

Porównanie możliwości i cech współczesnych Linuxowych systemów plików: ext4, XFS, Btrfs

Bartłomiej Kossak*, Maciej Pańczyk*

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W artykule zostały porównane trzy nowoczesne systemy plików powszechnie stosowanych w systemach Linux: ext4, Btrfs i XFS. Została omówiona budowa i cechy funkcjonalne poszczególnych systemów plików. Systemy plików zostały porównane pod względem wydajności. Uwzględniono specyfikę dysków SSD i przystosowanie do pracy z nimi. Do testów funkcjonalności oraz wydajności został użyty system Ubuntu 15.10

Słowa kluczowe: system plików; ext4; btrfs; xfs

*Autor do korespondencji

Adresy e-mail: Bartek.kossak@gmail.com;m.panczyk@pollub.pl

Comparison of features and capabilities of modern Linux file systems ext4, XFS, Btrfs

Bartłomiej Kossak*, Maciej Pańczyk*

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B,20-618 Lublin, Poland

Abstract. This article compares three modern Linux file systems: ext4, Btrfs and XFS. Their construction and functional features were discussed. File systems were compared in terms of performance. Additionally modern SSD drives were included in discussion. Ubuntu 15.10 distribution was used to test functionality and performance of these three Linux file systems.

Keywords: file system; ext4; btrfs; xfs

*Corresponding author

E-mail addresses: Bartek.kossak@gmail.com; m.panczyk@pollub.pl

1. Wstęp

System plików służy do kontroli sposobu zapisu i odczytu danych. Tworzy on warstwę, która pozwala operować na nazwach wszystkim powłokom znajdującym się nad nią. Zaś wszystkim warstwom znajdującym się pod nią operować na adresach fizycznych [1].

Prawie wszystkie systemy operacyjne posługują się systemami plików. W przypadku najbardziej popularnych systemów operacyjnych, takich jak Windows, Linux, czy MacOS mamy do wyboru po wiele systemów plików, w zależności od potrzeb. Znaczący rozwój technologii informatycznej na przestrzeni ostatnich lat zrodził dużą ilość problemów, z którą systemy plików muszą sobie poradzić. W systemach Linux wszystko tworzone jest w formie plików. Plik to zbiór powiązanych ze sobą informacji. Użytkownik widzi jedynie nazwę pliku, natomiast system Linux traktuje plik jako ciąg bajtów. Dobór systemu plików zależy od zastosowania. Inny będzie właściwy dla systemów o wysokich wymaganiach ciągłej i bezawaryjnej pracy, inny dla systemów, w których krytyczna jest wydajność a jeszcze inny dla systemów, w których istotne są możliwości migracji bądź zmiany pojemności w locie (jak np. w przypadku maszyn wirtualnych).

W związku ze znaczącym rozwojem technologii wzrosły wymagania związane z bezpieczeństwem danych oraz wydajnością systemów informatycznych. Zmiana nośników danych z magnetycznych na technologie półprzewodnikową spowodowała wzrost zapotrzebowania na całkiem nowe cechy systemów plików.

Linux oferuje szeroką paletę systemów plików. W artykule zostaną porównane trzy najpopularniejsze: ext4, XFS i Btrfs. Ext4 jest to jeden z najpopularniejszych systemów plików domyślny w wielu dystrybucjach Linux. System plików ext4 jest kontynuacją systemu plików ext, który był pierwszym stworzonym od podstaw systemem plików dla Linuksa [2]. XFS jest dostępny w Linuxie od dwudziestu lat, jest stale rozwijany przez duże koncerny informatyczne, pozwala na obsługę bardzo dużych plików i wolumenów [3]. XFS można uznać za współczesny system plików dla wydajnego sposobu radzenia sobie z dyskami w erze „big data”. Btrfs to najmłodszy systemem plików, który jest oferowany w Linuxie [4]. Jest stale rozwijany, posiada wiele nowych cech np. optymalizacyjnych, czy zabezpieczających, a wiele z nich jest jeszcze w fazie testów [4].

2. System plików

System plików odpowiada za porządkowanie, budowanie hierarchii oraz tworzenie struktury porządkującej plików.

Ponadto realizuje operacje wymagane do działania systemu operacyjnego m.in.:

- tworzy i usuwa pliki;
- otwiera pliki do odczytu i zapisu;
- wyszukuje dane w plikach;
- zamyka pliki;
- tworzy katalogi;
- listuje zawartość katalogów;
- usuwa pliki z katalogów [1].

Linux niemal od samego początku, z powodów historycznych (odziedziczona różnorodność wersji i producentów systemów UNIX) miał do dyspozycji więcej niż jeden system plików. Do ułatwienia implementacji nowych systemów plików stworzona została warstwa abstrakcji, nazwana VFS (Virtual File System). Warstwa VFS to interfejs łączący jądro i implementację systemu plików. Dzięki tej warstwie operacje zapisu lub odczytu z punktu widzenia programisty są takie same. Zadaniem VFS jest przetłumaczenie podstawowych poleceń na polecenia zgodne ze specyfikacją systemu plików [1].

Z plikami w systemach UNIX związane są tzw. i-węzły. Każdy plik w systemie Linux posiada swój własny i-węzeł. W każdym i-węzle znajdują się elementy struktury opisującej plik (z wyjątkiem danych i nazwy) oraz wskaźniki do bloków danych [1].

Za obszarem rozruchowym (może on być ew. pusty) znajduje się superblok zawierający atrybuty i metadane danego systemu plików [1]. Ostatnim z podstawowych elementów jest obszar systemu plików zawierający bloki danych.

3. EXT4

System ext4 jest kontynuacją systemu plików z rodziny ext. System ext jest pierwszą wersją najpopularniejszego systemu plików dostępnego w Linuxie.

W ext nazwa pliku wynosiła 255 znaków, maksymalny rozmiar partycji i pliku wynosił 2GB [2]. System posiadał jednak dużą wadę - użycie listowego przydziału bloków prowadziło do występowania niekorzystnego zjawiska jakim była nieciągłość obszarów danych na dysku twardym, czyli fragmentacja [1].

W ext2 sunięty został problem związany z fragmentacją plików, poprzez zastosowanie mechanizmu przydzielania bliskich bloków oraz prelokację [1,2]. Lokalizacja bloków z danymi odbywa się na dwa sposoby: bezpośrednio - adresy pierwszych bloków są zapisane bezpośrednio w i-węzle - oraz pośrednio - adresy bloków zapisane są w bloku, którego adres znajduje się w i-węzle [1]. Wprowadzony został również mechanizm naprawy systemu plików po awarii (fsck), który uruchamiany jest przy starcie systemu - naprawia szkody poprzez przywracanie uszkodzonych plików do odpowiedniego katalogu [5]. Skanowanie było mniej wydajne niż w przypadku następnych systemów plików z księgowaniem.

System plików ext3 nie różnił się budową od ext2. Najważniejszym elementem, który został wprowadzony do ext3 był mechanizm księgowania [6]. Księgowanie

zabezpiecza pliki przed utratą lub uszkodzeniem i polega na zapisaniu metadanych - informacji o danej - na początku do dziennika, a następnie zapisaniu danych na dysku. W systemie plików ext3 mamy do wyboru trzy tryby księgowania:

- journal - najbezpieczniejszy i najwolniejszy tryb tworzenia kroniki, tworzone są dane i meta dane. W tym sposobie występuje najmniejsza szansa na stratę danych ale wydłuża czas pracy dysku, ponieważ w trakcie tworzenia nowego pliku bloki danych muszą być zapisane w dzienniku dwa razy.
- ordered - najpopularniejszy tryb księgowania, w którym zapisywane są jedynie meta dane. Tryb zagwarantuje integralność systemu plików lecz w końcowych częściach plików mogą wystąpić nieprawidłowości w danych.
- writeback - do dziennika zapisywane są jedynie meta dane, operacja zapisu dokonywana jest zgodnie z harmonogramem, zapisywane są wtedy dane, a następnie metadane. Jest to tryb najszybszy lecz mniej bezpieczny, zezwala na modyfikację danych, których metadane zostały już zmodyfikowane, a dane nie zostały zapisane jeszcze na dysk [5].

Nową cechą wprowadzoną do ext3 jest indeksowanie katalogów. W ext3 odpowiada za nią funkcja hashetrees, dzięki której katalogi mają postać drzewa [5]. Katalogowanie jest potrzebne w przypadku posiadania dużej ilości plików w katalogach. Drzewiasta struktura znacznie skraca czas dostępu do szukanego pliku. System ext3 dopuszcza podział bloku na równe części, w których jest możliwość przechowywania plików, dzięki temu zmniejsza fragmentację w przypadku dużych ilości małych plików.

W systemie plików ext4 wielkość pojedynczego pliku została zwiększona do 16 TiB, oraz obsługa pojedynczych partycji do 1 EiB, liczba możliwych podkatalogów została zwiększona do 64 tysięcy [5].

Główną cechą różniącą ext4 od starszych wersji ext jest mechanizm ekstentów [5], który zastępuje adresowanie pośrednie bloków. Ekstent jest to ciągły zbiór bloków, w którym przechowywane są pliki, ekstenty przechowywane są w i-węzle zamiast wskaźników do bloków [5]. Dzięki ekstentom zwiększył się rozmiar pojedynczego pliku, zwiększona została wielkość pojedynczej partycji i zredukowany jest rozmiar meta danych. Redukcja wielkości meta danych znacznie przyspiesza prace na dużych plikach. Włączenie usługi ekstents umożliwia realizację prelokacji bloków, który odpowiada za ciągły przydział obszaru dla pliku.

Nową cechą ext4 jest opóźniona alokacja [6], technika ta opóźnia zapis danych na dysku przez przetrzymywanie ich w pamięci podręcznej. Opóźniona alokacja odbywa się w warstwie wirtualnej systemu plików. Rozszerzeniem opóźnionej alokacji jest alokacja wieloblokowa (mballoc). Zaletami tego rozwiązania jest zwiększenie wydajności w przypadku zapisu plików rosnących - połączenie jednoczesnej i wieloblokowej alokacji zwiększa szanse na to, że kolejne alokacje zgrupują się w jedną. Następną zaletą jest zmniejszenie fragmentacji danych dla plików rosnących, oraz duża szansa na brak alokacji dla plików tymczasowych [2].

Rozwiązanie posiada również wady, zwiększone jest ryzyko utraty danych podczas awarii zasilania.

W ext4 został usprawniony również mechanizm księgowania dzięki dodaniu sum kontrolnych dziennika [5]. Dziennik jest najintensywniej używanym obszarem na dysku, z tego powodu uszkodzenia w tym obszarze mogą doprowadzić do awarii całego systemu plików. Suma kontrolna to ciąg znaków, powstających w wyniku operacji matematycznych podczas przesyłania danych. W przypadku systemu plików, suma kontrolna wyliczana jest dla każdej transakcji i deskryptora grupy bloków [5]. Nową cechą, która została wprowadzona jest mechanizm barier. Odpowiada za wydawanie poleceń sterownikowi dysku zapisu danych w określonym porządku, co wzmacnia spójność danych na dysku [5].

4. XFS

XFS to wysokowydajny 64-bitowy system plików z obsługą księgowania. Obsługiwany jest w większości systemów operacyjnych Linux.

Projektując system plików XFS przyjęto następujące cele [1]:

- system plików powinien nadawać się do pracy w naukowych serwerach plików, komercyjnych serwerach obróbki danych oraz w serwerach mediów elektronicznych;
- musi odzyskać szybko sprawność po awarii, zapewnić dużą dostępność danych, utrzymywać dane dyskowe w spójnym stanie;
- musi obsługiwać 64-bitowe pliki;
- musi obsługiwać wydajnie pliki rozrzedzone – czyli pliki, które w dowolnym miejscu zawierają dziury (obszary, w których zapis nigdy się nie odbył, odczytywane są jako bloki zer);
- zwiększona wydajność dla obsługi małych plików, mniejszych niż 1KB;
- musi wydajnie obsługiwać duże katalogi;
- obsługa listy kontroli dostępu oraz funkcji POSIX 1003.6 – rozszerzenia zabezpieczeń systemu spełniające kryteria opublikowane przez Departament obrony USA;
- umożliwienie dynamicznej zmiany logicznych rozmiarów bloków [1,3].

System plików obsługuje pliki o wielkości 8 EiB oraz maksymalną wielkość partycji do 16 EiB. Długość nazwy pliku reprezentowana jest za pomocą ciągu 255 znaków z tablicy ASCII [18]. Rozmiar jednostki alokacji wynosi od 512KB do 1MB. Jak wszystkie systemy plików, XFS również pracuje pod kontrolą VFS. XFS jest dziennikowym systemem plików, co oznacza, że aktualizacja meta danych odbywa się szeregowo w obszarze dziennika zanim nastąpi rzeczywista aktualizacja [1]. W przypadku awarii, operacje zapisu można powtórzyć lub cofnąć za pomocą danych do przywrócenia systemu plików zawartych w dzienniku. Głównymi komponentami systemu XFS są:

- menedżer dziennika – wszelkie zmiany w meta danych są rejestrowane szeregowo w specjalnie wyszczególnionym do tego obszarze. Każdy system plików prowadzi

oddzielny dziennik, dzięki któremu możliwa jest szybka rekonstrukcja wadliwego systemu plików.

- menedżer pamięci podręcznej buforów – w pamięci podręcznej buforu przechowywane są bloki dysków, przechowuje się tu meta dane systemu pliku i dane plików
- menedżer rygli – realizuje ryglowanie plików użytkownika
- menedżer przestrzeni dyskowej – rozporządza alokacją przestrzeni dyskowej w systemie plików
- menedżer atrybutów – realizuje operacje na atrybutach systemów plików
- interfejs wywołań systemowych i VFS
- menadżer przestrzeni nazw – tłumaczy ścieżki dostępu do odwołania do plików [1].

System plików XFS jest podzielony na grupy alokacji jednakowej wielkości. Grupa alokacji jest to autonomiczna jednostka systemu plików, która zawiera osobne struktury potrzebne do zarządzania jej przestrzenią [1].

Grupy alokacji składają się z superbloku umieszczonego w lokalizacji 0, nagłówka grupy alokacji oraz danych wskazanych przez nagłówek grupy alokacji [1]. Nagłówek grupy alokacji składa się z następujących pól:

- magicznego numeru nagłówka alokacji;
- numeru wersji nagłówka grupy alokacji;
- numeru sekwencyjnego grupy alokacji;
- w zależności od używanej alokacji, jeśli jest alokacja z bitmapą – położenie oraz rozmiar bitmapy wolnych bloków, w przypadku alokacji z dwoma B-drzewami [33] – położenie korzenia obu B-Drzew;
- położenia (względnego numeru bloku) i-węzła, który zawiera tablicę i-węzłów [1].

Przestrzeń magazynowa plików w XFS reprezentowana jest na trzy sposoby [1]. Wybór metody zależy od rozmiaru i ciągłości pliku. Dla plików małych, dane przetrzymywane są w i-węźle. W przypadku plików średnich wykorzystywane są ekstenty (i-węzeł zawiera wskaźnik do danego ekstentu, który zawiera dane pliku). W przypadku plików dużych stosowane są B-Drzewa [3].

System XFS można aktywnie rozszerzać poprzez dodanie przestrzeni do bazowego woluminu, operację taką przeprowadza menadżer przestrzeni dyskowej [3]. XFS obsługuje mechanizm opóźnionej alokacji, który podczas operacji zapisu pliku jest zapisywany do bufora pamięci podręcznej [3]. Dzięki tej metodzie zwiększona jest szansa, że plik zostanie zapisany w ciągłej grupie bloków, co zmniejsza problemy z fragmentacją i wydajnością.

System XFS obsługuje tzw. pliki rozrzedzone [1,3]. Plik rozrzedzony to typ pliku, który zezwala zapisywać dane w dowolnym miejscu bez zajmowania niezapisanych bloków pliku.

Dla zwiększenia wydajności XFS udostępnia bezpośrednią realizację operacji we/wy. XFS pozwala na niebuforowanie we/wy, ponieważ dane przekazywane są poprzez bufor aplikacji i dysku używającego DMA (umożliwiający zapis pełnego pasma we/wyz podstawowych urządzeń).

Cechą unikalną systemu XFS jest gwarantowana przepustowość operacji we/wy, za rezerwację odpowiedniej przepustowości odpowiada interfejs API [3]. Wykorzystywane jest to głównie do zastosowań w czasie rzeczywistym np. strumieniowe transmisje wideo.

XFS zapewnia również defragmentację plików podczas pracy, tzw. defragmentacja online oraz zmianę rozmiaru online, dzięki czemu system plików może zwiększać swój rozmiar tak długo jak jest dostępne wolne miejsce na urządzeniach fizycznych [3].

XFS obsługuje również mechanizm księgowania, który ma za zadanie zapewnienie spójności danych w przypadku awarii zasilania lub systemu [1,3]. System XFS zapewnia księgowanie meta danych, których zapis odbywa się w pierwszej kolejności przed tym, jak zostaną zaktualizowane bloki dysku. Pliki dziennika systemu są zapisywane w odrębnej części dysku, która nie jest wykorzystywana podczas normalnej pracy systemu. W XFS możemy wyróżnić dwa rodzaje dzienników: logiczny i fizyczny [1]. Logiczny opisuje w sposób zrozumiały dla człowieka jakie czynności zostały wykonywane, natomiast dziennik fizyczny przechowuje kopie bloków zmodyfikowanych podczas każdej operacji. Aktualizacje dzienników wykonywane są asynchronicznie, dzięki temu zwiększona jest wydajność systemu plików [1]. W przypadku awarii XFS potrafi odzyskać pliki, które nie zostały zapisane zgodnie z informacjami zawartymi w dzienniku. Odzyskiwanie odbywa się automatycznie w trakcie pierwszego uruchomienia systemu plików po awarii, szybkość odzyskiwania nie zależy od wielkości plików, a od operacji w nich dokonanych.

5. Btrfs

Kolejnym rozważanym formatem plików jest 64-bitowy Btrfs. W systemie Btrfs maksymalna wielkość pliku wynosi 16 EiB, przestrzeń nazw wynosi 255 znaków z tablicy ASCII, a liczba plików, która może zmieścić się w katalogu wynosi 264 [4].

Główne założenia systemu plików Btrfs to:

- wykorzystanie mechanizmu kopiowania przy zapisie do tworzenia migawek systemowych;
- możliwość odtworzenia stanu systemu z punktów w przeszłości,
- użycie sum kontrolnych w celu zapewnienia integralności danych;
- efektywniej wykorzystywana przestrzeń dyskowa po przez przyjazną dla użytkownika kompresję danych;
- możliwość pracy na wielu woluminach, pełna integralność z macierzami RAID;
- możliwość dynamicznej zmiany pojemności woluminu, za pomocą usuwania i dodawania urządzeń fizycznych [4].

System Btrfs w całości jest oparty o las B-drzew w zmienionym wariantcie. W systemie Btrfs zastosowano zmodyfikowaną wersję zwaną B+drzewami [4]. Zostały usunięte połączenia między sąsiadującymi liśćmi, zamiast tego została stworzona lista jednokierunkowa, która znacząco przyspieszała przechodzenie po zawartości struktury. Na rzecz systemu Btrfs struktura Drzew została wzbogacona

o kilka rozwiązań: zastosowano aktualizację drzewa od wierzchołka do liścia, usunięto powiązania między liśćmi, wprowadzono zliczanie dowiązań. Dzięki tego typu modyfikacjom operacje dodawania i usuwania elementów nie wymagają kopiowania całego drzewa, lecz tylko ścieżki prowadzonej do elementu poddanego modyfikacji. W systemie plików Btrfs drzewo jest wykorzystywane jako podstawowa. Struktura tego typu pozwala tworzyć dynamiczny system do przechowywania danych. Wszystkie informacje i struktury, które są częścią architektury systemu (wykluczając superblok), są przechowywane wewnątrz lasu drzew [6].

Jak już wspomniano system plików Btrfs opiera się na mechanizmie kopiowania przy zapisie [4]. Mechanizm kopiowania przy zapisie to technika optymalizacji w przypadku potrzeby pracy na dużej ilości danych. Zamiast rzeczywistego kopiowania pamięci, które wymaga dużych nakładów czasu i pamięci, zwracany jest jedynie wskaźnik do oryginalnych danych. Kopiowanie odbywa się wtedy, gdy zachodzi potrzeba ich modyfikacji. Najważniejszą zaletą jest brak wykonywania kopii pliku, który nie został zmodyfikowany [6]. W przypadku Btrfs funkcja COW daje możliwość skopiowania sektora dysku, podczas gdy dysk nadpisywany jest nową porcją informacji [4]. Dzięki temu w łatwy sposób można tworzyć kopie zapasowe. Dotychczas, aby to zrobić należało kopiować fizycznie dane w inne miejsce na dysk.

Dzięki strukturze Btrfs nie jest wymagana fizyczna kopia. Wystarczy wykonać dodatkowe powiązanie w systemie plików, dzięki któremu dwa odrębne pliki będą wskazywać na ten sam blok danych fizycznie zapisanych na nośniku. Rozwiązanie jest nie tylko szybkie, ale również i oszczędne, ponieważ kopiowane są tylko informacje które są różne w pliku oryginalnym i jego kopii [4].

W jednym wolumenie Btrfs może znajdować się kilka podwoluminów [4]. Podwolumin może działać jako odrębny system plików. Podczas tworzenia podwoluminów nie jest wymagane partycjonowanie. Podwoluminy tworzone są z dostępnej przestrzeni dysku. Tak utworzony wolumen widziany jest jako katalog główny [4]. W połączeniu z mechanizmem kopiowania przy zapisie, można stworzyć kopię całego systemu plików bez utraty dużej ilości przestrzeni dyskowej. W tym przypadku zmianie ulegają jedynie połączenia w logicznych strukturach, a kopiowanie następuje dopiero, gdy plik oryginalny zostanie zmodyfikowany. W przypadku rozwiązań zdalnych i chmurowych, podczas aktualizacji danych nie ma konieczności kopiowania wszystkich danych. Kopiowane są jedynie sektory, które uległy modyfikacji [4]. Rozwiązania wprowadzone w Btrfs znacząco przyspieszają operacje na dużych plikach [4].

Wprowadzenie sum kontrolnych znacząco zwiększa bezpieczeństwo plików na mikro uszkodzenia. W Btrfs tworzona jest suma kontrolna dla każdego sektora dysku zawierającego informację. Podczas każdego odczytu suma kontrolna jest ponownie obliczana, a w przypadku zmiany wartości któregośkolwiek z bitów, suma kontrolna nie będzie się zgadzać z tą zapisaną wcześniej. W przypadku błędu, system Btrfs może automatycznie przywrócić informację,

zapobiegając tym samym trwałym uszkodzeniom pliku [4].

W systemie Btrfs wprowadzono szereg funkcji wspomagających zarządzanie partycjami bez potrzeby odmontowania systemu pliku [4]. Rozwiązanie wykorzystywane jest podczas dynamicznej zmiany rozmiaru wolumenu. Dzięki funkcji dynamicznej zmiany rozmiaru wolumenu jest możliwość zmian rozmiaru partycji przy działającym systemie.

Nową cechą wprowadzoną do Btrfs jest możliwość kompresji „w locie” [4]. W przypadku Btrfs za kompresję odpowiadają dwa algorytmy ZLIB (wolniejszy, większe ratio) i LZO (szybszy, mniejsze ratio) [4]. Mechanizm kompresji jest inteligentny, plik który nie został wcześniej skompresowany, zostaje automatycznie poddany kompresji. Dzięki funkcji kompresji nie tylko jest zmniejszona objętość, ale również poprawiona jest wydajność całego systemu plików [4].

Nową cechą wprowadzoną do Btrfs jest funkcja migawek [4]. Tradycyjne menedżery woluminów wymagają, by cały dysk logiczny był objęty migawką. Btrfs natomiast pozwala tworzyć migawki zarówno całych folderów, jaki pojedynczych plików znajdujących się w dowolnym miejscu na dysku [4]. Migawka jest bardziej użytecznym rozwiązaniem niż kopiowanie całego woluminu. W środowiskach produkcyjnych, można uniknąć wyłączenia maszyny i zachować podzielność danych podczas archiwizacji całych plików lub folderów. Migawki można zaplanować zgodnie z harmonogramem, co znacznie ułatwia administratorom powrót do archiwalnej wersji pliku (przypadku modyfikacji lub skasowania), nie zatrzymując środowiska produkcyjnego [4].

Rynek macierzy danych stale się rozwija. Na przestrzeni ostatnich kilku lat popularność zyskały dyski zbudowane na bazie pamięci flash tzw. SSD (ang. solid-statedrive). System Btrfs posiada szereg funkcji mających na celu sprawną pracę dysków SSD m.in. TRIM [4].

Prace nad Btrfs cały czas trwają. Niektóre funkcje, które zostaną dodane to: szybka weryfikacja systemu plików, dynamiczna weryfikacja systemu plików, równoważenie obciążenia danych w przypadku obsługi maszyn wielowątkowych, dodanie nowych algorytmów kompresji, czy funkcje związane z wydajnością transferu.

6. Badania

W testach została zbadana szybkość odczytu i zapisu dużej liczby małych plików oraz dużych plików.

6.1 Środowisko testowe

Do przeprowadzenia testów został użyty Laptop Dell Precision M4500 z dyskiem twardym HDD 500GB, o prędkości obrotowej 7200 obr/min oraz zainstalowanym systemem Ubuntu 15.10. Do przeprowadzenia testów użyto własnego prostego skryptu bash.

6.2 Przebieg badania

Przed przeprowadzeniem testów, należało wykonać poprawną konfigurację systemu Ubuntu. Aby uzyskane testy

były wiarygodne, zainstalowany został trzy razy system Ubuntu wraz z testowanym systemem plików.

Testy wydajności będą obejmować czas wykonywania operacji kopiowania przez system plików. Testy obejmą kopiowanie dużej ilości małych plików o rozmiarach 1KB, 1M, 1GB, 5GB, 10GB, 75GB. Do testów wykorzystano skrypt bash, który ma za zadanie generowanie plików, wykonywanie operacji kopiowania na danej partycji z folderu do folderu, zmierzenie czasu wykonania operacji, usunięcie plików. Poniżej został przedstawiony użyty skrypt bash do testowania wydajności systemu plikowego.

Przykład 1. Skrypt bash badający czas wykonywania operacji kopiowania w systemie plików

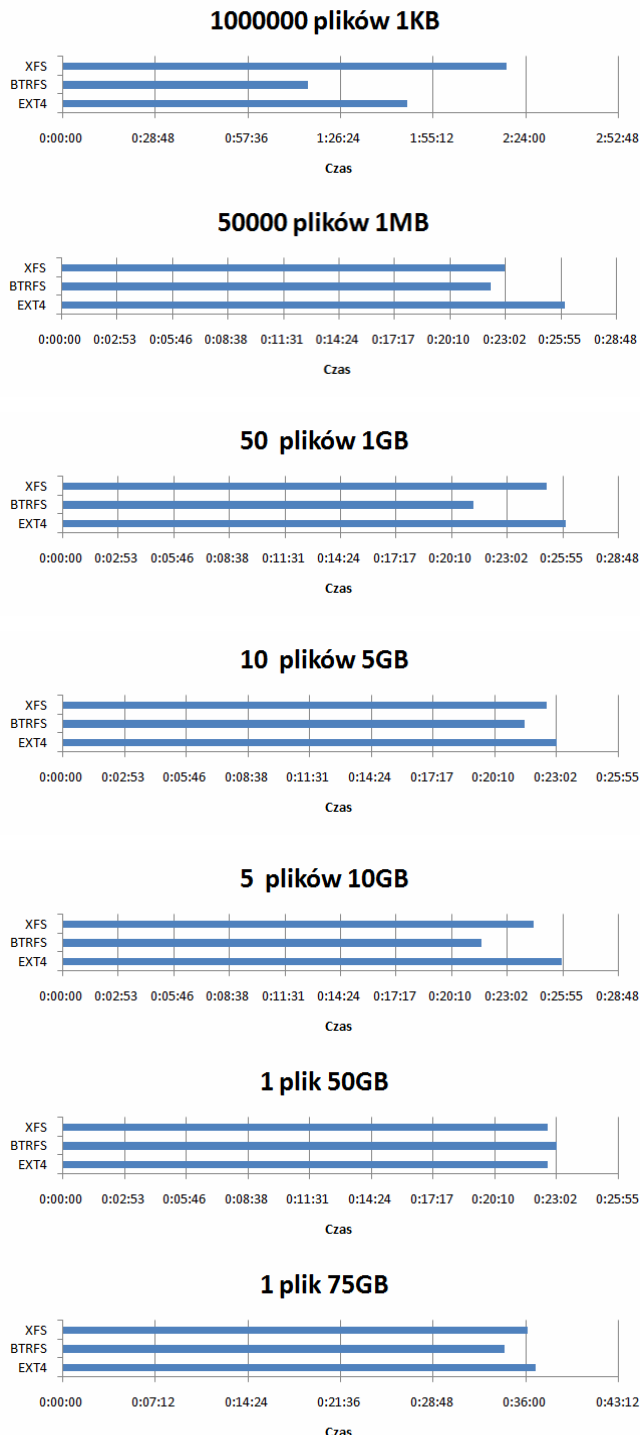
```
1. #!/bin/bash
2. NUM_FILES=10 #ilość plików
3. SIZE_FILE=500M #rozmiar pliku
4. FROM_COPY=from_copy
5. TO_COPY=to_copy
6. mkdir $FROM_COPY #folder tworzenia plików
7. mkdir $TO_COPY #folder do którego kopiowane są pliki
8. cd $FROM_COPY #przejdź do katalogu, w którym
   generowane będą pliki
9. echo "Tworzę pliki"
10. for i in $(seq $NUM_FILES); do
11. dd if=/dev/zero of=file_${i}bs=$SIZE_FILE count=1 &&
   /dev/null #funkcja generująca pliki
12. done
13. echo "Utworzyłem pliki"
14. cd ..
15. echo "Rozpaczynam kopiowanie"
16. echo "Kopiowanie $NUM_FILES plików o rozmiarze
   $SIZE_FILE zajęło: "
17. time(cp $FROM_COPY $TO_COPY -rf; sync)#pomiar
   czasu kopiowania
18. rm $FROM_COPY $TO_COPY -rf#kasowanie plików
19. sync
```

6.3 Prezentacja rezultatów badań

Testy wykazały, że najwydajniejszym systemem plików z omawianych jest system Btrfs. Na rysunku 1 zostały przedstawione wyniki testów. Uzyskał on najlepszy czas w sześciu z siedmiu przeprowadzonych testów. Z testów wynikało, że Btrfs gorzej radzi sobie z dużą ilością małych plików, lecz bardzo dobrze poradził sobie z plikami dużymi. Na drugiej pozycji znalazł się system XFS, otrzymał najlepszy wynik w jednym z sześciu przeprowadzonych testów. Czas wykonywania operacji przez system XFS w przypadku dużych plików był gorszy o około 10%. XFS najgorzej poradził sobie w przypadku operacji na dużej ilości małych plików, wynik uzyskany przez XFS był o 50% gorszy od wyniku uzyskanego przez system BTRFS. W teście drugim system XFS uzyskał zbliżony wynik do Btrfs, w pozostałych czterech testach wynik był od 3% do 6% gorszy od wyników uzyskanych przez Btrfs.

System Btrfs jest systemem plików, który służy do zapewnienia najwyższej wydajności. Wysoka wydajność obsługi plików w połączeniu z przystosowaniem do obsługi dysków SSD stanowią bardzo dobre połączenie dla rozwiązań wysoko wydajnościowych. System XFS mimo, że na rynku dostępny jest od ponad dwudziestu lat nie ustępuje znacząco wydajnościowo najnowszemu systemowi Btrfs. XFS od 2013 roku jest stale rozwijany i ma szanse w niedługim czasie

wyprzeć jeden z najpopularniejszych systemów jakim jest ext4. Ext4 powoli przestaje być domyślnym systemem plików w wielu dystrybucjach Linuxa. Jest mniej wydajny od pozostałych testowanych systemów plików oraz zajmuje najwięcej miejsca w przypadku zamontowania systemu plików na partycji.



Rys 1. Wyniki testów wydajnościowych poszczególnych systemów plików

6.4 Zestawienie cech funkcjonalnych

W tabeli 1 zawarte jest zestawienie cech funkcjonalnych porównywanych systemów plików.

7. Wnioski

Aktualnie najlepszy okazał się system Btrfs - jest najwydajniejszy, posiada szereg ciekawych funkcji oraz jest stale rozwijany. Btrfs idealnie nadaje się do rozwiązań domowych jak i niekomercyjnych. Na drugim miejscu uplasował się system XFS, który odbiega wydajnością od Btrfs, ale jest bardzo stabilny oraz wydajnie obsługuje systemy zawierające wiele małych plików. W połączeniu z kompresją w locie daje nieograniczone możliwości powiększania systemu plikowego. Najgorszy okazał się system ext4. Mimo, że łatwo można z niego odzyskać pliki oraz, że posiada kompresję w locie – tylko w przypadku zastosowania partycji GTP- jest to nadal bardziej zbiór ulepszeń niż nowy w sensie, zaprojektowany od podstaw system plików, który jest w stanie zaspokoić potrzeby ciągle rozwijającego się zapotrzebowania na rynku informatycznym

Tabela 1. Cechy funkcjonalne porównywanych systemów plików

Cecha funkcjonalna	EXT4	XFS	BTRFS
Programy testujące i naprawiające	posiada	brak	posiada
Mechanizm sum kontrolnych	posiada	brak	posiada
Mechanizm opóźnionej alokacji	posiada	posiada	posiada
Dynamiczne rozszerzanie rozmiaru woluminu	z pomocą LVM	tak	tak
Indeksowanie katalogów	posiada	brak	brak
Obsługa plików rozrzedzonych	nie	tak	nie
Obsługa plików 64-bitowych	nie	tak	tak
Migawki systemu	nie	nie	tak
Defragmentacja podczas pracy systemu plików	brak	posiada	Posiada
Mechanizm księgowania	tak	nie	tak
Konwersja na inny system plików	tak na ext3	brak	tak na ext4
Gwarantowana przepustowość danych	brak	posiada	Brak
Obsługa macierzy RAID	tak	tak	tak

Literatura

- [1] Bar M.: Linux systemy plików, Wyd. I Warszawa: RM 2002. ISBN 83-87-02948-1.
- [2] Seidel U .: System plików ext. „Linux Magazine”, 2013, nr 12, s. 63-66.
- [3] Oficjalna strona systemu XFS: http://xfs.org/index.php/Main_Page [06.06.2013]
- [4] Wiki systemu Btrfs: https://Btrfs.wiki.kernel.org/index.php/Main_Page [18.01.2016]
- [5] Wiki systemu ext4 https://ext4.wiki.kernel.org/index.php/Main_Page [03.09.2016]
- [6] Brendel J.C .: Porównanie Ext3, Ext4, XFS i Btrfs. „Linux Magazine”, 2014, nr 9, s.18- 21