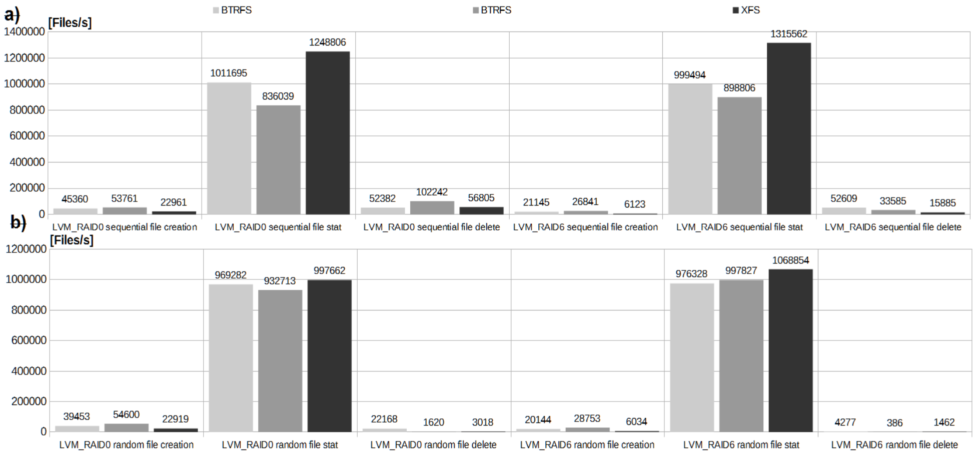

Figure 8. Filesystems performance comparison in LVM storage scenarios: RAID0 allocation policy with 3 disks and RAID6 allocation policy with 5 disks with 128KB chunk size: number of (a) sequential file operations, (b) random file operations

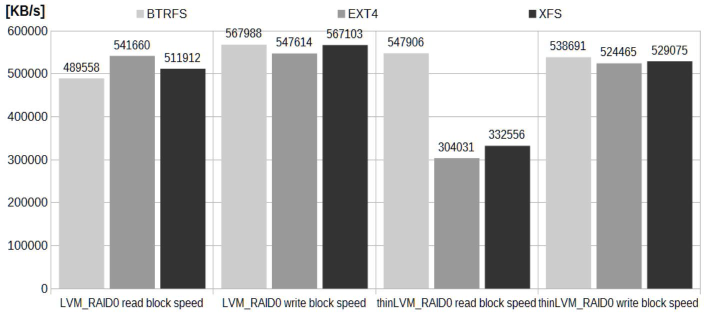

Figure 9. File data read and write speed comparison in multi-disk storage scenarios with RAID0 allocation policy with 5 disks and 128 KB chunk size for standard and thin provisioned logical volume

Important parameter in thin provisioned volume is a allocation unit size used when blocks are set from thin provisioned pool. This parameter has been configured in multi-disk storage configuration scenarios with thin provisioned volume in range: from 64KB to 8MB. Impact of pool allocation unit size in read and write file speed shows figure 11.

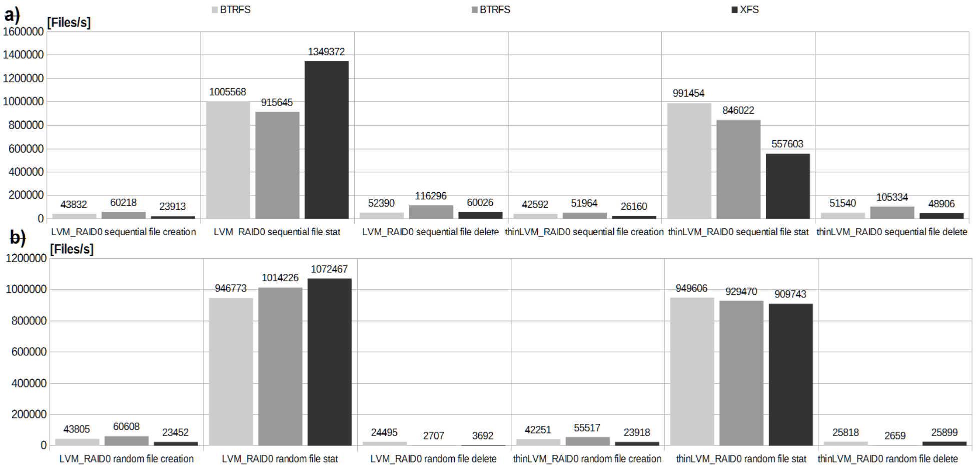

Figure 10. Filesystems performance comparison in multi-disk storage scenarios with RAID0 allocation policy with 5 disks and 128 KB chunk size for standard and thin provisioned logical volume: number of (a) sequential file operations, (b) random file operations

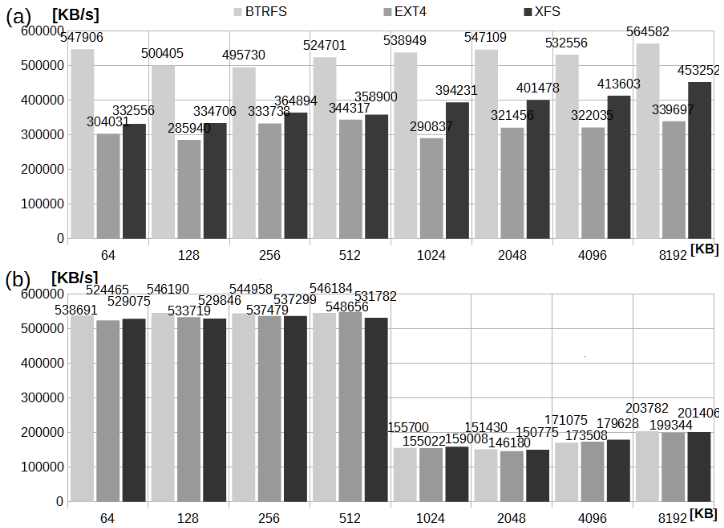

Figure 11. Filesystems performance depending on chunk size used by thin provisioned volume with data striping between 5 disks and 128KB chunk size: (a) data read speed from regular file, (b) data write speed to regular file

All tested filesystem types have more than double performance drop in data write speed to regular file in multi-disk storage configuration scenarios with thin provisioned volume using at least 1MB size of pool allocation unit.

5. Conclusions

Configuration of storage zones in operating system has impact on data access efficiency. File operation performance in storage zone is dependent on selection of filesystem type and block device configuration. Filesystem structures limits data access delay for file operations. In example, in all storage space configuration scenarios EXT4 filesystem has lowest performance of random file deletion.

The efficient configuration of storage zone with logical volume managed by LVM should distribute data between physical volumes localized on separate disks and requires selection of allocation policy and unit size. For fixed number of disks a RAID0 striping allocation policy provides better performance that RAID5 and RAID6, which was confirmed for all tested filesystem. Analysis of research results shows that increasing number of disk in storage space configuration provides better performance for regular file read and write data operations but not always guarantees improvement of all other file operation performance. In example, for XFS or BTRFS filesystem stored in logical volume with RAID0 striping policy a random file deletion has performance drop according to single disk storage space.

The allocation unit size for logical volume also has impact on file operations performance in filesystem localized in logical volume. For EXT4 and XFS bigger unit size for allocation policy provides performance grow of data read from regular file.

Additionally using LVM thin volume in storage zone configuration causes performance drop for all tested filesystems in file data read and write speed. The drop effect varies according to allocation unit size used in storage space allocation from thin provisioned pool. In multi-disk storage space configuration with thin provisioned volume recommended pool allocation unit size is up to 512KB. Beyond this limit regardless of the filesystem localized in a thin volume data write speed to regular file is significantly reduced.

REFERENCES

[1] Anderson E., Swaminathan R., Veitch A., Alvarez G., Wilkes J.: Selecting RAID Levels for Disk Arrays, Proceedings of the 1st USENIX Conference on File and Storage Technologies, 2002.

[2] Avantika, M., MingMing, C., and Suparna, B.: The new ext4 filesystem: current status and future plans, In: Linux Symposium, 2007.

[3] Bryant, R; Forester, R; Hawkes, J.: Filesystem performance and scalability in linux 2.4.17, USENIX ASSOCIATION PROCEEDINGS OF THE FREENIX TRACK, 2002.

[4] Hellwig C.: XFS the big storage file system for Linux, USENIX, Vol. 34, No. 2, 2009.

[5] Rodeh O., Bacik J., Mason C.: BTRFS: The Linux B-Tree Filesystem ACM TRANSACTIONS ON STORAGE, Vol. 9, Issue 3, 2013.