VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

POROVNÁNÍ VÝKONNOSTI RŮZNÝCH SOUBOROVÝCH SYSTÉMŮ

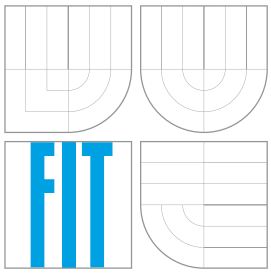
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR BEŇAS

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

POROVNÁNÍ VÝKONNOSTI RŮZNÝCH SOUBOROVÝCH SYSTÉMŮ

A COMPARISON OF A PERFORMANCE OF DIFFERENT FILESYSTEMS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR BEŇAS

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ALEŠ SMRČKA, Ph.D.

BRNO 2011

Abstrakt

Cílem této práce je porovnat výkonnost souborových systémů běžně používaných na operačním systému GNU/Linux. Jsou popsány dnes běžné souborové systémy a existující nástroje pro výkonnostní testování souborových systémů. V textu práce je zdokumentován použitý hardware, nastavení operačního systému a v rámci práce vytvořený nástroj pro analýzu výsledků benchmark programu Iozone. Jsou uvedeny a analyzovány výsledky benchmark programů Iozone a Postmark.

Abstract

The aim of this work is to compare the performance of common local filesystems used in GNU/Linux operating system. State-of-the-art filesystems together with existing benchmarks are described. The work also describes in detail the hardware used for performing benchmarks and the settings of the operating system. A tool for the analysis of benchmarks performed by the Iozone tool has been developed. Finally, the analysis and the discussion upon the results of Iozone and Postmark has been made.

Klíčová slova

výkonnostní testování, porovnání, souborové systémy, Ext2, Ext3, Ext4, XFS, Btrfs, Iozone, Postmark, Beaker

Keywords

performance testing, comparison, filesystems, Ext2, Ext3, Ext4, XFS, Btrfs, Iozone, Postmark, Beaker

Citace

Petr Beňas: Porovnání výkonnosti různých
souborových systémů, bakalářská práce, Brno, FIT VUT v Brně, 2011

Porovnání výkonnosti různých souborových systémů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doktora Aleše Smrčky, další informace mi poskytli RNDr. Jiří Hladký a Ondřej Hudlický ze společnosti Red Hat. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Petr Beňas
11. května 2011

Poděkování

Zde bych chtěl poděkovat doktoru Aleši Smrčkovi za odborné vedení této práce a dále konzultantům z firmy RedHat doktoru Jiřímu Hladkému a Ondřeji Hudlickému, kteří mi poskytli odbornou pomoc.

Obsah

Kapitola 1

Úvod

Tato práce se zabývá výkonnostním testováním souborových systémů. Obecným cílem většiny druhů testování je odhalení chyb ve vyvíjeném software. Za chybu v softwarovém produktu můžeme považovat takové případy, kdy se skutečné chování produktu liší od chování požadovaného specifikací. Žádný programátor nebo tým z principu nevytváří bezchybný kód, proto je testování nedílnou součástí softwarových projektů jakéhokoli rozsahu. Přínosem testování ve vývoji softwarového produktu je odhalování a poukazování na chyby, které tak mohou být opraveny. Tím testování významně přispívá k výsledné kvalitě vytvářeného produktu.

Primárním účelem výkonnostního testování je měření výkonu nějakého softwarového produktu nebo jeho části. Na rozdíl od jiných typů testování, které lze provádět ručně nebo automatizovaně, je k výkonnostnímu testování zapotřebí použití nástroje pro zatížení testovaného produktu a měření výkonnosti. Výsledek testování výkonnosti má typicky podobu propustnosti (množství zpracovaných dat za jednotku času) nebo počtu provedení operací za jednotku času. Význam měření výkonnosti softwarového produktu spočívá v tom, že tato naměřená data je možné dále porovnávat. Porovnáním výkonnosti různých verzí produktu lze v průběhu vývoje odhalit chybu nebo potvrdit pozitivní dopad provedených změn na výkonnost. Oproti tomu, tato práce bude mezi sebou porovnávat z hlediska výkonnosti různé souborové systémy.

V současné době je v produkčním prostředí pod operačním systémem GNU/Linux využívána řada souborových systémů. Tato práce má za cíl porovnat lokální souborové systémy podporované Red Hat Enterprise Linuxem z hlediska výkonnosti. Z důvodu paralelní existence a podpory řady lokálních souborových systémů předpokládám, že každý souborový systém je díky svým odlišným výkonnostním vlastnostem vhodný za jiných okolností. V rámci této práce se budu snažit tyto okolnosti pro každý souborový systém najít. Cílem této práce tedy není rozdělit současné souborový systémy na výkonné a nevýkonné nebo najít univerzálně nejvýkonnější souborový systém.

Nejprve budou představeny souborové systémy používané pod GNU/Linux, mezi které patří Ext2, Ext3, Ext4 a XFS. Zmíněn bude i v současné době vyvíjený Btrfs. Následuje popis programů určených k měření výkonnosti souborových systémů. Po teoretickém úvodu jsou popsány programy Iozone, Postmark a FileBench. Ve čtvrté kapitole bude zdokumentováno použité počítačové vybavení, nastavení operačního systému a použité parametry programu Iozone. Následně je popsán vytvořený nástroj pro analýzu výstupů Iozone a porovnávány testované souborové systémy proti sobě.

Kapitola 2

Souborové systémy pod GNU/Linux

Tato kapitola popisuje současné souborové systémy podporované pod operačním systémem Red Hat Enterprise Linux. Mezi ně patří Ext2, Ext3, Ext4 a XFS. Souborový systém Btrfs je postaven na řadě zajímavých konceptů, proto je zde uveden i přesto, že v současném stavu nepatří mezi souborové systémy běžně nasaditelné v produkčním prostředí. U každého souborového systému bude uveden jeho základní popis a zdůrazněny vlastnosti, které mohou mít vliv na jeho výkonnost.

2.1 Ext2

Alfa verze souborového sytému Ext2 (The Second Extended Filesystem) byla podle článku [9] zveřejněna v roce 1993. Ext2 vychází ze staršího systému Ext, který byl spolu s vrstvou VFS (Virtual File System) přidán do linuxového jádra v roce 1992. Ext2 je stabilní a ve své době byl standardním souborovým systémem v GNU/Linux.

2.1.1 Limity

Limity souborového systému Ext2 jsou z dnešního pohledu nedostatečně nízké. Maximální velikost celého souborového systému je 4 TiB, velikost souboru je limitována dvěma GiB. Jméno souboru může mít maximálně 255 znaků, tuto hodnotu lze však zvětšit až na 1 012 znaků.

2.1.2 Základní vlastnosti Ext2

Ext2 implementuje základní koncepty unixového operačního systému. Mezi ně patří běžné soubory, adresáře, speciální soubory pro přístup ke znakovým a blokovým zařízením a symbolické odkazy.

Mezi další vlastnosti Ext2, které dnes již považujeme za samozřejmé, patří například dědičnost atributů aktuálního adresáře při vytváření souboru, neměnné soubory nebo soubory, jejichž obsah lze pouze přidávat a číst jej. Ext2 rezervuje typicky 5 % bloků pro superuživatele, aby mohl vyřešit přeplnění souborového systému způsobené procesy běžných uživatelů. Dále umožňuje při připojování souborového systému zvolit režim bezpečného mazání, kdy je obsah mazaného souboru přepsán náhodnými daty.

Kontrola souborového systému je vynucena po určitém množství připojení souborového systému nebo po vypršení zadaného času. Tyto hodnoty mohou být specifikovány při vytváření souborového systému (`mke2fs`) nebo později změněny nástrojem `tune2fs`.

Souborový systém Ext2 je složen ze skupin bloků (angl. block groups). Každá skupina bloků obsahuje redundantní kopii kritických metadat souborového systému (superblok a deskriptor souborového systému) a dále část souborového systému – bitmapy bloků a i-uzlů, část tabulky i-uzlů a datové bloky. Toto uspořádání usnadňuje obnovení souborového systému v případě poškození dat superbloku.

2.1.3 Vlastnosti Ext2 zajímavé z hlediska výkonnosti

Velikost logického bloku je nezávislá na velikosti bloku použitého fyzického úložiště. Volba většího bloku znamená lepší výkonnost z důvodu menšího přesunu hlaviček na disku, ale také větší plýtvání diskovým prostorem. Poslední blok souboru nemusí být zcela naplněn, ale alokován být musí. Celá jeho prázdná část tedy zůstává nevyužitá a čím větší blok bude, tím větší bude i objem alokovaného, ale nevyužitého diskového prostoru.

Rychlé symbolické odkazy jsou další vlastností souborového systému Ext2, která bude mít dopad na jeho výkonnost. Jméno cíle není uloženo v datovém bloku, ale přímo v i-uzlu. Zde je však k dispozici pouze omezené místo, proto jsou rychlé symbolické odkazy používány pouze v případě, kdy jméno cíle nepřesáhne 60 znaků. V opačném případě je potřeba uložit jméno do datového bloku.

Čtení dopředu (angl. readahead) je způsobem, kterým Ext2 zlepšuje výkonnost čtení souvislého bloku dat. Při sekvenčním čtení je při čtení bloku načteno i několik následujících bloků. Tím je zajištěno, že při požadavku na čtení následujícího bloku již bude tento blok uložen ve vyrovnávací paměti umístěné v operační paměti. Odtud je pak čtení výrazně rychlejší než přímo z disku.

Optimalizace alokace bloků se snaží shlukovat příslušící i-uzly a data. Snaží se vždy alokovat datové bloky ve stejné skupině bloků, kde jsou i příslušné i-uzly. Při alokaci bloku Ext2 předalokuje až 8 přilehlých bloků. To má za následek zrychlení zápisu při velké zátěži. Podle článku [9] je úspěšnost předalokace okolo 75 % na značně zaplněných souborových systémech. Optimalizace alokace způsobují dobrou lokalitu souvisejících bloků a souborů, tím je ovlivněna i rychlost čtení takto uložených bloků.

2.2 Ext3

Má stejný formát dat na disku jako Ext2, souborový systém Ext2 tedy může být připojen jako Ext3 a naopak. Díky této vlastnosti je možné přejít z Ext2 na Ext3 bez vytváření nového souborového systému [15]. Podle tohoto článku je také možné tento převod provést i když je souborový systém připojený. Přechod z Ext2 na Ext3 je snadno vratná akce, lze tedy bez problémů přejít i z Ext3 na Ext2.

2.2.1 Žurnálování

Žurnálování je způsob, jak se vyhnout poškození souborového systému pomocí udržování žurnálu. Žurnál je speciální soubor pro ukládání změn v souborovém systému určených k provedení. Změny jsou ukládány v kruhovém bufferu [12].

Ext3 umožňuje žurnálování nejen metadat, ale i dat. Žurnálování je zajištěno použitím vrstvy JBD (Journaling Block Device). Souborový systém informuje JBD o změnách které

provádí a vyžádá si od něj povolení před modifikací dat na disku. To dává JBD možnost udržovat žurnál na žádost ovladače Ext3.

Žurnál je ukládán do i-uzlů, díky tomu nemusí být uložen v metadatech Ext3. To zaručuje zpětnou kompatibilitu s Ext2. Ext3 podporuje fyzické žurnálování, kdy jsou žurnálovány kompletní bloky. Oproti tomu například XFS implementuje logické žurnálování, kdy jsou žurnálovány rozsahy změněných bytů.

2.2.2 Výkonnost

Žurnálování přináší další režii navíc, která zpomaluje prováděné operace. Čím větší velikost žurnálu je zvolena, tím je tento efekt méně patrný. Zdroj [11] však uvádí že Ext3 je výkonnější než Ext2 z toho důvodu, že žurnálování optimalizuje pohyb hlavičky na disku. Žurnál může být v paměti a větší množství změn je pak zapsáno najednou, rychlost takové operace je pak vyšší, než kdyby byl žurnál čten z disku.

Ext3 poskytuje tři úrovně žurnálování [11]. V režimu `data=writeback` je záruka integrity dat limitována na úroveň souborového systému Ext2. Tato možnost poskytuje nejvyšší výkon [11], v případě pádu systému ale pravděpodobně dojde ke spuštění kontroly souborového systému.

Režim `data=ordered`, který je výchozí, zajišťuje pouze žurnálování metadat. V případě pádu systému tak s největší pravděpodobností nebude zapotřebí spustit kontrolu souborového systému. Metadata a data jsou rozdělena do transakcí o velikosti jeden, dva nebo čtyři kiB. Po zápisu dat transakce jsou zapsána metadata. Tím je zabezpečeno zapisování nových souborů, po pádu nebudou obsahovat nesmyslná data. Tento žurnálovací režim si ale neudržuje informace o tom, které datové bloky již byly zapsány. V případě pádu při přepisování existujících dat není vyloučen výskyt nesmyslných dat v souboru.

V režimu `data=journal` jsou žurnálovány i datové bloky. Jsou tedy udržovány dva žurnály, jeden pro data a druhý pro metadata. To může mít negativní dopad na výkon, zejména pokud jsou žurnály malé. V případě zaplnění žurnálu se musí čekat dokud nejsou změny zaneseny na disk, a pokud je žurnál malý, dochází k takovému čekání častěji.

2.3 Ext4

Ext4 je postaven na předcházejícím Ext3, tvůrci Ext4 chtěli těžit z robustnosti a spolehlivosti souborového systému Ext3. Souborový systém Ext4 byl odštěpen s cílem získat lepší škálovatelnost, výkonnost a posunout limity Ext3. Ext4 a Ext3 nejsou vzájemně natolik kompatibilní jako Ext3 a Ext2. Ext4 nově zavádí extenty, díky tomu je přechod z Ext3 na Ext4 snazší než naopak.

Maximální velikost souborového systému Ext3 je díky 32-bitovému číslování bloků omezena na 16 TiB. Ext4 zavádí 48-bitové číslování bloků. Při velikosti bloku 4 kiB je tak maximální velikost celého souborového systému jeden exbibyte (1 048 576 GiB). 48-bitové číslování bloků musí být podporováno i JBD, JBD podporující 48-bitové číslování bloků je označeno jako JBD2. Kapacita je však stále ještě omezena maximálním počtem skupin bloků. Ext4 zavádí metaskupiny bloků (metablock group). Metaskupina bloků je skupina skupin bloků taková, aby se všechny deskriptory skupin bloků v ní obsažených vešly do jediného bloku. Přidáním vrstvy metaskupin bloků je tedy teoreticky možné dosáhnout velikosti 1 EiB. Tato hodnota je však nereálná z hlediska implementace nástrojů Ext4. Například kontrola takového souborového systému nástrojem `e2fsck` by trvala řádově desítky let [14].

2.3.1 Extenty

Souborový systém Ext3 mapuje logický sektor vždy na jeden fyzický. Ext4 zavádí extenty, kdy jeden deskriptor reprezentuje rozsah fyzických bloků. Tato oblast může obsahovat až 2^{15} bloků, to je při velikosti sektoru 4 kiB 128 MiB diskového prostoru.

Extenty jsou obzvlášť výhodné pro velké souvislé soubory. Čtyři extenty mohou být uloženy přímo v i-uzlu, pokud je jich potřeba více, vzniká strom s konstantní hloubkou. Odkazy na bloky souboru jsou pak uloženy v listech tohoto stromu. Použití extentů lze povolit nebo zakázat volbou při připojování souborového systému. Extenty nejsou zpětně kompatibilní se souborovým systémem Ext3, při přechodu z Ext4 na Ext3 je tedy nutné extenty zakázat a kopírováním všech souborů zajistit jejich odstranění.

2.3.2 Změny v alokaci

Trvalá předalokace

Předalokace zrychluje operaci zápisu a zlepšuje lokalitu souborů, která přispívá k rychlejšímu čtení. Souborový systém Ext4 přichází s trvalou předalokací, která na rozdíl od předalokace u Ext2 a Ext3 přetrvá restart operačního systému. Ext2 a Ext3 řeší předalokaci pomocí rezervování bloků. Ext4 umožňuje alokovat extenty, ale neinicializovat je. V případě čistě sekvenčního zápisu by stačilo udržovat informaci o tom, kde je hranice zapsaných dat a neinicializovaných bloků. Protože však zápis nemusí být sekvenční, extent obsahuje informaci o tom, jestli je inicializován nebo nikoli. Díky tomu je možné extenty předalokovat. Ext4 tak poskytuje stejně jako třeba XFS volání `fallocate`.

Zpožděná a vícebloková alokace

Souborový systém Ext4 přináší zpožděnou a víceblokovou alokaci pro operace s využitím vyrovnávací paměti umístěné v operační paměti. Ext3 při zápisu alokuje v jednom okamžiku vždy jen jeden blok, zpožděná alokace Ext4 alokuje více bloků najednou při vyprazdňování této vyrovnávací paměti. Celý extent je tak alokován najednou místo postupné alokace všech jeho bloků.

Tento přístup není podporován VFS, zajišťuje jej ovladač souborového systému Ext4. Zpožděná vícebloková alokace má příznivý vliv na rychlost intenzívního zápisu a také redukuje fragmentaci. Nižší fragmentace je výhodná pro následné čtení.

2.3.3 Kontrolní součty a další vlastnosti Ext4

Kontrolní součty

Kontrolní součty slouží k lepší detekci poškození souborového systému. Kontrolních součtů je využito i v žurnálu, obnova z poškozeného žurnálu by mohla způsobit ještě větší poškození souborového systému. Je spočítán kontrolní součet transakce a zapsán na disk. Pokud při obnově poškozeného systému nesouhlasí kontrolní součet zapsaných metadat se součtem metadat, která měla být zapsána, nebyla metadata zapsána správně a jejich zápis by se měl opakovat. Díky zápisu kontrolního součtu metadat při zápisu dat nemusí další transakce čekat na dokončení zápisu metadat předchozí transakce. To přináší zrychlení operace zápisu.

Další vlastnosti

Ext4 zavádí přesnější časovou známku souboru. V dnešní době je rychlost procesorů taková, že je možné, aby byl jeden soubor změněn během jedné sekundy vícekrát. Rozlišení časové známky je tedy zvětšeno z jedné sekundy na jednu nanosekundu.

Maximální velikost souboru byla zvětšena na 16 TiB a byl odstraněn limit počtu podadresářů jednoho adresáře. Ext3 umožňuje adresáři mít maximálně 32 000 podadresářů [14], u Ext4 není počet podadresářů nijak omezen. Ext4 také přináší podporu defragmentace připojeného souborového systému. Lze defragmentovat jediný soubor nebo celý souborový systém, k defragmentaci je využita vyrovnávací paměť umístěná v operační paměti a dočasné i-uzly.

Dále byly přidány tabulky nepoužitých skupin bloků a i-uzlů, které by měly ovlivnit rychlost obnovení poškozeného souborového systému nástrojem `e2fsck`.

2.4 XFS

Souborový systém XFS byl poprvé uveden s operačním systémem Irix 5.3 v roce 1994 [16]. Pro GNU/Linux byl XFS uvolněn v březnu 2 000 [2]. Souborový systém XFS může mít celkovou velikost až 18 exabytů, limit velikosti souboru je 9 exabytů. Takto vysoké hodnoty jsou možné díky tomu, že XFS adresuje bloky 64 bity.

2.4.1 Alokační skupiny a B+ stromy

Při vytváření souborového systému je vytvořeno minimálně 8 stejně velkých souvislých oblastí, které jsou v XFS nazývány alokační skupiny (angl. allocation groups). Alokační skupiny slouží ke zrychlení paralelního zápisu. Každá alokační skupina si sama spravuje i-uzly a volné místo, není tedy problém zapisovat v jednom okamžiku do vícero alokačních skupin.

XFS se snaží v maximální míře využívat efektivní datové struktury B+ stromu. Alokační skupina si udržuje informaci o tom, kde lze který i-uzel nalézt na disku pomocí B+ stromu. B+ stromy jsou rovněž využity při správě volného místa. Zde jsou využity dva B+ stromy, XFS si udržuje informaci o volném místě jednak podle jeho umístění na disku, tak i podle jeho velikosti. Rychlé hledání volného místa urychluje operaci zápisu.

2.4.2 Zpožděná alokace a žurnálování

XFS využívá zpožděnou alokaci, kdy je při zápisu pouze rezervováno místo na disku, ale není ještě rozhodnuto, které bloky se použijí. Zapisuje se pouze do vyrovnávací paměti umístěné v operační paměti a na disk jsou data zapsána až při vyprazdňování této vyrovnávací paměti. To pozitivně ovlivňuje rychlost zápisu. Zpožděná alokace je obzvláště výhodná při zápisu velkého množství malých souborů, které jsou po krátké době smazány. V takovém případě nemusí být tyto soubory na disku vůbec vytvářeny.

Souborový systém XFS žurnáluje pouze metadata, na rozdíl od Ext3 a Ext4 je použito logické žurnálování. Nejsou žurnálovány celé bloky, využívá vlastní formát k ukládání informací o změnách v metadatech na disk. Díky tomu může být velikost žurnálu menší než u Ext3 a Ext4.

2.4.3 Další vlastnosti XFS

Souborový systém XFS má řadu dalších zajímavých vlastností, které už ale nejsou z hlediska jeho výkonnosti tak významné. Patří mezi ně rozšířené atributy, kvóty a zálohování souborového systému [3].

Jsou podporovány rozšířené atributy pro všechny typy i-uzlů (běžné soubory, adresáře, symbolické odkazy a speciální soubory). Rozšířené atributy jsou dvojice klíč-hodnota a jsou děleny do třech úrovní. Uživatelské rozšířené atributy jsou přístupné všem uživatelům, kteří mají přístup k tomuto souboru. Systémové atributy jsou přístupné superuživateli a slouží například pro uložení rozšířených práv k souboru (angl. access control list). Bezpečnostní rozšířené atributy jsou využívány bezpečnostními moduly jako SELinux.

XFS podporuje nastavení kvót pro uživatele a skupiny. Kvóta je považována za metadata souborového systému a je žurnálována. To usnadňuje kontrolu kvóty při opravě poškozeného souborového systému.

Souborový systém XFS disponuje nástroji pro zálohování a obnovení celého souborového systému. Jedná se o utility `xfsdump` a `xfsrestore`. Zálohy zachovávají rozšířené atributy a kvóty. Jsou také přenosné mezi různými architekturami a dokonce i mezi operačními systémy GNU/Linux a Irix.

2.5 Btrfs

Btrfs je souborový systém, který je v současné době ve fázi vývoje. Není běžným rozšířeným souborovým systémem, ani není přímo podporován žádnou verzí Red Hat Enterprise Linux. Je postaven na řadě zajímavých myšlenek, které mu dávají ambice vytlačit Ext3 a Ext4 z pozice výchozích linuxových souborových systémů. Z toho důvodu jsem se rozhodl jej pro srovnání zařadit do této práce.

2.5.1 Principy a základní vlastnosti

Btrfs organizuje všechna data na disku do B-stromu [8]. Všechny objekty (i-uzly, položky v adresáři, rozsah dat souboru, data souborů) jsou organizovány do jednoho B-stromu. Objekty jsou rozlišovány podle jedinečného čísla objektu (obdoba čísla i-uzlu) a svého typu. Pomocí těchto dvou údajů je možné lokalizovat na disku libovolný objekt. Výhodou tohoto řešení je, že kód Btrfs pro manipulaci s objektem uvnitř B-stromu je nezávislý na typu objektu.

Uvnitř B-stromu (ve větvích stromu) se uzly skládají pouze z klíčů a hlaviček bloků. Klíče slouží k dohledání hledané položky a hlavička bloku určuje blok, na kterém se nachází další uzel nebo list. Listy stromu reprezentují objekty souborového systému. Soubory menší než blok listu stromu jsou uloženy přímo v tomto listu. Data většího souboru nejsou uložena přímo v B-stromu, ale v extentu mimo strom. V listu B-stromu jsou uloženy informace o tomto extentu. Inody jsou relativně malé, neobsahují žádná přídavná data jako třeba rozšířené atributy. Ty jsou uloženy v B-stromu zvlášť a na jiném místě.

Souborový systém Btrfs implementuje mechanismus kopie při zápisu (angl. copy on write). Data nejsou nikdy měněna na místě kde se nachází, při změně jsou vždy kopírována jinam. Zajímavou vlastností Btrfs je, že nerozlišuje bloky na disku na bloky pro data a bloky pro metadata. Jeden blok může obsahovat zároveň data i metadata. Metadata jsou ukládána ze začátku bloku, data od konce. Volné místo v bloku se tak zmenšuje z obou stran. To je výhodné jednak z hlediska úspory místa, ale také času. Pro čtení konkrétních dat nemusí

být čteno velké množství bloků ležících na různých místech disku. Díky tomu není potřeba tolik přesunů hlaviček na disku.

V dokumentaci návrhu Btrfs [6] je uvedeno, že cílem je dosáhnout dobré výkonnosti u dlouhodobě používaného a rozrůstajícího se souborového systému. Tato vlastnost má být důležitější než krátkodobá výhra v nějakém konkrétním programu pro měření výkonnosti.

2.5.2 Další vlastnosti

Každý z extentů obsahuje kontrolní součet, počet na něj vedoucích odkazů a zpětné odkazy na extenty, které na něj ukazují. Podle článku [8] jsou zpětné odkazy hlavní výhodou Btrfs, protože umožňují efektivní přesouvání dat. Díky tomu například Btrfs umožňuje snadné zmenšení velikosti souborového systému.

Mezi další vlastnosti Btrfs patří maximální velikost souboru 2^{64} B, dynamická alokace i-uzlů, zapisovatelné zálohy, komprese, vestavěná podpora více disků, efektivní inkrementální zálohy a kontrola a defragmentace připojeného souborového systému [5].

Kapitola 3

Volně šiřitelné programy pro měření výkonnosti

Je dostupná celá řada volně šiřitelných programů pro měření výkonnosti souborových systémů (angl. benchmark), řada z nich je ale zastaralá. V této kapitole bych chtěl nejprve uvést základní rozdělení programů pro měření výkonnosti a některé blíže popsat. Popíši programy nejzajímavější pro testování výkonnosti současných souborových systémů. Takový program by neměl být moc zastaralý, nebo by měl být konfigurovatelný ze strany uživatele. Staré a nekonfigurovatelné programy pro měření výkonnosti nejsou schopny dnešní souborové systémy dostatečně zatížit a jejich užitím by byly získány výsledky neodpovídající zátěži, jaké jsou souborové systémy dnes běžně vystavovány. Závěry vyvozené z takovýchto výsledků by se pak pravděpodobně lišily od závěrů vyvozených z výsledků testů odpovídajících dnešní zátěži.

3.1 Rozdělení programů pro měření výkonnosti

Studie [17] rozděluje programy pro měření výkonnosti na makrobenchmarky, programy založené na přehrávání nahraných stop (angl. replying traces) a mikrobenchmarky.

3.1.1 Makrobenchmarky

Makrobenchmarky napodobují zatížení souborového systému typické pro některou oblast nasazení počítačů. Může se jednat o webový server, poštovní server, databázový server nebo stanici používanou k vývoji software. Používají se pro získání představy o souhrnné výkonnosti souborového systému pro daný typ zátěže. Makrobenchmarky prování různé operace nad souborovým systémem, mohou je provádět v určeném pořadí pro realističtější napodobení konkrétního typu zátěže. Dále mohou pracovat paralelně pomocí vytvoření více procesů nebo vláken.

3.1.2 Přehrávání stop

Přehrávání stop má blízko k makrobenchmarkům, cílem je dosáhnout zátěže typické pro nějaké prostředí. U makrobenchmarků bývá tato zátěž vytvářena synteticky. Zde jsou přehrávány stopy získané v tomto prostředí. Stopy by měly být pro daný typ zátěže reprezentativní. Stopy mohou být nahrány na různých úrovních, nejčastěji na úrovni systémových

volání. Problémem bývá reprodukovatelnost testů, nahrané stopy dosahují velikostí od jednotek po stovky GiB a problematické tedy může být jejich dostupnost a šíření. Existuje více možností jak stopy přehrávat, mohou být přehrávány maximální možnou rychlostí, nebo s časováním, s jakým byly nahrány.

3.1.3 Mikrobenchmarky

Mikrobenchmarky nejčastěji slouží ke zdůraznění nějaké vlastnosti souborového systému za určitých podmínek. Často je testováno pouze několik málo operací nad souborovým systémem. Mikrobenchmarky mohou být například použity k podrobnějšímu zkoumání výsledku makrobenchmarku nebo ke sledování dopadu změny v implementaci souborového systému na jeho výkonnost.

3.2 Iozone

Iozone bych z hlediska předchozího rozdělení klasifikoval jako rozsáhlý mikrobenchmark. Iozone podporuje následující operace : čtení a zápis, opakované čtení a zápis, čtení a zápis voláními `fread` a `fwrite`, čtení pozpátku, asynchronní čtení a zápis, paměťové mapování souboru, čtení a zápis s offsetem a další [1]. Iozone umožňuje i měření v rámci několika procesů nebo vláken. Tento program pro testování výkonnosti funguje pod řadou operačních systémů, je podporováno několik na UNIXu založených operačních systémů, Solaris, MAC OS a řada verzí Windows. Iozone je dále aktivně vyvíjen[1].

Uživatel specifikuje maximální velikost souboru a bloku, po kterých je operace prováděna. Tyto hodnoty lze volit tak, aby se zapisovaná data vešla a nebo nevešla do vyrovnávací paměti. Je také možné zvolit jaké operace budou prováděny. Porovnáním výsledků různých souborových systémů bude možné zjistit, jak je který souborový systém výkonný pro určitou kombinaci parametrů.

Výsledky měření jsou dostupné v textovém formátu jako tabulka. Protože tato tabulka obsahuje data pro všechny kombinace velikostí souboru a bloku, je vhodné ji dále zpracovat. Ve firmě Red Hat je ke zpracování výsledků Iozone v současné době používána sada skriptů v jazyce Perl, jejichž výstup je opět v textovém formátu. Technický konzultant této práce RNDr. Jiří Hladký začal v roce 2010 přepisovat tyto skripty do jediného nástroje v jazyce Python, s grafickým výstupem. Jako součást této práce byla vytvořena sada zhruba šedesáti patchů na tento nástroj. Více informací bude uvedeno v sekci 5.2.

3.3 Postmark

Postmark je program pro testování výkonnosti navržený k tomu, aby imitoval zátěž serveru použitého pro elektronickou poštu, webové zprávy nebo elektronický obchod [13]. Taková zátěž souborového systému spočívá ve velkém množství pomíjivých malých souborů, typicky stovky tisíc až miliony souborů s velikostí v řádu jednotek až stovek kiB, které se neustále mění.

V době, kdy byl Postmark vytvořen již existovala řada různých programů pro měření výkonnosti, žádný však nebyl orientován na webové služby. Měřit výkonnost bylo možné na produkčních systémech, zátěž však byla nedeterministická. S ohledem na reprodukovatelnost testů byl tedy vytvořen Postmark.

[1] V době psaní této práce byla poslední změna zdrojového kódu Iozone datována k 26. 4. 2011

Postmark provádí takzvané transakce. Každá transakce se skládá ze dvou podtransakcí. V jedné je vytvořen nebo smazán soubor a v druhé soubor čten nebo je k jeho obsahu přidáván další. Soubory jsou voleny náhodně, aby se omezil efekt využití vyrovnávací paměti. Soubor je vytvořen s určitou počáteční velikostí a pak je zvětšován postupným přidáváním náhodného objemu dat až do maximální velikosti. Čteny jsou soubory celé. Na konci běhu jsou všechny zbývající soubory smazány, rychlost tohoto mazání je také zahrnuta do výsledků.

Výhodou programu Postmark je vestavěný generátor pseudonáhodných čísel. Výsledky naměřené na různých architekturách a operačních systémech tak nejsou ovlivněny použitím různých generátorů pseudonáhodných čísel. Postmark se konfiguruje pomocí příkazů, kterými lze například zvolit poměry provádění jednotlivých operací nebo zapnout či vypnout využití vyrovnávací paměti. Konfigurační příkazy mohou být uloženy a načteny z konfiguračního souboru. Nevýhodou je, že neexistuje žádná standardní konfigurace [17]. Výsledky naměřené za použití rozdílných konfigurací se budou pravděpodobně lišit.

Postmark funguje na operačních systémech typu UNIX a na Windows. Později byl vytvořen klon Postmark pojmenovaný Filemark, který je rozšířen o podporu běhu ve více procesech a přesnější měření času.

3.4 FileBench

FileBench je framework pro simulaci zatížení souborového systému různými aplikacemi [4]. Byl vytvořen pro operační systém OpenSolaris, port pro GNU/Linux se nazývá FSL. V dokumentaci [4] z roku 2010 je označen jako ve vývoji. V roce 2010 ale proběhla akvizice SUN Microsystems firmou Oracle a samotný fakt, že dokumentace nebyla změněna od dubna 2010 může svědčit o tom, že vývoj programu FileBench nepokračuje.

Chování se specifikuje pomocí vlastního jazyka pro popis zatížení (angl. workflow definition language). Uživatel specifikuje proměnné určující parametry zátěže, jaké soubory se mají použít a procesy včetně vláken, které budou vykonávat požadovanou činnost. Díky svým možnostem nastavení dokáže FileBench emulovat jiné programy pro měření výkonnosti.

Příkazy pro konfiguraci mohou být uloženy do souboru a distribuovány pro zajištění reprodukovatelnosti testu. FileBench je distribuován s řadou konfiguračních souborů. Z makrobenchmarků to jsou emulace poštovního serveru, souborového serveru, databázového serveru, webového serveru a proxy serveru. Přiložené konfigurační soubory pro mikrobenchmarky obsahují sekvenční čtení a zápis do jednoho souboru, víceprocesové náhodné nebo sekvenční čtení a zápis do vícero souborů, kopírování adresářového stromu a tvorbu adresářového stromu a souborů v něm.

FileBench podporuje zadání statistického rozložení, které má splňovat vytvářený adresářový strom. Rozložením lze rovněž specifikovat velikost souborů, nemusí tedy mít všechny stejnou velikost. Kromě podpory běhu ve vícero vláknech FileBench poskytuje i prostředky pro jejich synchronizaci.

Dokumentace uvádí doporučená nastavení pro velké a malé systémy. Pro malý systém (jeden až dva pevné disky) je doporučeno 50 000 souborů pro makrobenchmarky a soubor o velikosti 1 GiB pro mikrobenchmarky a emulaci databáze. Pro velké systémy (více než dvacet pevných disků) dokumentace doporučuje použít 100 000 souborů a velikost souboru 5 GiB. Pro náhodné čtený a zápis je u velkých systémů doporučeno použít 256 vláken.

Kapitola 4

Návrh výkonnostních testů souborových systémů

V této kapitole bude zdokumentováno použité počítačové vybavení, nastavení operačního systému a použité parametry programu Iozone. Následuje popis existujících Beaker testů sloužících pro spouštění programu Iozone.

4.1 Testovací hardware

Cílem této práce je porovnávání výkonnosti různých souborových systémů, nejde o porovnávání výkonnostních vlastností použitých úložných médií ani jiných parametrů použitého hardware. Základní popis komponent použitého počítače uvádím zejména z důvodu reprodukovatelnosti testů.

Testování probíhalo na počítači HP Proliant DL360 G6. Verze BIOSu je označena jako P64 a pochází z 6. února 2009. Testovací počítač byl vybaven procesory Intel Xeon X5550 (Quad-Core), celkem 16 jader o frekvenci 2,67 GHz. Použité procesory disponují vyrovnávací pamětí o velikosti 128 kiB v úrovni L1, 1 MiB v L2 a 8 MiB v úrovni L3. V testovacím počítači je nainstalováno 18 paměťových modulů DIMM DDR3 s kapacitou 2 GiB, pracujících na frekvenci 1 333 MHz.

Testovací počítač je vybaven dvěma pevnými disky připojenými přes HP Smart Array P410i Controller. Smart Array Controller má nainstalovaný firmware ve verzi 1.62 a pracuje se sektory o velikosti 8 kiB. Oba disky mají velikost 500,07 GB.

4.2 Nastavení operačního systému

Nastavení operačního systému popisované v této sekci vychází z doporučení technického konzultanta a autor této práce není výhradním autorem tohoto nastavení. Zde popsaná konfigurace je inspirována nastavením používaným firmou Red Hat před spouštěním programu Iozone.

Před spuštěním testu je vypnuto odkládání stránek na disk (swapování) a spuštěn program, který alokuje paměť a zapisuje do ní až do okamžiku, kdy zůstane volné místo o zadané velikosti. Jednodušším řešením omezení velikosti operační paměti je použití příslušného parametru jádra. V takovém případě je ale skutečná velikost používané operační paměti menší o paměť využívanou jádrem. Program snižující velikost volné operační paměti je tedy využíván proto, že díky faktu, že je spouštěn až po startu operačního systému

zajistí přesnější hodnotu velikosti volné operační paměti před spuštěním testu.

Ještě před omezením volné paměti jsou zastaveny služby `cron` (automatické spouštění naplánovaných úloh), `anacron` (obdoba cron), `sendmail` (transportní agent elektronické pošty), `avahi-daemon` (mDNS démon pro konfiguraci síťových zařízení , `ntpd` (síťová konfigurace systémového času) a všechny procesory jsou po celou dobu běhu testu nastaveny na maximální frekvenci. Před vlastním spuštěním testu je vynuceno vyprázdnění vyrovnávací paměti.

Cílem výše uvedeného nastavení je minimalizovat faktory, které by mohly v průběhu testu způsobit zátěž testovacího počítače. Takovéto, byť i jednorázové zatížení by mohlo ovlivnit výsledky testu. Stejně tak by výsledky mohly být ovlivněny změnou systémového času v průběhu testu. Velikost volné operační paměti je snižována za účelem zmenšení kapacity vyrovnávací paměti vstupně-výstupních operací.

Při testování výkonnosti souborových systémů je potřeba minimalizovat výkyvy ve výkonu použitého datového média. Operační systém je nainstalován na jednom pevném disku a testovaní probíhá na jiném disku. Tím je zajištěno, že na testovacím disku nedochází během testu k provádění jiných, s testem nesouvisejících operací. Velikost testovacího diskového oddílu omezena na 20 GiB a testovací oddíl při všech testech začíná na prvním sektoru. Použitím pouze malé oblasti disku jsou minimalizovány rozdíly v počtu sektorů na stopě, rozdíly v rotačním zpoždění a také snížena vystavovací doba hlaviček disku. Limit 20 GiB je v poměru s velikostmi v praxi používaných souborových systémů velmi malý, výhody využívání stále stejné oblasti disku ale tuto nevýhodu převažují. Pro ukládání skutečně velkých dat se ale lokální souborové systémy nepoužívají a dalším faktorem, který mluví proti testování na velkých souborových systémech a souborech, je prodloužení doby běhu testu.

4.3 Iozone v systému Beaker

Implementace skriptů pro spouštění Iozone v prostředí Beaker již existuje ve firmě Red Hat v pěti variantách. Nejstarší varianta pochází z roku 2006, pro získání dat pro tuto práci byla využita poslední varianta `iozone_devel_with_library`, odštěpená z předchozí verze v roce 2010. Z důvodu existence několika Beaker implementací Iozone jsem nevytvářel další, ale soustředil se na vytvoření nástroje pro zpracování výstupů existujících Beaker testů. Použitý test `iozone_devel_with_library` pouze stručně popíši, popis vytvořeného nástroje pro zpracování výsledků Iozone lze najít v sekci 5.2.

Beaker test `iozone_devel_with_library` je sadou obalujících skriptů, které v prostředí Beaker nastavují systém a spouští program Iozone. `Iozone_devel_with_library` využívá pomocný test `/performance/common_functions/lib`, který obaluje funkce pro vytváření a připojování souborových systémů. Tento pomocný test je využíván pro Beaker implementaci programů pro měření výkonnosti Iozone a Postmark, jeho cílem je zajistit použití vždy stejných parametrů standardních příkazů pro vytváření a připojování souborových systémů. Nejsou nastavovány žádné hodnoty parametrů pro dosažení lepšího výkonu pro nějaký specifický případ, `/performance/common_functions/lib` používá výchozí parametry s vynucením tvorby souborového systému a připojení s vypnutím bariér. Viz tabulka 4.1. Bariéry slouží k zabránění změny pořadí provádění transakcí uložených ve vyrovnávací paměti disku. Používají se u žurnálovacích souborových systémů a jejich cílem je zabránit zápisu bloků metadat před dokončením zápisu žurnálu. Bariéry tedy přispívají k udržování konzistence žurnálu, mají ale negativní dopad na výkon souborového systému, podle

článku [10] až o 30 %. Bariéry jsou vypínány z toho důvodu, že použití výsledků naměřených se zapnutými bariérami by ovlivnilo výsledky porovnání s nežurnálovacím Ext2.

Souborový systém	**parametry mkfs**	**parametry mount**
Ext2	-F	
Ext3	-F	-o barrier=0
Ext4	-F	-o barrier=0
XFS	-f	-o nobarrier

Tabulka 4.1: Použité parametry testovaných souborových systémů

Beaker iozone test byl spouštěn s parametry `--rsync --iozone_rec_size --eatmem -r 5 --swap --incache --directio --outofcache -f ext2,ext3,ext4,xfs`.

- Parametr `-r` specifikuje, že test má proběhnout pětkrát, aby jeho výsledky získaly ze statistického hlediska nějakou výpovědní hodnotu. Mezi jednotlivými běhy je souborový systém odpojen a znovu připojen, aby došlo k vyprázdnění vyrovnávací paměti.
- Parametr `--rsync` způsobuje přenos výsledků testu na k tomu určený server firmy Red Hat.
- Parametr `--eatmem` aktivuje před spuštěním Iozone program pro zabrání volné operační paměti do zadaného limitu. Při sběru dat pro tuto práci bylo ponecháno výchozí nastavení programu eatmem, které ponechává volné místo o velikost 512 MiB.
- Parametrem `--swap` je zvoleno vypnutí swapování před testem.
- `--iozone_rec_size` způsobuje, že Iozone samo zvolí množinu velikostí bloku pro všechny velikosti souboru.
- Parametry `--incache`, `--outofcache` a `--directio` nastavují velikosti zapisovaných souborů. Při volbě incache jsou testovány operace se soubory velikostech od 4 kiB po 4 GiB, volba outofcache testuje pouze soubory o velikosti v rozsahu 512 MiB až 2 GiB. Volba directio testuje stejné velikosti souborů jako incache, na rozdíl od volby incache ale vypíná použití vyrovnávací paměti nastavením příznaku `O_DIRECT`. Při testování souborových systémů bez použití vyrovnávací paměti navíc nejsou prováděny operace `fwrite`, `frewrite`, `fread` a `freread`.

Konkrétně, při testování souborového systému Ext3 je program Iozone spouštěn s následujícími parametry.

- incache: `iozone -U /RHTSspareLUN1 -a -f /RHTSspareLUN1/ext3 -n 4k -g 4096m`
- outofcache: `iozone -U /RHTSspareLUN1 -a -f /RHTSspareLUN1/ext3 -n 512m -g 2048m`
- directio: `iozone -U /RHTSspareLUN1 -a -f /RHTSspareLUN1/ext3 -I -g 4096m -i 0 -i 1 -i 2 -i 3 -i 4 -i 5`

Kapitola 5

Interpretace výsledků

V této kapitole budou nejprve stručně uvedeny používané statistické termíny a představen vytvořený nástroj pro zpracování výstupů Iozone. Dále bude následovat vlastní porovnání souborových systémů za použití tohoto nástroje. Na základě výsledků Iozone budou navzájem porovnány současné souborové systémy ve výchozí konfiguraci. Dále budou tyto souborové systémy porovnány se souborovým systémem Btrfs, který je v době psaní této práce ve fázi vývoje. Výkonnost souborových systémů ve výchozí konfiguraci bude porovnána s výkonností týchž souborových systémů se změněnou konfigurací. Budou také uvedeny výsledky porovnání souborových systémů ve výchozí konfiguraci programem Postmark.

5.1 Použité statistické termíny

V této kapitole budou uvedeny statistické termíny, které budou později v textu využívány. Nejedná se o kompletní definice, ale pouze o základní vysvětlení významu těchto termínů.

- *Aritmetický průměr* je nejzákladnější statistickou veličinou popisující sadu hodnot. Hodnota aritmetického průměru se vypočte jako $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.
- *Standardní odchylka* je hodnota, která se vypočítá ze statistického rozptylu hodnot a jejím cílem je určit, jak moc se navzájem liší hodnoty v rámci jedné sady dat.
- *Interval spolehlivosti* je rozmezí hodnot, kde se s určenou hodnotou pravděpodobnosti nachází skutečná hodnota dané veličiny. V této práci jsou používány 90% intervaly spolehlivosti výskytu hodnoty aritmetického průměru. Velikost intervalu spolehlivosti tedy souvisí s hodnotou standardní odchylky.
- *Rozložení pravděpodobnosti* náhodné veličiny je funkce, která každé hodnotě náhodné veličiny určuje pravděpodobnost, s jakou tato hodnota může nastat. Pro tuto práci je důležité, že některé statistické veličiny popisující data v datové sadě mohou být zavádějící, pokud tato datová sada nesplňuje nějaké rozložení.
- *Geometrický průměr* je statistickou veličinou podobnou aritmetickému průměru, využívá ale místo sčítání násobení a místo dělení operaci odmocňování. Geometrický průměr je vhodný pro jiné druhy dat než aritmetický a naopak. Předpis pro výpočet geometrického průměru je $\overline{x}_g = (\prod_{i=1}^{n} x_i)^{\frac{1}{n}}$.

- *Percentil* je druhem kvantilu. Kvantily jsou hodnoty, které dělí seřazenou sadu dat na určitý počet stejně velkých částí. Percentily dělí sadu dat na sto stejně velkých částí.
- *Kvartil* je opět druhem kvantilu, který ale dělí sadu dat na čtyři stejně velké části. Interval určený hodnotami prvního a třetího kvartilu určuje rozmezí, kde leží prostředních 50 % hodnot.
- *Medián* je druhým kvartilem neboli 50% percentilem, hodnotou ležící přesně veprostřed všech seřazených dat sady.
- *Hladina významnosti* je hodnota pravděpodobnosti určující zvolenou hranici mezi náhodným a statisticky významným rozdílem dvou výsledků. V této práci je používána hladina významnosti 10 %, pokud je rozdíl mezi daty obou datových sad více než 10 %, považujeme tyto sady za rozdílné.
- *Studentův t-test* slouží ke zjištění, zda-li jsou dvě sady dat statisticky rozdílné [18]. T-test porovnává obě sady podle jejich průměrů, bere však v úvahu jejich šířku (variabilitu). T-hodnota T-testu je poměrem signál/šum, kde signálem je rozdíl průměrů obou datových sad a šumem variabilita sad vypočtená z jejich rozptylu. T-hodnota je kladná v případě, že průměr první sady byl větší než průměr druhého, záporná v opačném případě. Se znalostí t-hodnoty je po určení hladiny významnosti (použito 10 %) a stupně volnosti ze statistických tabulek standardním způsobem získaná pravdivostní p-hodnota.

5.2 Nástroj pro zpracování výstupů Iozone

Jako součást této práce jsem vytvořil sadu 64 patchů na nástroj pro zpracování výstupů Iozone (dále jen nástroj). Nástroj začal vytvářet technický konzultant této práce Jiří Hladký a založil jej na skriptu na analýzu výsledků programu Netperf pro testování výkonnosti počítačové sítě. Autorem tohoto skriptu je Adam Okuliar, nástroj má tedy uvedeny tři autory. Přílohu této práce je kromě zdrojového kódu nástroje i git repozitář, aby bylo možné jasně odlišit, jaká část nástroje je prací autora této práce.

Nástroj je určen pro výpočet statistických veličin a porovnání dvou sad dat. Jeho vstupem jsou soubory `.iozone` obsahující výsledky programu Iozone v textové podobě. Tyto soubory jsou vytvářeny již existujícími Beaker Iozone testy a jejich obsah je shodný se standardním výstupem Iozone. Při spuštění skriptu je nutné zadat vstupní datové soubory pro sadu *Baseline* a pro sadu *Set1*. Obě sady jsou po vypočtení statistických veličin porovnávány proti sobě Studentovým t-testem [18], výsledek porovnání bude stejný nezávisle na tom, která sada dat bude označena jako *Baseline* a která jako *Set1*. Výstupem skriptu je webová stránka obsahující tabulky statistických hodnot a grafy. Kromě HTML výstupu nástroj také vytváří `.csv` soubory obsahující vstupní data a vypočtené statistiky. Tyto výstupy jsou importovatelné například tabulkovým procesorem OpenOffice.org Calc nebo nástrojem na statickou analýzu dat GNU R.

Skript neslouží pouze pro porovnávání různých souborových systémů, datové soubory v sadě mohou pocházet z libovolných běhů Iozone. Je plánováno, že nástroj bude sloužit ve firmě Red Hat k odhalování výkonnostních regresí souborových systémů mezi různými verzemi jádra operačního systému. Skript může být spouštěn na serveru, kde jsou ukládány výsledky Iozone Beaker testů a výstupy skriptu mohou být vzdáleně prohlíženy prostřednictvím webového prohlížeče.

Nástroj je navržen tak, aby nevyžadoval shodné parametry Iozone v obou sadách. V obou sadách tedy nemusí být testována stejná množina operací. Průměrná doba běhu programu Iozone pro jeden souborový systém s provedením všech operací podporovaných Iozone s pěti opakováními se pohybuje *v řádu dnů*. Možnost spustit Iozone jen s omezeným počtem operací a s možností takto získané výsledky porovnat s výsledky s jiným počtem operací bez jejich ruční editace může tuto dobu výrazně zkrátit. Tato vlastnost nástroje není v současné době plně implementována, data pro tuto práci obsahují výsledky pro všechny operace. Podpora různého počtu operací v porovnávaných sadách bude doimplementována po nasazení nástroje ve firmě Red Hat.

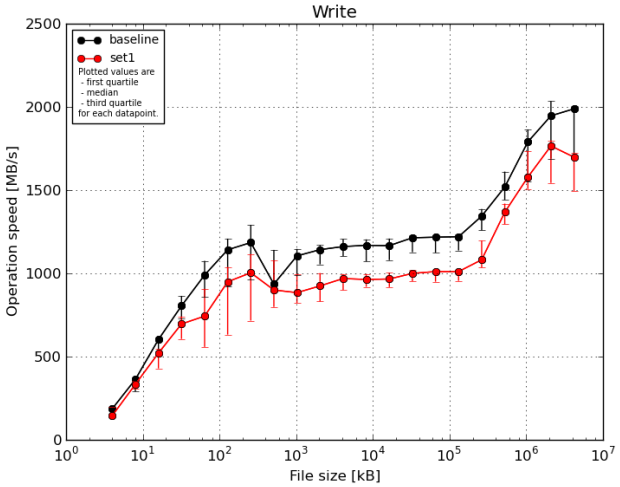
5.2.1 Normální režim

Normální režim slouží pro prvotní porovnání dvou sad dat. Normální režim je výchozím, je vyžadováno pouze zadání datových souborů pro *Baseline* a *Set1*. Výstupem je jedna HTML stránka, která se skládá ze sekcí věnovaných jednotlivým operacím a závěrečného shrnutí.

Sekce věnovaná operaci se skládá z grafu a tabulky statistických veličin popisujících data agregovaná podle velikosti bloku, podle velikosti souboru a grafu vizuálního porovnání obou datových sad s aproximací přímkou. Obrázky 5.1 a 5.2 tvoří ukázku součástí sekce věnované jedné operaci, přičemž obdobné tabulky a grafy jsou použity pro porovnání různých dvojic výsledků jedné operace. Tabulka statistických hodnot pro data agregovaná podle velikost bloku na obrázku 5.1 obsahuje pro obě datové sady statistické veličiny popisující výsledky pro danou operaci a velikost souboru. Jsou zobrazeny aritmetické průměry hodnot, jejich standardní odchylka, meze intervalu 90% spolehlivosti založeného na Studentově t-rozložení, geometrický průměr hodnot, medián (50% percentil), první kvartil (25% percentil), třetí kvartil (75% percentil) a minimální a maximální hodnota. Tabulka dat agregovaných podle velikosti souboru na obrázku 5.2 zobrazuje stejné statistické veličiny, liší se pouze v tom, že ve sloupcích jsou uvedeny velikosti bloku namísto velikostí souboru. Poslední tři řádky obou tabulek obsahují výsledky porovnání obou datových sad. Jsou zobrazeny procentní rozdíly mediánů, p-hodnota Studentova t-testu a výsledek Studentova t-testu na hladině významnosti 10 %.

Graf statistických veličin popisujících data agregovaná podle velikosti bloku zobrazuje pro oba datové sady pro všechny velikosti souboru bod určený hodnotou mediánu a interval určený hodnotami prvního a třetího kvartilu. Případné překrytí zobrazovaných intervalů značí, že nelze prohlásit data obou sad pro danou velikost souboru za rozdílná ani v případě, že hodnoty mediánů jsou značně rozdílné. Tomu ve většině případů odpovídají i příslušné výsledky Studentova t-testu na shodnost obou sad pro danou velikost souboru. Obdobným způsobem je vytvořen i graf statistických veličin popisujících data agregovaná podle velikosti souboru. Hodnoty na vodorovné ose mají význam velikostí bloku namísto velikostí souboru, jak tomu bylo v předchozím případě.

Graf vizuálního porovnání obou datových sad zobrazuje bod pro každou dvojici velikosti souboru a bloku, kde jeho souřadnicí na vodorovné ose je aritmetický průměr všech výsledků pro danou velikost souboru a bloku v *Baseline* a jeho souřadnicí na svislé ose je totéž v *Set1*. Body ležící pod černě zobrazenou přímkou $y = x$ (rychlejší v *Baseline*) jsou vykresleny červeně a body ležící nad touto přímkou (rychlejší v *Set1*) černě. Dvě červené přímky vyznačují oblast, v níž s pravděpodobností 90 % leží přímka ve tvaru $y = ax$ aproximující vykreslené body. Pod tímto grafem je zobrazen koeficient a, jeho standardní chyba a hodnoty sklonu přímky, mezi nimiž se na základě Studentova t-testu na hladině významnosti 10 % nachází skutečná hodnota koeficientu a.

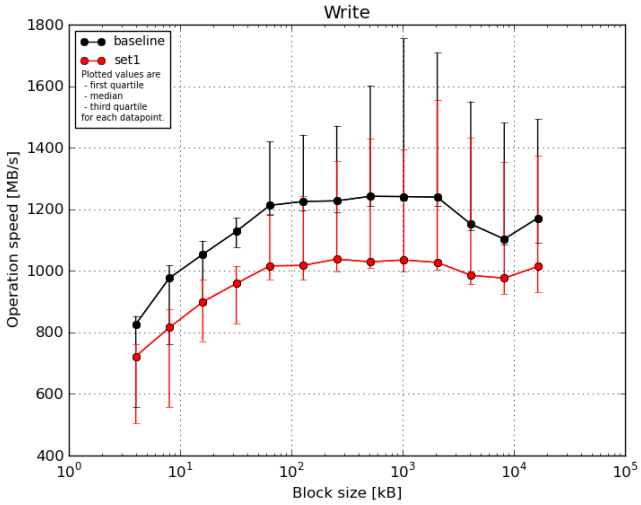


Write	File size [kB]	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536	131072	262144	524288	1048576	2097152	4194304
baseline	mean val.	187.53	307.1	559.9	788.53	905.97	1025.99	1097.1	996.05	1063.87	1102.84	1157.22	1124.94	1129.85	1183.26	1185.98	1191.12	1327.15	1527.02	1736.27	1880.63	1875.84
	standard dev.	4.57	143.03	69.03	104.14	262.55	264.3	250.22	211.12	125.97	120.49	212.53	110.01	107.28	62.59	57.85	59.23	101.98	121.3	164.49	193.51	187.21
	ci. min. 90%	183.17	224.19	528.5	748.27	816.13	944.0	1025.58	939.81	1032.31	1074.27	1109.26	1101.2	1107.64	1167.58	1171.49	1176.28	1301.6	1496.64	1695.07	1832.16	1828.95
	ci. max 90%	191.9	390.01	591.29	828.8	995.81	1107.98	1168.62	1052.3	1095.42	1131.41	1205.18	1148.67	1152.06	1198.94	1200.47	1205.95	1352.69	1557.41	1777.47	1929.1	1922.73
	geom. mean	187.49	248.03	555.43	780.98	846.41	981.03	1064.45	975.06	1055.87	1095.92	1141.49	1119.06	1124.32	1181.59	1184.57	1189.64	1323.2	1522.3	1728.36	1870.47	1866.16
	median	186.59	362.96	602.32	805.08	989.95	1141.71	1185.09	935.48	1104.98	1142.59	1161.07	1168.63	1166.61	1213.96	1217.52	1220.2	1343.87	1518.75	1790.03	1946.61	1988.54
	first quartile	184.49	290.03	512.9	727.96	857.22	921.74	961.78	891.34	989.27	1054.57	1103.13	1072.35	1079.69	1127.28	1124.58	1137.57	1258.35	1441.54	1551.63	1689.65	1704.65
	third quartile	186.59	373.19	602.32	864.85	1075.22	1208.6	1291.73	1140.99	1146.99	1169.3	1205.88	1204.19	1209.11	1230.49	1228.93	1236.65	1386.84	1611.37	1864.74	2035.28	2004.71
	minimum	184.49	50.43	381.45	482.23	248.05	399.36	584.23	676.54	755.21	831.23	822.54	820.42	826.78	1068.88	1084.69	1084.92	1102.84	1308.45	1451.74	1566.63	1556.63
	maximum	195.5	462.79	625.31	924.31	1135.78	1260.9	1351.6	1366.4	1351.34	1306.27	2093.28	1238.96	1240.39	1245.09	1252.2	1259.3	1543.92	1755.07	1974.82	2169.82	2046.01
set1	mean val.	163.86	321.6	473.65	627.66	706.69	836.92	928.5	926.32	885.57	920.09	942.0	944.16	951.87	988.95	993.31	995.17	1118.34	1361.57	1587.76	1677.46	1619.05
	standard dev.	28.98	53.45	126.43	181.94	228.37	243.75	220.24	162.99	133.75	110.91	82.89	79.1	73.42	41.69	40.73	39.79	94.38	85.3	154.45	148.76	139.1
	ci. min. 90%	136.23	290.62	416.16	557.32	628.55	761.3	865.55	882.9	852.07	893.8	923.3	927.1	936.67	978.51	983.11	985.2	1094.7	1340.2	1549.07	1640.2	1584.21
	ci. max 90%	191.48	352.59	531.15	698.01	784.83	912.53	991.45	969.74	919.07	946.39	960.71	961.22	967.07	999.4	1003.52	1005.14	1141.98	1382.93	1626.45	1714.72	1653.89
	geom. mean	161.87	316.67	443.92	585.32	660.36	795.5	898.84	912.32	875.28	913.51	938.15	940.62	948.87	988.09	992.49	994.38	1114.55	1358.96	1580.25	1670.76	1612.91
	median	145.24	333.4	520.93	696.06	743.03	949.61	1003.61	900.26	884.16	924.68	969.01	963.39	966.29	999.9	1010.85	1010.82	1081.81	1367.23	1578.25	1765.54	1698.16
	first quartile	143.96	307.14	428.96	605.76	557.98	631.98	716.1	798.25	822.44	831.41	899.47	919.17	917.74	953.09	949.83	955.12	1034.81	1294.99	1506.4	1543.22	1495.24
	third quartile	195.5	352.32	547.17	741.33	907.68	1038.6	1115.76	1078.08	997.03	1001.77	995.63	996.81	1003.83	1019.2	1021.81	1019.84	1197.08	1419.07	1734.38	1799.07	1725.29
	minimum	139.08	190.73	96.55	165.33	264.84	371.1	442.47	595.98	588.61	693.74	724.88	721.49	749.8	911.5	919.7	916.96	984.69	1173.88	1288.72	1438.83	1374.14
	maximum	195.5	373.19	596.84	820.2	975.22	1106.55	1195.9	1210.96	1219.7	1140.25	1076.44	1061.99	1049.94	1058.79	1054.09	1055.67	1309.81	1538.97	1814.9	1835.59	1752.88
baseline set1 difference		-22.16 %	-8.14 %	-13.51 %	-13.54 %	-24.94 %	-16.83 %	-15.31 %	-3.76 %	-19.98 %	-19.07 %	-16.54 %	-17.56 %	-17.17 %	-17.63 %	-16.97 %	-17.16 %	-19.5 %	-9.98 %	-11.83 %	-9.3 %	-14.6 %
ttest p-value		0.1088	0.7673	0.0279	0.0015	0.0062	0.0056	0.0039	0.1022	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ttest equality		SAME	SAME	DIFF	DIFF	DIFF	DIFF	DIFF	SAME	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF

See raw data iwrite_BS_baseline.csv and iwrite_BS_set1.csv.

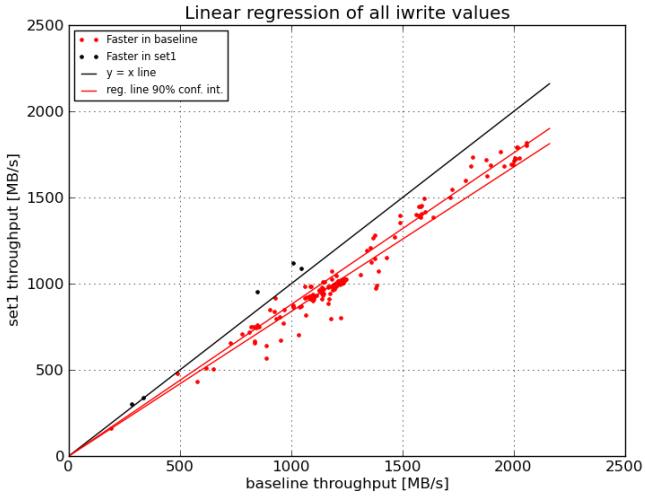
Obrázek 5.1: Ukázka grafu a tabulky popisující data agregovaná podle velikosti bloku – části sekce věnované jedné operaci v normálním režimu

Závěrečné shrnutí se skládá z tabulky a grafu zobrazujících statistické veličiny popisujících data agregovaná podle velikosti bloku a souboru, tedy všechna data příslušící k jedné operaci. Pseudooperace *ALL* navíc reprezentuje data agregovaná podle operace, tedy všechna data sady. Ukázkou závěrečného shrnutí je obrázek 5.3. Tabulka je obdobná jako tabulky v sekcích věnovaných jednotlivým operacím. Zobrazuje stejné statistické veličiny, mezi dolními řádky zobrazujícími srovnání jednotlivých datových sad přibyl řádek s intervalem, ve kterém se s 90% pravděpodobností nachází hodnota sklonu regresní přímky z vizuálního porovnání sad. Graf závěrečného shrnutí zobrazuje minimální hodnotu, první kvartil, medián, třetí kvartil a maximální hodnotu pro všechny operace formou Box-and-Whisker grafu [19]. Hodnoty zobrazené v závěrečném shrnutí jsou značně agregovány, jsou patrné pouze velké rozdíly mezi datovými sadami u jednotlivých operací. Tato vlastnost značně agregovaných dat se nejvíce projevuje u pseudooperace *ALL*, kde je vidět minimální rozdíl mezi *Baseline* a *Set1*. Z grafu ve shrnutí jsou ale dobře patrné rozdíly v rychlosti provádění jednotlivých operací a statistický rozptyl dat operací.



Write	Block size [kB]	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
baseline	mean val.	698.04	872.03	996.26	1067.98	1331.88	1354.78	1362.35	1411.75	1449.84	1444.45	1327.21	1260.65	1285.8
	standard dev.	248.07	243.34	171.97	209.75	323.59	328.36	339.31	332.78	340.04	319.29	242.72	210.96	212.59
	ci. min. 90%	646.68	819.53	957.45	1018.25	1273.5	1293.67	1297.09	1345.43	1379.45	1375.57	1272.44	1210.63	1232.55
	ci. max 90%	749.39	924.52	1035.06	1117.72	1390.25	1415.88	1427.62	1478.06	1520.23	1513.33	1381.98	1310.66	1339.05
	geom. mean	623.78	813.03	979.38	1038.3	1290.96	1315.06	1323.97	1377.03	1414.74	1413.17	1306.78	1244.13	1269.04
	median	826.78	976.36	1053.74	1128.11	1212.57	1225.53	1227.42	1242.14	1241.31	1239.46	1152.18	1103.26	1171.39
	first quartile	556.31	762.01	891.34	1075.84	1181.41	1195.79	1191.17	1210.54	1228.2	1210.42	1132.79	1085.34	1091.81
	third quartile	853.33	1017.8	1097.7	1172.47	1421.07	1441.8	1471.45	1601.38	1755.07	1708.33	1550.23	1481.02	1493.61
	minimum	50.43	52.78	602.32	297.62	323.71	463.23	716.33	927.61	1073.56	1153.42	1106.18	1042.75	1060.02
	maximum	992.12	1185.09	1268.22	1329.32	2030.52	2019.13	2090.31	2169.82	2158.5	2107.19	1773.13	1602.1	1595.07
set1	mean val.	622.66	715.6	845.56	904.2	1113.99	1134.57	1198.75	1189.09	1220.62	1234.79	1147.76	1117.09	1143.26
	standard dev.	220.06	220.52	188.72	194.58	306.87	312.56	300.9	320.48	315.82	322.81	244.97	215.99	223.19
	ci. min. 90%	577.1	668.02	802.97	858.06	1058.63	1076.41	1140.87	1125.23	1155.24	1165.15	1092.49	1065.88	1087.36
	ci. max 90%	668.21	763.17	888.15	950.33	1169.35	1192.73	1256.62	1252.96	1286.0	1304.43	1203.04	1168.3	1199.16
	geom. mean	564.12	668.03	818.41	872.27	1071.78	1095.94	1166.09	1151.76	1184.9	1197.7	1124.15	1097.45	1122.39
	median	722.26	815.77	899.13	958.86	1016.01	1017.76	1038.05	1029.26	1035.42	1027.3	985.77	976.31	1014.24
	first quartile	504.88	557.56	771.79	828.63	971.47	970.59	998.77	1008.7	998.75	1002.95	955.89	925.22	930.19
	third quartile	762.48	874.87	970.38	1015.63	1183.23	1242.26	1355.6	1428.28	1395.92	1554.03	1432.71	1353.84	1374.14
	minimum	139.08	96.55	264.84	180.59	315.53	612.63	856.89	755.24	861.27	837.87	925.9	885.07	904.46
	maximum	915.86	1028.21	1110.89	1187.65	1800.31	1785.83	1811.8	1835.59	1828.72	1810.79	1565.66	1457.42	1461.99
baseline set1 difference		-12.64 %	-16.45 %	-14.67 %	-15.0 %	-16.21 %	-16.95 %	-15.43 %	-17.14 %	-16.59 %	-17.12 %	-14.44 %	-11.51 %	-13.42 %
ttest p-value		0.0692	0.0003	0.0	0.0001	0.0	0.0	0.0021	0.0001	0.0001	0.0005	0.0002	0.0011	0.0026
ttest equality		DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF

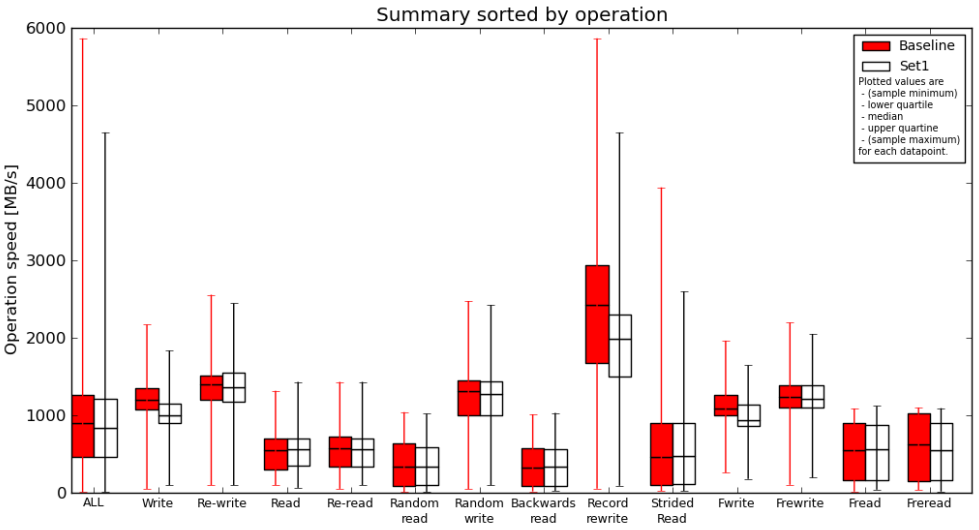
See raw data iwrite_FS_baseline.csv and iwrite_FS_set1.csv.



See comparison_of_averages.csv.

Regression line	
slope	0.85929
std. error	0.01233
ci. max 90%	0.8389
ci. min. 90%	0.87968

Obrázek 5.2: Ukázka grafu a tabulky popisující data agregovaná podle velikosti souboru a grafu lineární regrese dat z obou porovnávaných datových sad – části sekce věnované jedné operaci v normálním režimu



	Operation	ALL	Write	Re-write	Read	Re-read	Random read	Random write	Backwards read	Record rewrite	Strided Read	Fwrite	Frewrite	Fread	Freread
baseline	mean val.	952.81	1231.01	1387.1	505.99	543.81	410.68	1301.17	387.22	2451.92	617.3	1119.48	1241.77	562.18	626.86
	standard dev.	737.06	364.56	422.72	261.08	294.79	331.0	472.39	307.81	1098.92	641.87	269.75	318.96	349.73	395.82
	ci. min. 90%	941.03	1209.98	1362.71	490.93	526.8	391.59	1273.92	369.47	2388.54	580.27	1103.92	1223.37	542.01	604.03
	ci. max 90%	964.59	1252.04	1411.48	521.05	560.81	429.77	1328.41	404.98	2515.31	654.32	1135.04	1260.17	582.35	649.69
	geom. mean	644.99	1164.85	1311.19	408.31	430.83	249.26	1205.49	248.42	2171.38	325.84	1084.02	1192.68	410.99	443.44
	median	905.82	1201.91	1401.09	549.69	572.26	332.37	1312.21	324.69	2427.89	462.69	1081.31	1236.67	555.55	622.86
	first quartile	466.34	1079.67	1202.15	303.64	343.22	90.4	996.77	87.31	1680.41	100.72	997.23	1105.26	159.46	155.41
	third quartile	1257.44	1346.86	1514.79	695.23	719.79	643.61	1452.32	571.68	2943.2	903.55	1258.01	1385.93	894.79	1028.47
	minimum	9.68	50.43	98.88	98.57	51.06	9.68	47.37	17.2	52.78	23.26	259.34	102.91	15.26	36.87
	maximum	5864.95	2169.82	2555.63	1315.83	1430.74	1040.27	2474.34	1011.31	5864.95	3941.27	1962.75	2196.6	1087.03	1099.63
set1	mean val.	884.4	1052.24	1366.52	510.6	521.62	384.96	1293.46	387.27	2041.91	614.4	988.57	1227.38	550.09	558.15
	standard dev.	640.56	333.19	421.88	277.98	269.84	296.6	459.92	308.75	870.22	620.56	249.54	316.08	351.24	345.67
	ci. min. 90%	874.16	1033.02	1342.18	494.56	506.05	367.85	1266.93	369.46	1991.72	578.61	974.17	1209.15	529.83	538.21
	ci. max 90%	894.63	1071.46	1390.85	526.63	537.18	402.07	1319.99	405.08	2092.11	650.2	1002.96	1245.61	570.35	578.08
	geom. mean	613.45	992.05	1286.28	394.81	422.85	244.61	1202.79	247.48	1845.33	332.25	952.66	1178.77	391.01	417.22
	median	832.46	1000.94	1358.94	558.1	559.78	336.33	1275.75	332.33	1993.63	469.04	942.31	1213.9	558.65	545.25
	first quartile	458.66	904.82	1169.9	353.1	341.1	102.94	1000.83	84.32	1498.73	112.42	868.66	1096.83	164.33	164.82
	third quartile	1209.61	1152.13	1547.14	700.64	703.55	587.21	1443.32	564.07	2301.95	896.91	1142.81	1388.48	878.57	895.64
	minimum	9.69	96.55	97.87	63.77	99.23	9.69	97.12	25.51	90.29	23.42	178.46	193.95	39.5	15.58
	maximum	4650.82	1835.59	2444.29	1430.74	1430.74	1020.38	2422.88	1030.38	4650.82	2599.18	1644.89	2049.78	1119.77	1091.27
linear regression slope 90%			0.84 - 0.88	0.96 - 1.0	1.01 - 1.04	0.93 - 0.97	0.9 - 0.94	0.97 - 1.01	1.0 - 1.01	0.8 - 0.85	0.97 - 0.99	0.86 - 0.91	0.96 - 1.01	0.98 - 1.0	0.86 - 0.91
baseline set1 difference		-8.1 %	-16.72 %	-3.01 %	1.53 %	-2.18 %	1.19 %	-2.78 %	2.35 %	-17.89 %	1.37 %	-12.85 %	-1.84 %	0.56 %	-12.46 %
ttest p-value		0.0	0.0	0.3254	0.7301	0.1131	0.0987	0.7386	0.9974	0.0	0.9263	0.0	0.3604	0.4864	0.0002
ttest equality		DIFF	DIFF	SAME	SAME	SAME	DIFF	SAME	SAME	DIFF	SAME	DIFF	SAME	SAME	DIFF

See raw data summary_sorted_by_operation_baseline.csv and summary_sorted_by_operation_set1.csv.
All data are aggregated in summary_all_baseline.csv.

Obrázek 5.3: Ukázka závěrečného shrnutí v normálním režimu

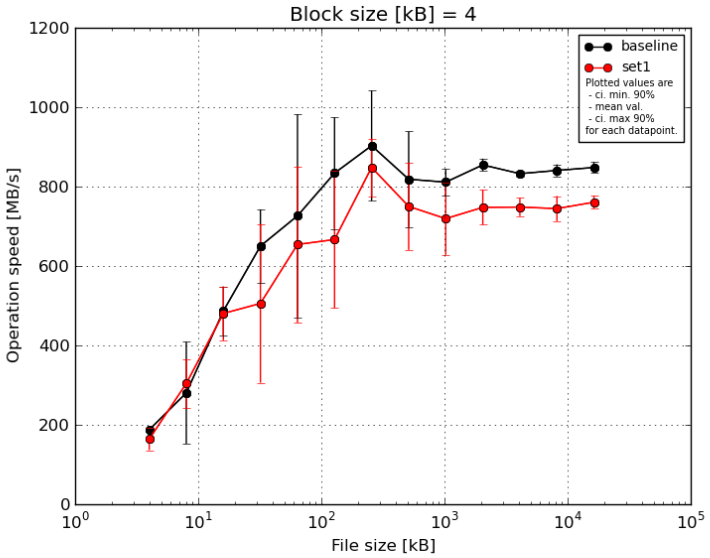
5.2.2 Detailní režim

Detail režim slouží k podrobnějšímu pohledu na výsledky a statistiky jedné operace. Spouští se zadáním parametru `--detail` následovaného jménem operace současně s parametry specifikujícími vstupní soubory, obdobně jako v normálním režimu. Nepoužívá se víceslovný název operace z Iozone dokumentace, ale název bez mezer (v tabulkách v normálním režimu v levém horním rohu, například `iread`, `iwrite` nebo `randrd`).

Výstupem detail režimu jsou dvě stránky, jedna pro fixní velikosti bloku, druhá pro fixní velikosti souboru. Stránka pro fixní velikosti bloku se skládá ze sekcí věnovaným konkrétním hodnotám velikosti bloku. V každé sekci je graf a tabulka. Tabulka obsahuje stejné statistické veličiny jako tabulky v normálním režimu, navíc ale obsahuje přímo i data pro danou velikost bloku. Uvedené statistické veličiny jsou vypočteny pouze z hodnot příslušících dané velikosti bloku. Graf podobně jako v normálním režimu zobrazuje hodnoty z tabulky pod sebou, rozdíl je v zobrazených hodnotách. V normálním režimu jsou vyneseny medián a interval určený prvním a třetím kvartilem. V detail režimu je poloha bodu určena

aritmetickým průměrem a meze zobrazeného intervalu mezemi intervalu 90% spolehlivosti. Obdobně pro stránku s fixní velikostí souboru. Zde hodnoty ve sloupcích tabulky a na vodorovné ose grafu reprezentují hodnoty velikosti bloku, ne souboru. Obrázek 5.4 je ukázkou sekce detailního režimu věnované jedné hodnotě velikosti bloku.

Nástroj v detail režimu nevytváří žádné `.csv` soubory. Všechna vstupní data jsou zobrazena přímo v tabulkách v HTML výstupu, navíc byly `.csv` soubory obsahující všechna data pro všechny operace vytvořeny v normálním režimu. Předpokládá se, že uživatel bude spouštět detail režim pro podrobnější zkoumání výsledků zobrazených v normálním režimu. Smyslem detail režimu je zobrazit podrobnější informace o konkrétní kombinaci velikosti souboru a bloku u jedné operace. To může být užitečné například pro odhalení anomálie ve vstupních datech.



	Block size [kB]	File size [kB]												
		4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
baseline	4	184.49	353.08	556.31	728.96	868.58	926.62	972.88	992.12	807.09	875.2	827.98	859.21	853.57
	4	184.49	50.43	486.15	666.0	846.16	578.17	954.29	725.71	843.79	853.21	846.74	853.33	826.78
	4	195.5	373.19	381.45	482.23	248.05	920.12	969.28	741.88	817.0	831.23	830.89	839.62	864.07
	4	186.59	356.93	504.88	694.22	812.08	839.12	976.5	913.87	755.21	863.58	822.54	820.42	843.62
	4	186.59	269.02	504.88	679.81	857.22	905.81	642.59	719.24	833.23	850.27	834.57	831.07	852.65
	mean val.	187.53	280.53	486.73	650.24	726.42	833.97	903.11	818.56	811.26	854.7	832.55	840.73	848.14
	standard dev.	4.57	134.88	64.39	96.8	268.25	147.16	145.88	126.07	34.4	16.37	9.07	15.87	13.97
	ci. min. 90%	183.17	151.94	425.35	557.95	470.67	693.66	764.03	698.37	778.47	839.09	823.89	825.6	834.82
	ci. max 90%	191.9	409.12	548.12	742.53	982.17	974.27	1042.19	938.76	844.06	870.3	841.2	855.86	861.45
	geom. mean	187.49	229.6	483.04	643.67	661.76	821.74	891.99	811.12	810.67	854.57	832.51	840.61	848.05
	median	186.59	353.08	504.88	679.81	846.16	905.81	969.28	741.88	817.0	853.21	830.89	839.62	852.65
	first quartile	184.49	269.02	486.15	666.0	812.08	839.12	954.29	725.71	807.09	850.27	827.98	831.07	843.62
	third quartile	186.59	356.93	504.88	694.22	857.22	920.12	972.88	913.87	833.23	863.58	834.57	853.33	853.57
	minimum	184.49	50.43	381.45	482.23	248.05	578.17	642.59	719.24	755.21	831.23	822.54	820.42	826.78
	maximum	195.5	373.19	556.31	728.96	868.58	926.62	976.5	992.12	843.79	875.2	846.74	859.21	864.07
set1	4	195.5	338.49	501.02	649.5	743.03	448.18	912.75	755.24	834.06	794.62	773.54	802.17	790.35
	4	139.08	325.06	520.93	440.13	762.48	516.64	883.53	595.98	588.61	693.74	726.48	728.46	756.29
	4	145.24	328.32	504.88	624.73	726.55	855.55	889.52	915.86	787.99	791.77	773.08	721.49	751.7
	4	143.96	190.73	354.62	165.33	285.31	690.94	722.26	713.12	678.93	747.39	744.34	742.87	755.61
	4	195.5	338.49	520.93	649.5	753.71	822.02	833.65	770.5	706.73	711.21	724.88	727.99	749.8
	mean val.	163.86	304.22	480.48	505.83	654.21	666.67	848.34	750.14	719.26	747.75	748.46	744.6	760.75
	standard dev.	28.98	63.73	70.94	209.57	206.66	180.72	76.15	115.15	95.86	45.79	23.93	33.12	16.77
	ci. min. 90%	136.23	243.46	412.84	306.03	457.19	494.37	775.74	640.36	627.87	704.09	725.65	713.02	744.76
	ci. max 90%	191.48	364.98	548.11	705.64	851.24	838.97	920.94	859.93	810.65	791.4	771.28	776.17	776.74
	geom. mean	161.87	297.55	475.63	453.47	615.75	646.01	845.44	743.05	714.04	746.62	748.16	744.03	760.6
	median	145.24	328.32	504.88	624.73	743.03	690.94	883.53	755.24	706.73	747.39	744.34	728.46	755.61
	first quartile	143.96	325.06	501.02	440.13	726.55	516.64	833.65	713.12	678.93	711.21	726.48	727.99	751.7
	third quartile	195.5	338.49	520.93	649.5	753.71	822.02	889.52	770.5	787.99	791.77	773.08	742.87	756.29
	minimum	139.08	190.73	354.62	165.33	285.31	448.18	722.26	595.98	588.61	693.74	724.88	721.49	749.8
	maximum	195.5	338.49	520.93	649.5	762.48	855.55	912.75	915.86	834.06	794.62	773.54	802.17	790.35
baseline set1 difference		-12.63 %	-8.44 %	-1.29 %	-22.21 %	-9.94 %	-20.06 %	-6.06 %	-8.36 %	-11.34 %	-12.51 %	-10.1 %	-11.43 %	-10.3 %
ttest p-value		0.1088	0.7317	0.8875	0.1994	0.6463	0.1471	0.4781	0.3964	0.0781	0.0012	0.0001	0.0004	0.0
ttest equality		SAME	SAME	SAME	SAME	SAME	SAME	SAME	SAME	DIFF	DIFF	DIFF	DIFF	DIFF

Obrázek 5.4: Ukázka detailu pro jednu operaci a jednu velikost zapisovaného bloku

5.2.3 Statistický aparát použitý v nástroji

V případě opakovaného měření jedné náhodné veličiny splňují naměřená data Gaussovo rozložení. V takovém případě je vhodné použít aritmetický průměr a standardní odchylku. Za jednu náhodnou veličinu lze považovat výkonnost souborového systému pro danou operaci na jedné velikosti souboru po blocích o jedné velikosti. Takovéto případy nastávají v detail režimu. V těchto případech lze spolehlivě Studentovým t-testem určit intervaly spolehlivosti. Proto jsou v detail režimu grafy kresleny z hodnot aritmetického průměru a hranic intervalu 90% spolehlivosti.

Při spojení dat z měření vícero náhodných veličin (agregace podle velikosti souboru nebo bloku) už ale nemusí výsledná datová sada splňovat Gaussovo rozložení. K takové situaci dochází v normálním režimu nástroje. Používat v takovém případě aritmetický průměr a interval spolehlivosti může být zavádějící. Agregovaná data je těžké statisticky popsat, nejvhodnější k tomu je využití 1. a 3. kvartilu ke zjištění, v jakém intervalu leží 50 % naměřených hodnot. Ani použití kvartilů ale neumožňuje statisticky přesně rozhodnout, zda má *Set1* vyšší, nižší nebo stejnou výkonnost jako *Baseline*. Iozone provádí pro velké soubory a malé velikosti bloku více měření a z toho důvodu mohou být rozdílným počtem hodnot ovlivněny i kvartily. Proto byl u každé operace v normálním režimu přidán graf lineární regrese, který umožňuje ověřit výsledek poměru mediánů určující procentní odlišnost obou datových sad. U souhrnu v normálním režimu je agregace dat nejvyšší, v grafu jsou zobrazeny i maximální a minimální hodnoty naměřených dat pro srovnání s hodnotami prvního a třetího kvartilu.

Studentův t-test ani výpočty většiny ostatních statistických veličin nebyly v nástroji přímo implementovány, ale byly využity existující implementace z modulů `numpy` a `scipy` jazyka Python. Výsledky byly ověřeny porovnáním s vyhledávačem Wolfram Alpha, který poskytuje rozhraní k nástroji Mathematica od firmy Wolfram Research.

5.3 Porovnání výsledků pomocí vytvořeného nástroje

Nástroj je navržen pro porovnávání dvou sad dat, proto není možné porovnat více souborových systémů zároveň. Porovnal jsem tedy současné souborové systémy podporované Red Hat Enterprise Linuxem každý s každým. Porovnávaná data pochází ze tří různých verzí Enterprise Linuxu 5 a dvou Enterprise Linuxu verze 6. Bylo by zajímavé porovnat výkonnost souborových systémů v různých verzích Red Hat Enterprise Linuxu, bohužel vzhledem k rozsahu této práce a převážně velké časové náročnosti to není možné.

5.3.1 Interpretace srovnání

V následujících podkapitolách budou uvedeny výsledky porovnání naměřených dat vytvořeným nástrojem. Pro každou porovnávanou dvojici souborových systémů budou uvedeny případy, ve kterých byl vyhodnocen jako výkonnější první souborový systém a případy, kdy druhý. Případy, které nejsou uvedeny, nebyly T-testem na hladině významnosti 10 % vyhodnoceny jako odlišné. Odlišnosti ve výkonu obou porovnávaných souborových systémů jsou značeny dvěma procentuálními rozdíly odpovídajícími jiným způsobům výpočtu (tzv. *med* a *reg*). Označení *med* značí hodnotu vypočtenou podílem mediánů obou porovnávaných datových sad pro jednu operaci. Hodnota v procentech určuje o kolik je ve zkoumaném souborovém systému (*Set1*, vždy v porovnání uvedený jako druhý) medián hodnot naměřených pro danou operaci vyšší nebo nižší než medián v referenčním souborovém systému (*Base-*

line, v porovnání první uvedený). Procentní hodnota vypočtená ze sklonu regresní přímky, označená jako *reg* je vypočtena jako $|1 - a|$, kde a je koeficient sklonu regresní přímky. Nejedná se o poměr sklonů regresních přímek aproximujících hodnoty porovnávaných datových sad, jde o sklon jedné regresní přímky aproximující data obou datových sad. Detaily konstrukce grafu lineární regrese a regresní přímky jsou popsány v podkapitole 5.2.1.

V případech, kdy jsou hodnoty procentních rozdílu *med* a *reg* shodné nebo velmi podobné (absolutní hodnota *reg* se neliší o více jak 10 % absolutní hodnoty *med*), je uvedena pouze jediná hodnota procentního rozdílu. Pro porovnání souborových systémů ve výchozí konfiguraci s využitím vyrovnávací paměti je připojen graf závěrečného shrnutí vysvětlující rozdíly mezi výsledky *med* a *reg*. V některých případech se v datových sadách vyskytují maximální hodnoty výrazně vyšší než hodnoty mediánů. Pro velké soubory a malé bloky je naměřeno více hodnot a tyto pak díky svému počtu ovlivňují hodnotu mediánu. Sklon regresní přímky je ovlivněn všemi aproximovanými body stejně. V případech, kdy je jeden porovnávaný souborový systém výkonnější například pro malé soubory a druhý pro velké se pravděpodobně budou výsledky *med* a *reg* lišit.

5.3.2 Ext4/XFS

S využitím vyrovnávací paměti

Graf 5.5 zobrazuje souhrnné výsledky porovnání Ext4 s XFS pro všechny operace.
Souborový systém Ext4 je výkonnější pro:

- opakované čtení souborů (12 % *reg*, 5 % *med*)
- přepis čerstvě zapsaných dat v souborech o velikosti mezi 128 kiB a 512 kiB
- čtení s posuvem, nejvíce v pásmu středních velikostí souboru, zapisovaných po malých blocích (12 %)
- zápis voláním `fwrite` pro soubory mezi 512 kiB a 1 MiB
- čtení voláním `fread` kromě souborů o velikostech 512 MiB a 1 GiB (6 %)

Souborový systém XFS je výkonnější pro:

- zápis souborů větších než 1 MiB po blocích větších než 4 kiB (11 % *reg*, 16 % *med*)
- přepis čerstvě zapsaných dat v souborech o velikosti do 32 kiB a nad 512 MiB (11 % *reg*, 18 % *med*)
- zápis voláním `fwrite` pro soubory do 256 kiB a nad 2 MiB (7 % *med*, 10 % *reg*)
- čtení voláním `fread` pro soubory o velikostech 512 MiB a 1 GiB

Při nastavení příznaku `O_DIRECT`

Při nastavení příznaku `O_DIRECT` jsou výsledky výrazně vyrovnanější než při použití vyrovnávací paměti. Kromě zápisu byly všechny ostatní operace vyhodnoceny jako shodné nebo rozdílné do 3 %.
Souborový systém XFS je výkonnější pro:

- zápis souborů meších než 64 MiB po blocích větších než 16 kiB (11 % *reg*, 62 % *med*)

5.3.3 Ext4/Ext3

S využitím vyrovnávací paměti

Graf 5.6 zobrazuje souhrnné výsledky porovnání Ext4 s Ext3 pro všechny operace.
Souborový systém Ext4 je výkonnější pro:

- přepis existujících dat od velikosti souboru 64 kiB (16 %)
- opakované čtení (46 % *reg*, 5 % *med*)
- náhodné čtení (64 % *reg*, 19 % *med*)
- náhodný zápis souborů od velikosti 64 kiB (28 % *reg*, 22 % *med*)
- přepis čerstvě zapsaných dat v souborech od velikosti 128 kiB (6 %)
- zápis voláním `fwrite` (42 %)
- přepis voláním `fwrite` pro soubory od velikosti 64 kiB (13 %)
- opakované čtení voláním `fread` v souborech o velikostech 64 kiB a 128 kiB

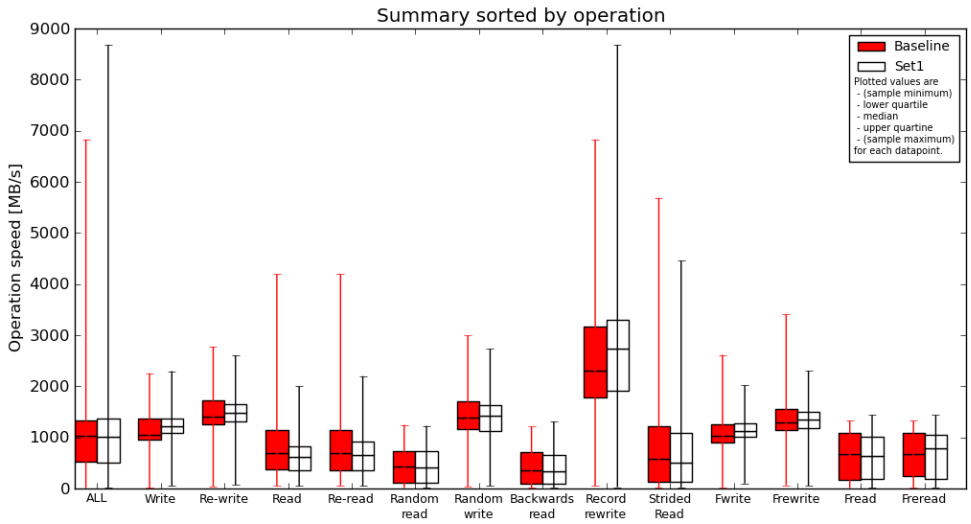
Souborový systém Ext3 je výkonnější pro:

- zápis (47 %)
- přepis existujících dat do velikosti souboru 32 kiB
- čtení (39 % *reg*, 5 % *med*)
- náhodný zápis souborů do velikosti 32 kiB
- zpětné čtení na souborech do velikosti 64 MiB (59 % *reg*, 37 % *med*)
- přepis čerstvě zapsaných dat v souborech do velikosti 64 kiB
- čtení s posuvem pozice pro soubory do velikosti 64 MiB a od velikosti 2 GiB (22 % *reg*, 32 % *med*)
- přepis voláním `fwrite` pro soubory do velikosti 32 kiB
- čtení voláním `fread` v souborech menších než 512 MiB (40 % *reg*, 20 % *med*)
- opakované čtení voláním `fread` v souborech do velikosti 64 MiB (47 % *reg*, 21 % *med*)

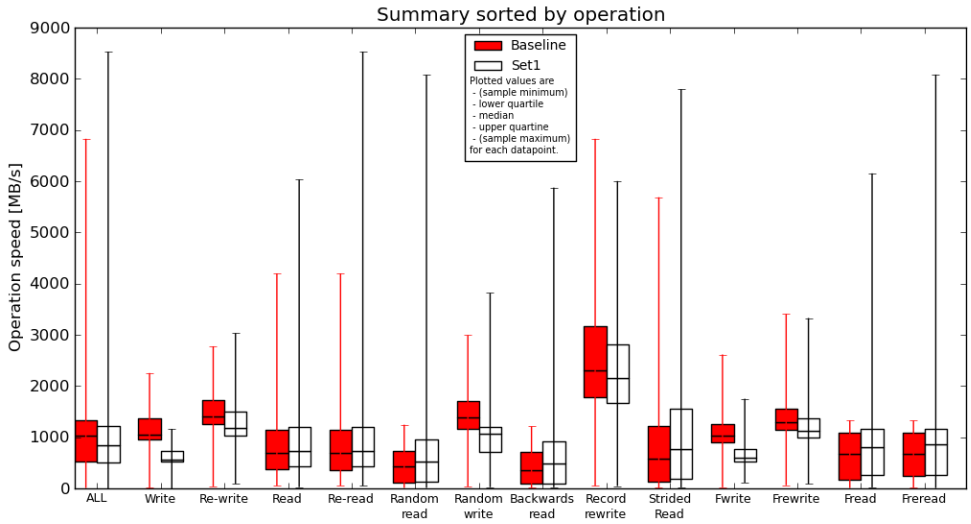
Při nastavení příznaku `O_DIRECT`

Zde jsou výsledky opět vyrovnanější, díky tomu se i ve většině případů shodují výsledky lineární regrese s podílem mediánů.
Souborový systém Ext4 je výkonnější pro:

- zápis souborů větších než 64 MiB
- přepis, nejvýrazněji na blocích od 8 MiB (18 % *med*, 9 % *reg*)
- náhodný zápis, nejvýrazněji na blocích od 8 MiB (20 % *med*, 6 % *reg*)



Obrázek 5.5: Souhrnné výsledky porovnání Ext4 (*Baseline*) s XFS (*Set1*)



Obrázek 5.6: Souhrnné výsledky porovnání Ext4 (*Baseline*) s Ext3 (*Set1*)

- přepis právě zapsaných dat od velikosti souboru 256 kiB (22 % *med*, 12 % *reg*)

Souborový systém Ext3 je výkonnější pro:

- zápis souborů menších než 16 MiB (7 % *reg*, 67 % *med*)

5.3.4 Ext4/Ext2

S využitím vyrovnávací paměti

Graf 5.7 zobrazuje souhrnné výsledky porovnání Ext4 s Ext2 pro všechny operace.
Souborový systém Ext4 je výkonnější pro:

- čtení a opakované čtení od velikosti souboru 512 kiB (7 %)
- náhodný zápis souborů mezi 64 kiB a 512 kiB a souborů větších než 256 MiB (6 %)
- přepis voláním `fwrite` pro soubory o velikosti mezi 128 kiB a 512 kiB a soubory větší než 256 MiB

Souborový systém Ext2 je výkonnější pro:

- zápis, s výjimkou souborů o velikosti 4 GiB (15 % *reg*, 25 % *med*)
- přepis právě zapsaných dat ve všech případech kromě souborů o velikostech 256 kiB a 512 kiB (25 %)
- zápis voláním `fwrite` (21 %)

Při nastavení příznaku `O_DIRECT`

Zde opět nenajdeme výraznější rozdíly.
Souborový systém Ext4 je výkonnější pro:

- zápis souborů od velikosti 128 MiB

Souborový systém Ext2 je výkonnější pro:

- zápis souborů do velikosti 64 MiB (16 % *reg*, 89 % *med*)
- všechny zbývající operace vyjma čtení s posuvem pozice (5 % *reg*, 7 % *med*)

5.3.5 Ext3/XFS

S využitím vyrovnávací paměti

Graf 5.8 zobrazuje souhrnné výsledky porovnání Ext4 s XFS pro všechny operace. Maxima výsledků Ext3 u všech operací vycházejících ze čtení dosahují hodnot zhruba desetkrát vyšších, než jsou odpovídající hodnoty mediánů. To má za následek výrazně vyšší čísla ve výsledcích lineární regrese.
Souborový systém Ext3 je výkonnější pro:

- čtení souborů do velikosti 128 MiB s výjimkou souborů o velikosti 64 kiB (15 % *med*, 38 % *reg*)

- opakované čtení s výjimkou souborů o velikosti 64 kiB (10 % *med*, 40 % *reg*)
- náhodné čtení (21 % *med*, 58 % *reg*)
- zpětné čtení (32 % *med*, 59 % *reg*)
- přepis právě zapsaných dat v souboru od velikosti 256 kiB (27 % *med*, 21 % *reg*)
- čtení s posuvem pozice (34 % *med*, 28 % *reg*)
- čtení voláním `fread` pro soubory mezi 256 kiB a 256 MiB (22 % *med*, 67 %)
- opakované čtení voláním `fread` (8 % *med*, 68 % *reg*)

Souborový systém XFS je výkonnější pro:

- zápis (119% *med*, 108 % *reg*)
- opakovaný zápis souborů větších než 32 kiB (25 % *med*, 21 % *reg*)
- náhodný zápis souborů od velikosti 64 kiB (32 % *med*, 36 % *reg*)
- zápis voláním `fwrite` (89 % *med*, 81 % *reg*)
- přepis souborů větších než 32 kiB voláním `fwrite` (20 % *med*, 15 % *reg*)

Při nastavení příznaku `O_DIRECT`

Souborový systém XFS je výkonnější pro:

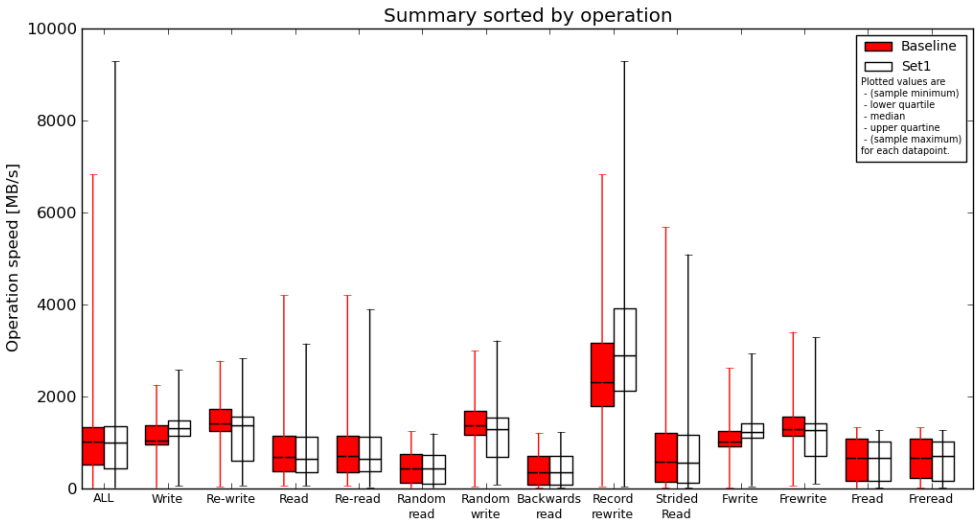
- přepis souborů od velikosti 16 MiB, výrazněji po blocích od velikosti 8 MiB (20 % *med*, 6 % *reg*)
- náhodný zápis, nejvíce pro bloky od velikosti 8 MiB (21 % *med*, 3 % *reg*)
- přepis čerstvě zapsaných dat od velikosti souboru 16 kiB (26 % *med*, 10 % *reg*)

5.3.6 Ext3/Ext2

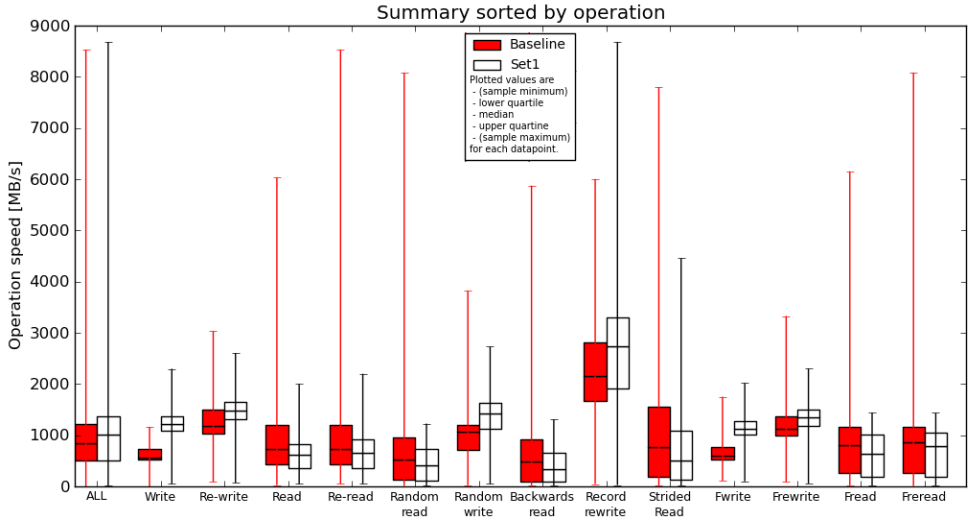
S využitím vyrovnávací paměti

Zde jako *Baseline* vystupují tatáž data Ext3, proto také pozorujeme maximální hodnoty výrazně vyšší oproti mediánům. Graf 5.9 zobrazuje souhrnné výsledky porovnání Ext3 s Ext2 pro všechny operace.
Souborový systém Ext3 je výkonnější pro:

- čtení (12 % *med*, 31 % *reg*)
- opakované čtení (12 % *med*, 35 % *reg*)
- náhodné čtení (17 % *med*, 41 % *reg*)
- zpětné čtení, výrazněji pro velké bloky (40 % *med*, 39 % *reg*)
- čtení s posuvem pozice (28 % *med*, 21 % *reg*)



Obrázek 5.7: Souhrnné výsledky porovnání Ext4 (*Baseline*) s Ext2 (*Set1*)



Obrázek 5.8: Souhrnné výsledky porovnání Ext3 (*Baseline*) s XFS (*Set1*)

- přepis souborů od velikosti 256 MiB voláním `fwrite`
- čtení voláním `fread` (18 % *med*, 31 % *reg*)
- opakované čtení voláním `fread` (16 % *med*, 33 % *reg*)

Souborový systém Ext2 je výkonnější pro:

- zápis (137 % *med*, 117 % *reg*)
- přepis, nejvíce pro soubory do velikosti 512 MiB (17 % *med*, -3 % *reg*)
- náhodné zápis (21 % *med*, 12 % *reg*)
- přepis právě zapsané hodnoty (34 % *med*, 35 % *reg*)
- zápis voláním `fwrite` (109 % *med*, 100 % *reg*)
- přepis souborů do velikosti 64 MiB voláním `fwrite` (14 % *med*, 31 % *reg*)

Při nastavení příznaku `O_DIRECT`

Zde se liší pouze operace zápisu, přepisu a přepisu právě zapsaných dat.
Souborový systém Ext2 je výkonnější pro:

- zápis, výrazněji od velikosti bloku 8 kiB (13 % *med*, 8 % *reg*)
- přepis, nejvíce od velikosti bloku 8 kiB (16 % *med*, 4 % *reg*)
- přepis čerstvě zapsaných dat (28 % *med*, 10 % *reg*)

5.3.7 Ext2/XFS

S využitím vyrovnávací paměti

Graf 5.10 zobrazuje souhrnné výsledky porovnání Ext2 s XFS pro všechny operace.
Souborový systém Ext2 je výkonnější pro:

- zápis do velikosti souboru 16 MiB, po blocích do velikosti 256 kiB (7 %)
- přepis souborů do velikosti 256 MiB po blocích do 32 kiB (28 % *reg*, 7 % *med*)
- náhodný zápis souborů menších než 64 kiB po blocích do 32 kiB
- přepis právě zapsaných dat (10 % *reg*, 5 % *med*)
- čtení s posuvem pozice po malých blocích (8 %)
- zápis voláním `fwrite` pro soubory do velikosti 128 MiB (9 % *med*, 10 % *reg*)
- přepis dat voláním `fwrite` pro soubory do velikosti 32 kiB po blocích menších než 64 kiB

Souborový systém XFS je výkonnější pro:

- čtení (11 % *reg*, 3,5 % *med*)

- opakované čtení (8 % *reg*, 2 % *med*)
- náhodný zápis souborů větších než 512 MiB po blocích od 32 kiB (25 % *reg*, 9 % *med*)
- přepis dat voláním `fwrite` pro soubory od velikosti 512 MiB po blocích větších než 64 kiB (5 % *med*, 4% *reg*)

Při nastavení příznaku `O_DIRECT`

Souborový systém Ext2 je výkonnější pro:

- zápis souborů do velikosti 2 MiB (14 % *med*, 4 % *reg*)

Souborový systém XFS je výkonnější pro:

- opakované čtení, nejvíce na souborech od 512 MiB (7 % *med*, 5 % *reg*)

5.3.8 Btrfs

Souborový systém Btrfs není v současné době běžně používaným souborovým systémem a tedy by neměl být obsahem této práce. Je však velmi diskutovaným a pravděpodobně se v budoucnosti stane běžným linuxovým souborovým systémem. Proto jsem se jej rozhodl porovnat se stávajícími souborovými systémy. Btrfs se nachází ve stadiu vývoje a jeho výkonnostní vlastnosti se budou pravděpodobně ještě měnit. Testy na souborovém systému Btrfs probíhaly pod Red Hat Enterprise Linuxem verze 6, konkrétně RHEL6.0-20100922.1. Btrfs je srovnáván s výstupy Iozone získanými na Red Hat Enterprise Linuxu 6.0, které byly také použity pro výše uvedené porovnání dnes běžných souborových systémů.

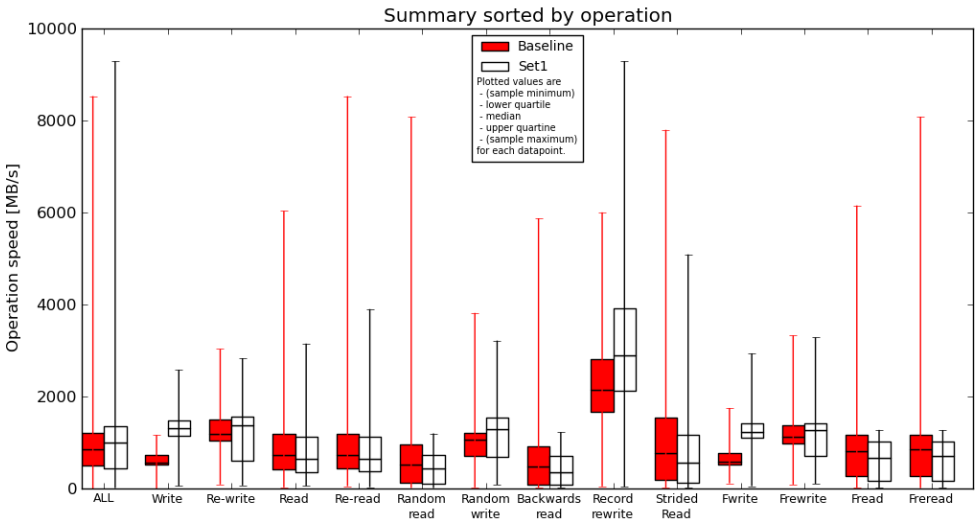
Btrfs/XFS

Btrfs je u operací vycházejících ze čtení výrazně pomalejší než XFS. Maximální hodnoty jsou však srovnatelné. Proto zde také nastává velký nepoměr mezi maximálními hodnotami a mediány, který způsobuje rozdíly mezi výsledky *reg* a *med*.
Souborový systém Btrfs je při využití vyrovnávací paměti výkonnější pro:

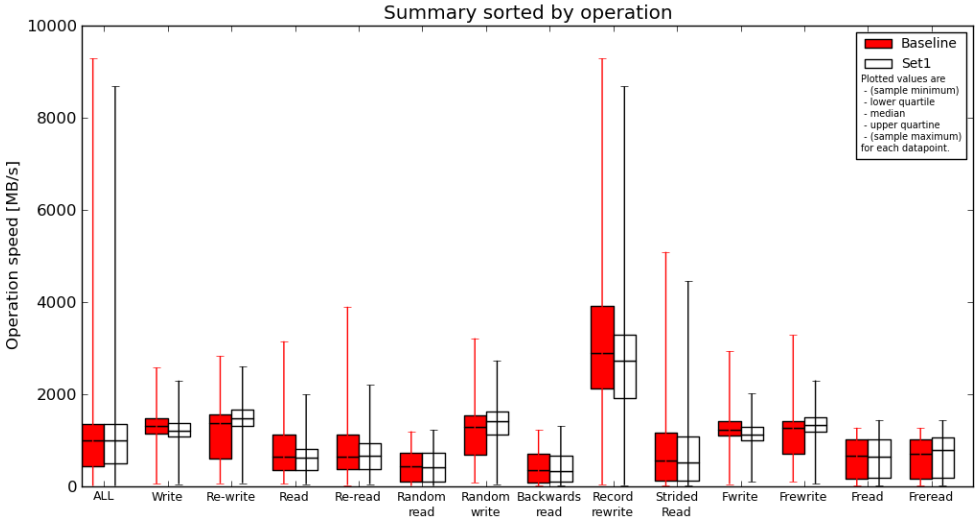
- zápis, s výjimkou souborů o velikosti 4 GiB (30 %)
- přepis souborů od velikosti 2 MiB po blocích od 128 kiB
- náhodný zápis souborů od 32 MiB do 2 GiB po blocích od 1 MiB
- čtení s posuvem pozice od velikosti souboru 512 MiB
- zápis voláním `fwrite` s výjimkami pro soubor o velikosti 4 GiB a 4 kiB blok (27 %)
- přepis souborů o velikosti nad 1 MiB voláním `fwrite` po blocích nad 64 kiB

Souborový systém XFS je při využití vyrovnávací paměti výkonnější pro:

- přepis souborů do velikosti 1 MiB po blocích do 64 kiB
- čtení (196 % *reg*, 515 % *med*)
- opakované čtení (182 % *reg*, 503 % *med*)



Obrázek 5.9: Souhrnné výsledky porovnání Ext3 (*Baseline*) s Ext2 (*Set1*)



Obrázek 5.10: Souhrnné výsledky porovnání Ext2 (*Baseline*) s XFS (*Set1*)

- náhodné čtení souborů do velikosti 512 MiB (280 % *reg*, 254 % *med*)
- náhodný zápis souborů do velikosti 1 MiB po blocích do 128 kiB
- čtení pozpátku, nárůst rychlosti je výraznější pro větší bloky (210 % *med*, 328 % *reg*)
- přepis právě zapsané hodnoty, s výjimkou souboru o velikosti 1 GiB, rozdíl je vyšší u menších velikostí bloku (29 % *reg*, 48 % *med*)
- čtení s posuvem pozice do velikosti souboru 256 MiB (402 % *med*, 192 % *reg*)
- přepis souborů o velikosti do 1 MiB voláním `fwrite` po blocích do velikosti 64 kiB
- čtení voláním `fread` (500 % *med*, 233 % *reg*)
- opakované čtení voláním `fread` (478 % *med*, 236 % *reg*)

Souborový systém XFS je při nastavení příznaku `O_DIRECT` paměti výkonnější pro:

- všechny operace vycházející ze čtení (okolo 430 % *reg*, 300 % *med*)
- zápis (238 % *med*, 458 % *reg*)
- náhodný zápis (348 % *med*, 503 % *reg*)
- přepis (392 % *med*, 522 % *reg*)
- přepis právě zapsaných dat (620 % *reg*, 574 % *med*)

Btrfs/Ext4

U výsledků testů s využitím vyrovnávací paměti srovnáváme stejné křivky příslušící Btrfs s Ext4 místo XFS. Na Btrfs lze pozorovat výrazné propady rychlosti všech operací vycházejících ze zápisu pro soubory o velikosti 4 GiB. V předchozí podkapitole bylo uvedeno, že rozdíly mezi XFS a Btrfs dosahují často hodnot stovek procent ve prospěch XFS. Rozdíly mezi souborovými systémy z rodiny Ext a XFS nedosahují zdaleka takových hodnot. Viz podsekce 5.3.2. Z tohoto důvodu nepovažuji za nutné detailně popisovat výsledky Btrfs ve srovnání s každým z těchto souborových systémů, protože většina výsledků budou rozdíly v řádech stovek procent v neprospěch Btrfs. Zmíním tedy pouze případy kdy je Btrfs výkonnější.
Souborový systém Btrfs je při využití vyrovnávací paměti výkonnější pro:

- zápis souborů do velikosti 2 GiB (39 % *med*, 34 % *reg*)
- přepis souborů od velikosti 2 MiB po blocích od 64 kiB (13 % *med*, 17 % *reg*)
- náhodný zápis souborů od velikosti 2 MiB po blocích od 64 kiB (14 % *med*, 8 % *reg*)
- zápis souborů voláním `fwrite` do velikosti 2 GiB (32 %)
- přepis souborů od velikosti 2 MiB po blocích od 128 kiB (11 % *med*, 6 % *reg*)

Při nastavení příznaku `O_DIRECT` jsou výsledky porovnání téměř shodné s porovnáním Btrfs s XFS. Ext4 je rychlejší o stovky procent než Btrfs ve všech operacích. Největší rozdíl je u přepisu právě zapsaných dat, dále u operací vycházejících ze čtení a nejmenší, přesto stále velmi vysoký rozdíl najdeme u zápisu.

Btrfs/Ext3 a Ext2

Zde se z důvodů uvedených v předchozí podkapitole opět zaměřím na oblasti ve kterých je Btrfs výkonnější.
Souborový systém Btrfs je při využití vyrovnávací paměti oproti Ext3 výkonnější pro:

- zápis (69 %)
- přepis souborů od velikosti 256 kiB po blocích od 32 kiB (28 % *med*, 32 % *reg*)
- náhodný zápis souborů od velikosti 512 kiB po blocích od 32 kiB (35 % *med*, 45 % *reg*)
- zápis voláním `fwrite` (26 % *med*, 35 % *reg*)
- přepis souborů od velikosti 512 kiB voláním `fwrite` (26 % *med*, 35 % *reg*)

Souborový systém Btrfs je při využití vyrovnávací paměti oproti Ext2 výkonnější pro:

- zápis (33 %)
- přepis souborů od velikosti 32 MiB po blocích od 64 kiB (16 % *med*, 35 % *reg*)
- náhodný zápis souborů od velikosti 32 MiB po blocích od 64 kiB (17 % *med*, 27 % *reg*)
- zápis voláním `fwrite` (26 % *med*, 35 % *reg*)
- přepis souborů od velikosti 32 MiB voláním `fwrite` (13 % *med*, 25 % *reg*)

U Ext3 je při nastavení příznaku `O_DIRECT` výstup nástroje pro zpracování výstupů Iozone opět téměř shodný s výsledkem porovnání Btrfs s XFS a Ext4. U Ext2 jsou také výsledky podobné, jediným rozdílem je, že operace zápisu není v Btrfs ve srovnání s Ext2 pomalejší méně než ostatní operace.

5.3.9 Souborové systémy se změněnými parametry

XFS

Při připojování souborového systému XFS je možné upravit parametrem `allocsize` velikost předalokovávaného místa. Podle [7] je výchozí hodnotou je 64 kiB. Změnil jsem hodnotu parametru `allocsize` na 512 kiB ve víře, že dojde ke zvýšení výkonnosti pro větší soubory. Dle mé úvahy mělo být daní za toto zrychlení zvýšení fragmentace. Výsledky porovnání s XFS připojeným s výchozími parametry ale ukázaly zpomalení okolo 20 % u souborů o velikosti 64 kiB u přepisu, čtení, náhodného zápisu, přepis voláním `fwrite` a čtení voláním `fread`. Dále byl podobně zpomalen i přepis souborů o velikosti 128 kiB a náhodný zápis souborů o velikosti 16 kiB. Jediným zaznamenaným zrychlením při zvětšení předalokovávaného místa byl 36 % přírůstek rychlosti opakovaného čtení voláním `fread` pro soubory o velikosti 4 kiB. Pro 8 kiB soubor však bylo zaznamenáno zpomalení o 13 %.

Při nastavení příznaku `O_DIRECT` nebyl naměřen téměř žádný rozdíl, na souborech o velikosti 512 MiB je čtení pomalejší o 8 %, opakované čtení o 9 % a čtení s posuvem o 19 %. Přepis právě zapsaných dat je o 13 % pomalejší pro bloky o velikosti 4 kiB.

Ext4

Na diskovém oddílu o velikosti 20 GiB se při použití výchozího nastavení vytvoří souborový systém Ext4 s 1 313 280 i-uzly, na jeden i-uzel tedy připadá 16 352 bytů. Při vytváření je možné zadat počet bytů na i-uzel, zvolil jsem dvojnásobek výchozí hodnoty v očekávání vyšší výkonnosti u velkých souborů vlivem snížení režie související s metadaty i-uzlů. Dále jsem očekával zvýšení fragmentace a snížení výkonnosti u malých souborů.
Souborový systém Ext4 s polovičním počtem i-uzlů je při využití vyrovnávací paměti výkonnější pro:

- přepis voláním `fwrite` pro soubory o velikosti 128 MiB o 8 % *med*
- náhodné čtení souborů o velikosti 4 kiB o 13 % *med*

Souborový systém Ext4 ve výchozí konfiguraci je při využití vyrovnávací paměti výkonnější pro:

- přepis voláním `fwrite` pro soubory mezi 4 kiB a 64 kiB a pro soubory o velikostech 128 kiB a 256 kiB (1,5 % *reg*, 1 % *med*)
- náhodný zápis 16 kiB souborů a 128 kiB souborů
- čtení souborů o velikosti 8 kiB
- zpětné čtení souborů o velikosti 8 kiB
- čtení souborů o velikosti 64 kiB s posuvem pozice
- opakované čtení voláním `fread` na souborech o velikosti 64 kiB
- čtení voláním `fread` na souborech o velikostech 64 kiB a 256 kiB
- opakovaný zápis souborů o velikostech 128 kiB a 256 MiB
- přepis právě zapsaných dat v souborech o 64 kiB, 128 kiB a 512 kiB

Souborový systém Ext4 ve výchozí konfiguraci je s nastaveným příznakem `O_DIRECT` výkonnější pro:

- přepis souborů o velikosti 1 GiB
- čtení souborů o velikosti 512 MiB
- přepis právě zapsané hodnoty od velikosti bloku 4 MiB
- čtení souborů o velikosti 512 MiB s posuvem pozice

Ext3

Souborový systém Ext3 ve výchozím nastavení vytvoří na 20 GiB oddílu 2 626 560 i-uzlů, na jeden tedy připadá 8 176 bytů. Testován byl Ext3 s dvojnásobnou hodnotou, tedy opět polovičním počtem i-uzlů.
Souborový systém Ext3 s polovičním počtem i-uzlů je při využití vyrovnávací paměti výkonnější pro:

- zápis souborů mezi 16 kiB a 256 kiB o 3 % *med*
- opakovaný zápis souborů o velikostech 128 kiB o 58 % *med*, 256 MiB o 14 % *med* a 4 GiB o 7 % *med*
- opakované čtení 8 kiB souborů o 14 % *med*
- náhodný zápis souborů mezi 16 kiB a 128 kiB o 5 % až 12 % *med*
- přepis čerstvě zapsaných dat v 128 kiB souboru o 58 % *med*
- přepis voláním `fwrite` na souborech o velikostech 16 kiB a 32 kiB o 19 % *med* a pro soubory mezi 32 MiB a 256 MiB o 4 % *med*

Souborový systém Ext3 ve výchozí konfiguraci je při využití vyrovnávací paměti výkonnější pro:

- náhodné čtení 8 kiB souborů
- přepis čerstvě zapsaných dat v 512 kiB souboru

Jediným nezanedbatelným rozdílem zjištěným při použití příznaku `O_DIRECT` bylo 6 % zpomalení přepisu souborů o velikosti 512 MiB.

Ext2

Pro Ext2 platí stejné hodnoty počtu i-uzlů jako pro ext3, opět byl testován souborový systém s polovičním počtem i-uzlů proti výchozímu nastavení.
Souborový systém Ext2 s polovičním počtem i-uzlů je při využití vyrovnávací paměti výkonnější pro:

- zápis souborů o velikosti od 32 kiB do 128 kiB a 256 MiB (1.23 % *med*, 5 % *reg*)
- přepis souborů o velikosti od 16 kiB do 128 kiB a od 512 MiB do 1 GiB, nejvíce po blocích do velikosti 16 kiB (3,72 % *med*, 6 % *reg*)
- čtení souborů o velikosti 256 kiB o 12 % *med*
- opakované čtení pro soubory o velikostech 8 kiB, 16 kiB, 64 kiB, 512 MiB a pro soubory od 1 MiB do 4 MiB (4,6 % *med*, 6 % *reg*)
- náhodný zápis souborů o velikosti mezi 8 kiB a 256 kiB po blocích do velikosti 32 kiB (3,56 %. *med*, 5,5 % *reg*)
- přepis právě zapsané hodnoty pro soubory mezi 8 kiB a 512 kiB po blocích do 32 kiB (5 %)
- čtení s posuvem pro soubory mezi 8 kiB a 64 kiB o 8 % až 36 % *med*
- zápis souborů mezi 32 kiB a 128 kiB voláním `fwrite` o 14 % až 48 % *med*
- přepis voláním `fwrite` pro soubory do velikosti 128 kiB s výjimkou souborů o velikosti 16 kiB po blocích do velikosti 32 kiB (4,61 % *med*, 8,5 % *reg*)
- čtení voláním `fread` pro soubory mezi 16 kiB a 256 kiB s výjimkou souborů o velikosti 64 kiB o 13 % až 21 % *med* a pro soubory mezi 2 MiB a 16 MiB.

- opakované čtení voláním `fread` pro soubory o velikostech 16 kiB a 32 kiB o 27 % *med* a pro soubory o velikostech 256 kiB a 1 MiB až 16 MiB o zhruba 4 % *med*

Souborový systém Ext2 ve výchozí konfiguraci je při využití vyrovnávací paměti výkonnější pro:

- přepis souborů o velikosti 2 GiB
- přepis právě zapsané hodnoty pro soubory o velikosti 4 kiB o 35 % *med*

5.4 Postmark

Studie [17] doporučuje užití makrobenchmarku a následované užitím mikrobenchmarku pro zjištění přesných detailů. Zvolil jsem opačný postup z toho důvodu, že nehledám nejvhodnější souborový systém pro aplikaci produkující konkrétní zátěž, ale snažím se poskytnout přehled výkonností souborových systémů pro různé druhy zátěže. Výstup programu Postmark neposkytuje tolik detailních informací jako výstup Iozone, proto jej není možné použít pro detailnější analýzu.

Podobně jako u Iozone byl využit již existující Beaker test. Ten slouží ke spouštění programu Postmark v prostředí Beaker. Data byla měřena programem Postmark v následující konfiguraci. Testované souborové systémy byly vytvářeny s výchozími parametry na diskovém oddílu o velikosti 20 GiB. Spodní limit velikosti souborů byl 8 kiB, horní 16 kiB. Operace probíhaly paralelně na 131 072 souborech, s každým byly provedeny 4 transakce. Poměr čtení a připisování byl nastaven na 10:1 pro čtení a poměr vytváření a mazání byl nastaven na 5:1 ve prospěch vytváření.

Výsledky programu Postmark se poměrně liší od výsledků Iozone. Výsledky Iozone se pohybují v řádu stovek MiB/s, oproti tomu se výsledky Postmark pro operace čtení a zápisu pohybují v řádek jednotek MiB/s. Za hlavní příčinu tohoto rozdílu považuji fakt, že test Postmark probíhal paralelně na velkém množství souborů, z toho důvodu dosahuje výkon zlomku hodnot naměřených Iozone. Aby naměřená data získala statistickou významnost byla data na každém souborovém systému v rámci jedné úlohy v systému Beaker měřena pětkrát. Statistiky byly vypočteny nad daty z vícero úloh, dále uváděná čísla jsou mediány.

Operace čtení byla nejrychlejší v souborovém systému Ext4, kde byla naměřena rychlost 18,5 MiB/s. Následuje Ext3 s 14,8 MiB/s, XFS s 8,82 MiB/s a nejpomalejší výsledek 5,89 MiB/s byl naměřen v Ext2. Shodné umístění souborových systémů je i u zápisu, nejvyšší hodnota 13,87 MiB/s byla naměřena u Ext4, následují Ext3 s 11,1 MiB/s, XFS s 6,62 MiB/s a Ext2 s 4,42 MiB/s.

Dalším zajímavým výstupem programu Postmark, jehož ekvivalent ale mezi výsledky Iozone nenajdeme, je rychlost vytváření a mazání souborů. Na první pozici je opět Ext4 s hodnotou 1 184,08 souboru za sekundu, následují Ext3 s 947,26, XFS s 564,82 a Ext2 s 377,09 souboru za sekundu. U mazání souborů je náskok Ext4 ještě výraznější, Ext4 dosáhl hodnoty 9 347,57 smazaných souborů za sekundu, následovaný Ext3 s výsledkem 4 846,89, XFS s 4 221,28 a Ext2 s výsledkem 1 270,54 souborů za sekundu.

Kapitola 6

Závěr

Při nastavení příznaku `O_DIRECT` byly naměřeny podstatně menší rozdíly mezi souborovými systémy než při použití vyrovnávací paměti. Hlavní příčinu odlišné výkonnosti souborových systémů vidím v jejich rozdílné práci s vyrovnávací pamětí.

Souborový systém Ext2 vyniká v zápisu, jak voláními `write` tak i `fwrite`. Největší náskok je u malých souborů a bloků. Ext2 je také nejvýkonnější v operaci přepisu čerstvě zapsaných dat, u náhodného zápisu se ale jeho náskok ztrácí. Ext2 byl ale hodnocen nejhůře při provádění paralelních operací v programu Postmark.

Souborový systém Ext3 vyšel jako nejrychlejší v operacích zpětného čtení, náhodného čtení a čtení s posuvem pozice. Dále také vyniká v sekvenčním a opakovaném čtení voláním `fread` a o něco méně ve čtení voláním `read`. Naopak jeho slabinou jsou operace zápisu.

Ext4 je ve čtení oproti Ext3 celkově mírně pomalejší, je však nejrychlejší ve čtení větších souborů. Ext4 snižuje ztrátu Ext3 v zápisu, v sekvenčním zápisu ale nedosahuje výkonu Ext2 a XFS, opakovaný, sekvenční i opakovaný zápis voláním `fwrite` již nejsou v porovnání se zbývajícími souborovými systémy v Ext4 pomalejší tak výrazně. Ext4 dobře zvládá velké množství paralelně prováděných operací s malými soubory, ve výsledcích programu Postmark byl nejvýkonnějším souborovým systémem ve všech kategoriích.

Souborový systém XFS se podle naměřených dat nejvíce osvědčil při operacích vycházejících ze zápisu na velkých souborech po velkých blocích dat. XFS je také nejvýkonnější v náhodném zápisu. Dobře zvládá i paralelní zpracování velkého množství souborů mezi osmi a šestnácti kiB, ve výsledcích Postmark stojí na druhém místě.

Souborový systém Btrfs byl záměrně z předchozího srovnání vynechán, protože jej nepovažuji za v současném stavu nasaditelný v produkčním prostředí. Jedním z důvodů jsou naměřené velmi nízké rychlosti operací vycházejících ze čtení, kde Btrfs velice výrazně zaostává za ostatními testovanými souborovými systémy. Silnou stránkou Btrfs je naopak zápis, kde překonává i Ext2.

Starší souborové systémy (Ext2 a Ext3) vychází podle naměřených dat výkonnější pro menší soubory po menších blocích než novější souborové systémy (Ext4 a Btrfs) a XFS a naopak. Rozdíly ve výkonnosti různých souborových systémů v závislosti na velikostech souboru a bloku nejsou tak výrazné jako rozdíly ve výkonnosti různých souborových systémů v závislosti na prováděné operaci. Změna jednoho z parametrů souborového systému kromě nejstaršího Ext2 přináší nepříliš výrazné zhoršení výkonnosti. Pouze na Ext2 došlo v některých případech k významnějšímu zrychlení a v jiných ke zpomalení. Z uvedených skutečností vyplývá, že koexistence různých souborových systémů má své opodstatnění, každý souborový systém přináší jisté přednosti a volba souborového systému závisí na typu aplikace.

Přínosem autora této práce je kromě samotného textu práce, který může sloužit administrátorům operačního systému GNU/Linux jako podklad pro volbu souborového systému vhodného pro konkrétní aplikaci i podíl na nástroji pro analýzu výstupů Iozone. V rámci úprav nástroje autorem této práce se zdrojový kód nástroje rozrostl z 419 na 1 570 řádků. V době psaní této práce (květen 2011) probíhá nasazení vytvořeného nástroje ve firmě Red Hat, kde bude sloužit k odhalování výkonnostních regresí souborových systémů.

Literatura

[1] IOzone Filesystem Benchmark [online]. Říjen 2006 [cit. 2010-12-24].
Dostupný z WWW: http://www.iozone.org/

[2] Open Source XFS for Linux : Providing the World's Most Scalable Journaling-Filesystem Technology to the Open Source Community [online]. Červenec 2006 [cit. 2010-12-27].
Dostupný z WWW: http://oss.sgi.com/projects/xfs/datasheet.pdf

[3] XFS: A high-performance journaling filesystem [online]. 2009-2010 [cit. 2010-12-27].
Dostupný z WWW: http://oss.sgi.com/projects/xfs/index.html

[4] Solaris Internals : Domovská stránka programu FileBench [online]. 2010-05-06 [cit. 2010-12-28].
Dostupný z WWW:
http://www.solarisinternals.com/wiki/index.php/FileBench

[5] Btrfs : Domovská stránka souborového systému Btrfs [online]. Prosinec 2010 [cit. 2010-12-29].
Dostupný z WWW: https://btrfs.wiki.kernel.org/index.php/Main_Page

[6] Btrfs : Návrh souborového systému Btrfs [online]. Prosinec 2010 [cit. 2010-12-29].
Dostupný z WWW: https://btrfs.wiki.kernel.org/index.php/Btrfs_design

[7] The SGI XFS Filesystem [online]. Prosinec 2010 [cit. 2011-04-23].
Dostupný z WWW: http://www.kernel.org/doc/filesystems/xfs.txt

[8] Aurora, V.: A short history of btrfs [online]. Červenec 2009 [cit. 2010-12-29].
Dostupný z WWW: https://lwn.net/Articles/342892/

[9] Card, R.; T'so, T.; Tweedie, S.: Design and Implementation of the Second Extended Filesystem. In *Proceedings of the First Dutch International Symposium on Linux*, 1995.

[10] Corbet, J.: Barriers and journaling filesystems [online]. Květen 2008 [cit. 2011-05-03].
Dostupný z WWW: http://lwn.net/Articles/283161/

[11] Johnson, M. K.: Whitepaper: Red Hat's New Journaling File System: ext3 [online]. 2001 [cit. 2010-12-26].
Dostupný z WWW: http://www.redhat.com/support/wpapers/redhat/ext3/

[12] Jones, M. T.: Anatomy of Linux journaling file systems : Journaling today and tomorrow [online]. Červenec 2008 [cit. 2010-12-26].

Dostupný z WWW: http://www.ibm.com/developerworks/library/l-journaling-filesystems/index.html

[13] Katcher, J.: PostMark: A New File System Benchmark [online]. Srpen 1997 [cit. 2010-12-28].
Dostupný z WWW: http://communities.netapp.com/servlet/JiveServlet/download/2609-1551/Katcher97-postmark-netapp-tr3022.pdf

[14] Mathur, A.; Cao, M.; Bhattacharya, S.; aj.: The new ext4 filesystem: current status and future plans. In *Proceedings of the Linux Symposium*, Červen 2007.

[15] Robbins, D.: Common threads: Advanced filesystem implementor's guide, Part 7 : Introducing ext3 [online]. Listopad 2001 [cit. 2010-12-26].
Dostupný z WWW:
http://www.ibm.com/developerworks/linux/library/l-fs7.html

[16] Robbins, D.: Common threads: Advanced filesystem implementor's guide, Part 9 : Introducing XFS [online]. Leden 2002 [cit. 2010-12-27].
Dostupný z WWW:
http://www.ibm.com/developerworks/linux/library/l-fs9.html

[17] Traeger, A.; Joukov, N.; Wright, C. P.; aj.: A Nine Year Study of File System and Storage Benchmarking. *ACM Transactions on Storage (TOS)*, ročník 4, č. 2, Květen 2008: s. 25–80.

[18] Trochim, W. M.: The T-Test [online]. Říjen 2006 [cit. 2011-04-12].
Dostupný z WWW: http://www.socialresearchmethods.net/kb/stat_t.php

[19] Weisstein, E. W.: Box-and-Whisker Plot [online]. Duben 2011 [cit. 2011-04-12].
Dostupný z WWW: http://mathworld.wolfram.com/Box-and-WhiskerPlot.html

Dodatek A

Obsah CD

Obsahem přiloženého CD jsou adresáře s následujícím obsahem.

- **beaker-xml** – konfigurační soubory pro Beaker, které byly použity k získání dat využitých v této práci
- **iozone-output** – výstupy Iozone Beaker testu použité jako vstupní data nástroje pro zpracování výstupů Iozone
- **postmark-output** – výsledky Postmark Beaker testu diskutované v sekci 5.4
- **results** – výstupy nástroje pro zpracování výstupů Iozone diskutované v sekci 5.3
- **src** – git repozitář dokumentující všechny změny provedené na nástroji pro zpracování výstupů Iozone autorem této práce
- **tex** – soubory se zdrojovým kódem pro LaTeX obsahující text této práce