JFS – MODERNÍ SOUBOROVÉ SYSTÉMY

Jan Koreň, 8.11.2011

Náplň přednášky

- Historie, vývoj
- Vlastnosti
- Žurnálování
- Struktura JFS
- Provoz, testy

Historie

- Journaling File System
- Vytvořen IBM
- 1990 – JFS1 pro AIX
- 1999 – JFS2 pro OS/2
- 2001 – JFS2 portován do OS Linux (dále jen JFS)
- Součástí linuxového jádra od verze jádra 2.4.2

Stav vývoje

- http://jfs.sourceforge.net/
- Konference LinuxWorld (leden 2003):
  - V dohledné době:
    - Defragmentace FS ☹
    - Sdílení externího logu více než jedním FS ☹
  - Ve vzdálenější budoucnosti:
    - Kvóty ☹
    - Data Management API (DMAPI) ☺

Dostupnost

- Obsažen ve většině linuxových distribucí
  - Debian GNU/Linux 3.0 >
  - Gentoo 1.4 >
  - Mandrake 8.1 >
  - Red Hat 7.3 >
  - Slackware 8.1 >
  - SuSE 7.3 >
  - Ubuntu 4.10 >
  - …

Kdo používá JFS

- Backblaze
  - http://www.backblaze.com/
  - Online záloha dat
- Technogym
  - http://www.technogym.com/
- Shark Linux
  - http://www.sharkos.org/
  - JFS jako výchozí FS
- Affymetrics
  - http://www.affymetrix.com/
  - Technologie pro genovou analýzu

Vlastnosti

- 64bitový FS
- Max. počet souborů: bez limitu
- Max. délka souboru: 255 znaků
- Min. velikost oddílu: 16 MB
- Max. velikost oddílu
  - Záleží na velikosti diskového bloku
  - 4PB (pro 512 B/blok) - 32 PB (4096 B/blok)
- Max. velikost souboru
  - 512 TB (pro 512 B/blok) - 4 PB (4096 B/blok)

Vlastnosti (2)

- Alokace místa pro diskové inody
  - Dynamická
  - Uvolnění místa, když už inode není třeba
  - Odpadá nutnost zadávat pevný počet inodů při tvorbě
  - Není možné, že nebude stačit počet inodů kvůli velkému počtu malých souborů

Vlastnosti (3)

- Extenty
  - Sekvence sousedních bloků agregátu alokované souborům jako celek
  - Redukce fragmentace
  - 24bit. hodnota pro délku extentu – 1 – $(2^{24} – 1)$ bloků
  - Pokud je velikost bloku 4 KB → max. velikost extentu je ~64GB
  - Vede k efektivnímu využití prostoru i v případě, že vel. souborů narůstá
  - Indexovány v B+ stromu

Vlastnosti (4)

- Hojné využití B+ stromů
  - Souborový layout
    - Inode obsahuje kořen B+ stromu, který popisuje extenty obsahující uživatelská data
  - Záznamy o adresářích
  - Čtení a zápis extentů
  - Přidání extentu souboru
  - Indexování extentů
  - Průchod celým B+ stromem pro odstranění souboru

Vlastnosti (5)

- **Organizace adresářů**
  - B+ strom
    - setříděný podle názvu
  - Obsah malých adresářů je uložen přímo v jejich inodech (v kořenu B+ stromu)
    - Až 8 záznamů v inodu vyjma aktuálního adresáře (.) a jeho nadřazeného adresáře(..)

Vlastnosti (6)

- ACL, rozšířené atributy (EA)
  - Je součástí většiny distribucí
  - attr, getfattr, setfattr, getfacl, setfacl
  - Pokud není, na obojí jsou patche – od JFS release 1.0.21
  - Nutno patchovat jádro pro úpravu virtuálního souborového systému, aby podporoval obojí
  - http://acl.bestbits.at – patch pro změny ve VFS
  - Patche pro EA a ACL jsou na stránkách IBM

Vlastnosti (7)

- Snapshoty
  - Pomocí LVM nebo EVMS
- Navýšení velikosti fs
  - Podpora navýšení velikosti FS za běhu
  - FS musí být namontován
  - Použití volby –o remount
  - Př: mount –o remount, resize /mount_point

Vlastnosti (8)

- Husté a řídké soubory
  - Podpora obou
  - Řídké soubory – umožňují zapsat data na náhodné pozice v souboru, aniž by bylo nutné zapisovat do mezilehlých souborových bloků
    - JFS pak udává velikost souboru podle nejvyššího použitého bloku, avšak alokoval pouze aktuálně použité bloky
  - Husté soubory – jsou alokovány všechny takové bloky, aby se zaplnila velikost souboru
    - A je jedno, jestli jsou do nich zapsána data nebo ne

Vlastnosti (9)

- Concurrent Input/Output (CIO)
  - Běžně read-shared, write-exclusive zamykání souborů
  - Volba CIO – vypnutí zamykání → snížení režie systému
  - Vhodné např. pro databáze, které si samy udržují konzistenci dat

Vlastnosti (10)

- JFS nepodporuje:
  - Kvóty
  - Deduplikaci
  - Zmenšení (shrink) FS
  - Nepřítomen nástroj pro defragmentaci (avšak v IBM OS ano)

Žurnálování

- Hlavním cílem JFS je rychlá obnova kozistence velkých file systémů po havárii – několik sekund
- Vyhne se tak fsck, které může trvat několikrát déle
- Pouze logování metadat = správa konzistence struktury filesystemu
  - Změny superbloku, inodů, adresářů a alokačních struktur
- Log (žurnál) může být umístěn na externím zařízení, pokud je toto zařízení uvedeno při vytváření FS
  - Argument –j
  - Př.: mkfs.jfs –j /dev/external_log /dev/xxxx

Žurnálování (2)

- Čas obnovy nezávisí na velikosti oddílu
- Po pádu systému se mohou objevit zastaralá data, ale soubory by měly zůstat konzistentní a použitelné
- Log zabírá předem danou oblast na disku
- Po havárii – přehrání logu (=uskutečnění nahraných transakcí) vede k obnově konzistence filesystému

Žurnálování (3)

- Pokud systém po přehrání žurnálu není *clean*:
  - log byl z nějakého důvodu přehrán špatně nebo neúplně
  - Je nutná úplná kontrola konzistence
- Konzistence na prvním místě, zachování dat je až druhořadé
- Nekonzistentní soubor/adresář je vyhozen
- Soubory/podadresáře, které osiřely po smazání poškozeného adresáře, jsou umístěny do adresáře lost+found (v /)

Co všechno se žurnáluje

- Vytvoření souboru (create)
- Linking (link)
- Vytvoření adresáře (mkdir)
- Odstranění souboru (unlink)
- Přejmenování (rename)
- Odstranění adresáře (rmdir)
- Symbolický odkaz (symlink)
- Zkrácení souboru

Logovací systém

- Podoperace všech transakcí, které mění metadata, jsou zaznamenány do logovacího souboru předtím, než jsou zapsány na disku
- Částečně hotové transakce mohou být vráceny zpět po restartu systému
- Logovací systém = logovací soubor + transaction manager

Logovací systém (2)

- Logovací soubor
  - Log transakcí má výchozí velikost 0,4% velikosti agregátu (zaokrouhleno na MB směrem nahoru)
  - Maximálně však 128 MB
- Transaction Manager
  - Funkcionalita logování

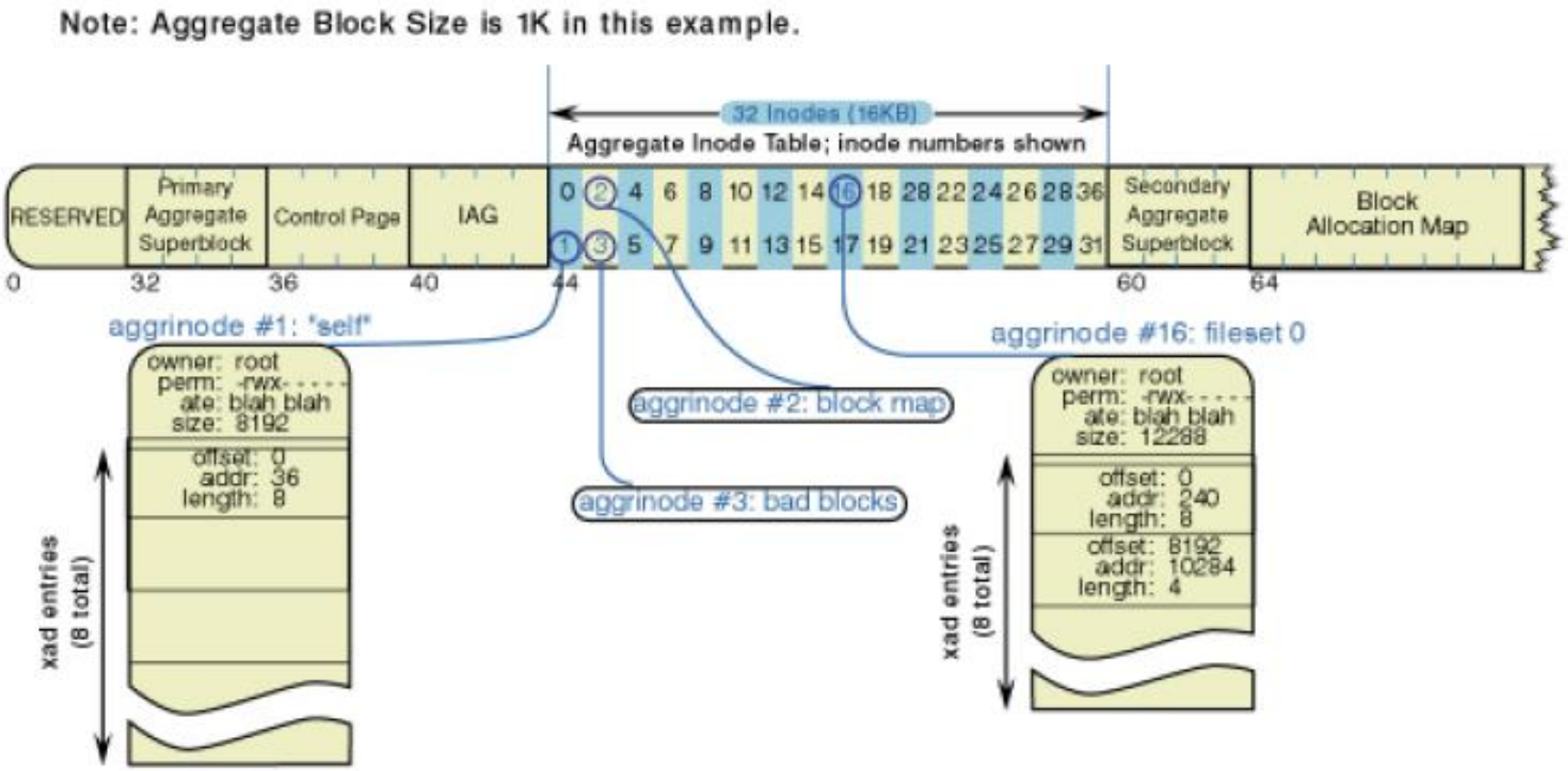
Logredo

- utilita JFS, která přehrává logovací soubor při startu FS
- Přehrání (čtení) logu pozpátku – od konce do nalezení prvního synchronizačního bodu
- Zpracování záznamů v logu
  - Je-li třeba, znovu se provedou uskutečněné transakce
  - nalezení neprovedených transakcí a roll-back všech jejich podoperací, které jsou zaznamenány v logu

Struktura JFS

- Diskový oddíl
  - Velikost
  - Velikost bloku (=IO granularita, 512/1024/2048/4096 B)
  - Obsahuje agregát
- Agregát
  - Souvislá oblast pevného disku
  - Právě 1 pro diskový oddíl
  - Velikost agregát. bloku (též logický blok)
    - Granularita alokace místa souborům
    - V současnosti pouze 4096 B
  - Teoreticky může obsahovat více filesetů, prakticky pouze jeden
- Fileset
  - Kolekce souborů a adresářů, které jsou ve stromové struktuře

Agregát


Note: Aggregate Block Size is 1K in this example.
32 Inodes (16KB)
Aggregate Inode Table; inode numbers shown
RESERVED
Primary Aggregate Superblock
Control Page
IAG
0 2 4 6 8 10 12 14 16 18 28 22 24 26 28 36
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Secondary Aggregate Superblock
Block Allocation Map
0
32
36
40
44
60
64
aggrinode #1: "self"
owner: root
perm: -rwx- - - - -
ate: blah blah
size: 8192
offset: 0
addr: 36
length: 8
xad entries (8 total)
aggrinode #2: block map
aggrinode #3: bad blocks
aggrinode #16: fileset 0
owner: root
perm: -rwx- - - - -
ate: blah blah
size: 12288
offset: 0
addr: 240
length: 8
offset: 8192
addr: 10284
length: 4
xad entries (8 total)

Superblok

- Souhrnné informace o agregátu
  - Velikost agregátu
  - Velikost alokačních skupin
  - Velikost agregátového bloku
  - Kolik obsahuje bloků
  - Stav systému (clean, dirty, …)
  - …
- Záložní superblok je přesná kopie primárního
  - Použit, pokud je primární poškozený
- Oba superbloky jsou na předem známých místech

Tabulka inodů agregátu

- Pole inodů
- Obsahuje inody popisující kontrolní struktury
- #0 je rezervováno
- #1 je samotná tabulka inodů agregátu
- #2 je mapa alokace bloků
- #3 logovací soubor uvnitř agregátu
- #4 vadné bloky nalezené při formátování agregátu
- #5 - #15 jsou rezervované pro budoucí rozšíření
- #16 kořenový soubor filesetu
- Sekundární tabulka inodů agregátu – kopie TIA
  - Samotná data inodů se neopakují, jen struktury potřebné k jejich nalezení

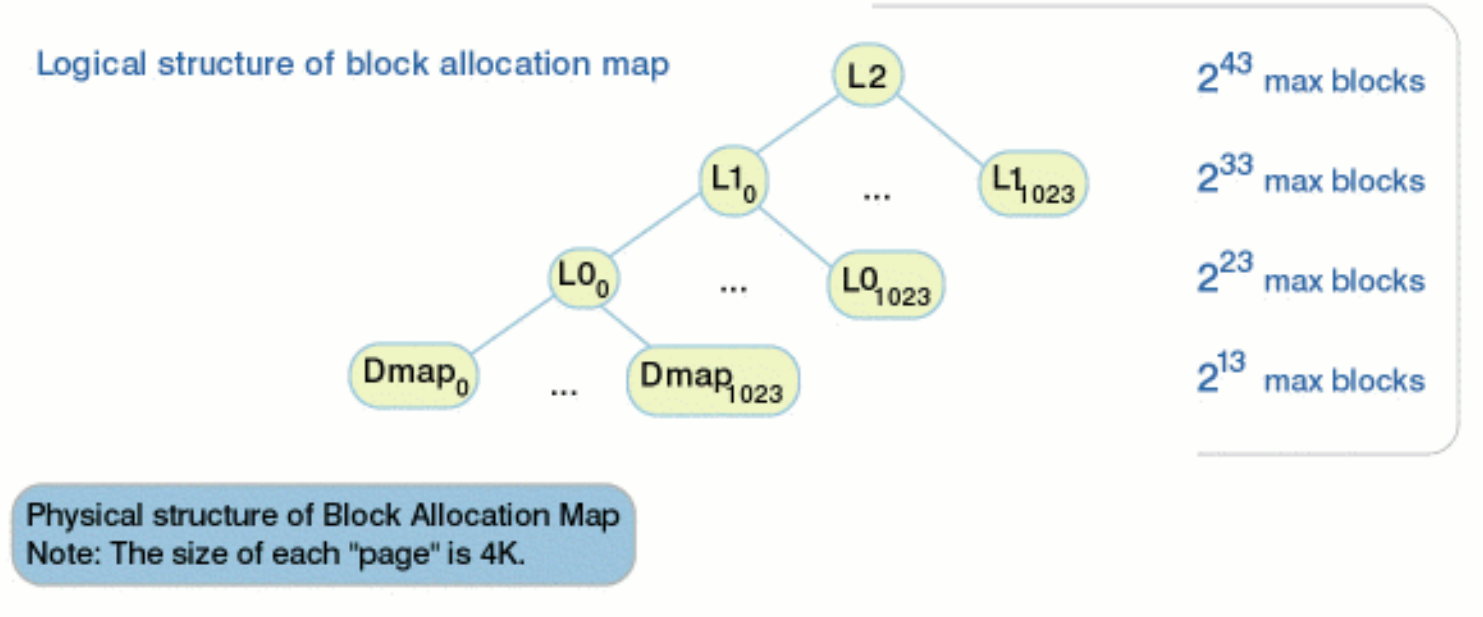
Mapa inodů agregátu

- Dynamické pole skupin alokace inodů (IAGs)
- Popisuje tabulku inodů agregátu
- Informace o stavu alokace
- Umístění na disku
- Sekundární MIA – popisuje sekundární TIA

Mapa alokace bloků

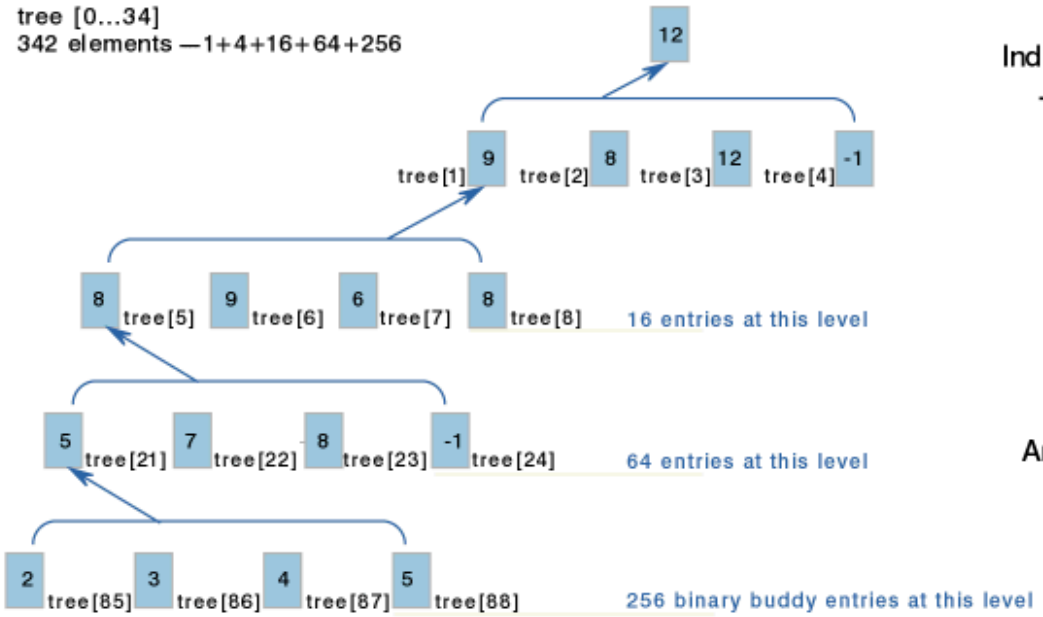
- Informace o alokovaných či volných blocích celého agregátu
- Soubor
- Struktura *dmap* popisující bloky – bitová mapa
  - 1 KB = 8192 bloků
- Stromová struktura mapy
  - Nejvýše 3 úrovně stromu, *dmap* je vždy v listu
  - Povoleno nejvýše 2^{43} bloků – odtud omezení velikosti FS (4KB bloky → 32 PB)

Mapa alokace bloků (2)



Mapa alokace bloků (3)

- Samotná struktura dmap je vnitřně reprezentována jako strom
- V každé úrovni stromu je zaznamenán max. počet sousedních volných bloků z nižší úrovně (4 potomci), až na poslední úroveň
- V té jsou obsaženy hodnoty nejdelšího řetězce volných bitů (jako mocnina 2) pro každé slovo (32b) bitmapy získané pomocí buddy systému



Binary Buddy performed on each word of bitmap to get the bottom level of the tree

FIGURE 6. Tree structure in dmap

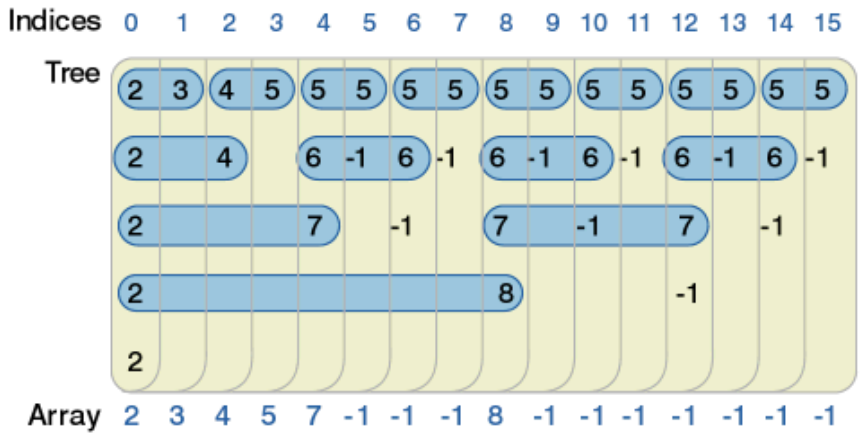
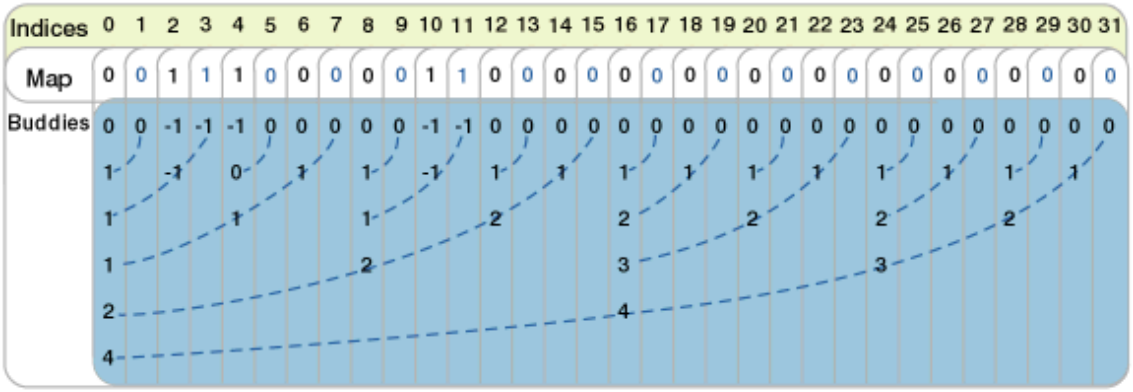


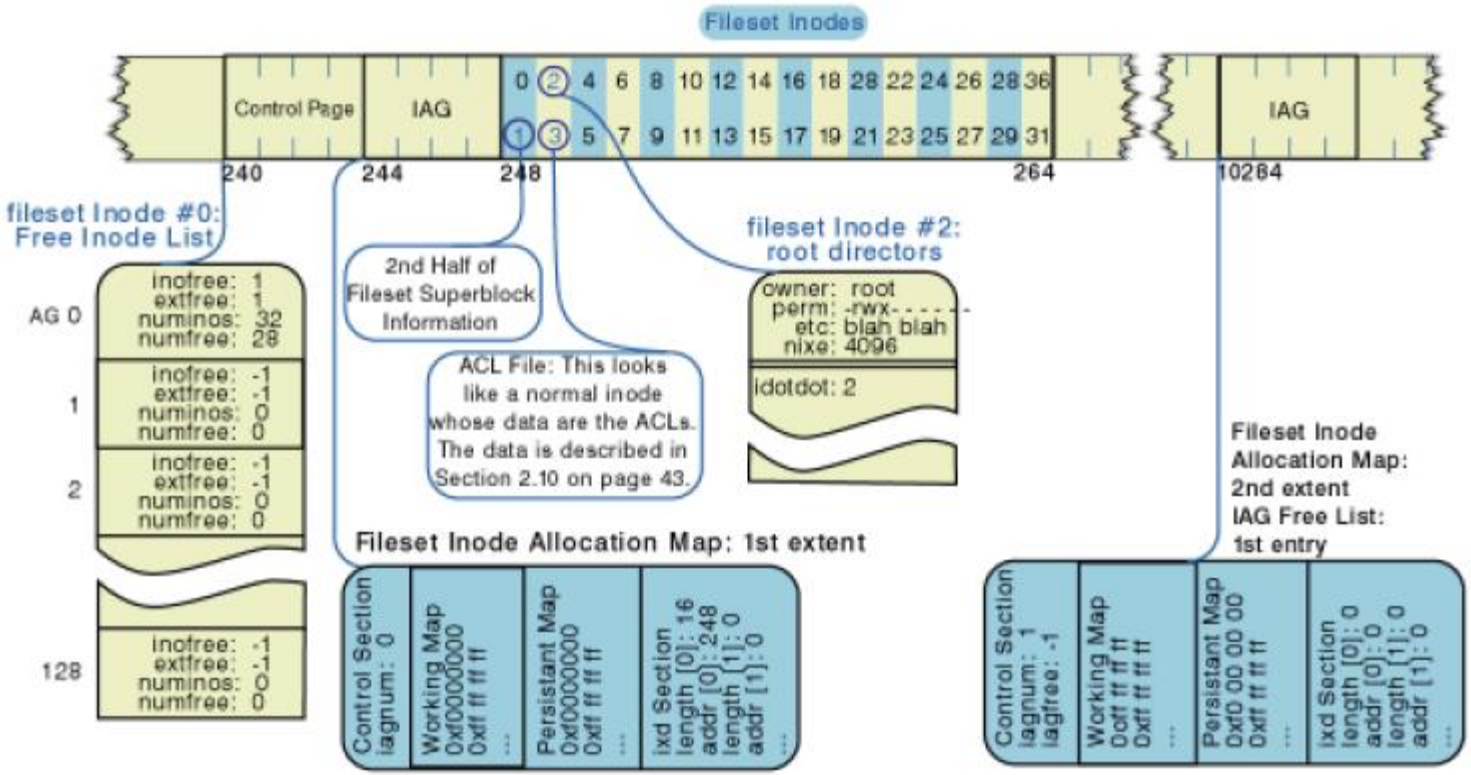
FIGURE 8. Binary Buddy

- Volná slova jsou sloučena s volnými buddy
- pak je buddy napravo označen -1, aby jej už nešlo použít

FIGURE 7. Binary Buddy of a word of the bitmap



Fileset


Fileset Inodes
Control Page
IAG
0 2 4 6 8 10 12 14 16 18 28 22 24 26 28 36
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
IAG
240
244
248
264
10264
fileset Inode #0:
Free Inode List
AG 0
1
2
128
inofree: 1
extfree: 1
numinos: 32
numfree: 28
inofree: -1
extfree: -1
numinos: 0
numfree: 0
inofree: -1
extfree: -1
numinos: 0
numfree: 0
inofree: -1
extfree: -1
numinos: 0
numfree: 0
2nd Half of
Fileset Superblock
Information
ACL File: This looks
like a normal inode
whose data are the ACLs.
The data is described in
Section 2.10 on page 43.
fileset Inode #2:
root directors
owner: root
perm: -rwx- - - - - -
etc: blah blah
nixe: 4096
idotdot: 2
Fileset Inode Allocation Map: 1st extent
Control Section
iagnum: 0
Working Map
0xf0000000
0xff ff ff ff
...
Persistant Map
0xf0000000
0xff ff ff ff
...
ixd Section
length [0]: 16
addr [0]: 248
length [1]: 0
addr [1]: 0
...
Fileset Inode
Allocation Map:
2nd extent
IAG Free List:
1st entry
Control Section
iagnum: 1
iagfree: -1
Working Map
0xff ff ff ff
0xff ff ff ff
...
Persistant Map
0x0 00 00 00
0xff ff ff ff
...
ixd Section
length [0]: 0
addr [0]: 0
length [1]: 0
addr [1]: 0
...

Tabulka inodů filesetu

- Obsahuje inody popisující kontrolní struktury
- 0 – rezervován
- 1 – dodatečné informace o filesetu, „super-inode“
- 2 – inode kořenového adresáře filesetu
- 3 – ACL data filesetu

Mapa alokace inodů filesetu

- Popisuje tabulku inodů filesetu
- Informace o stavu alokace inodů filesetu
  + jejich umístění na disku

Další součásti agregátu

- Pracovní prostor pro *fsck*
  - Množství virtuální paměti pro *fsck* se odvíjí od počtu souborů a adresářů (32 B/1 soubor n. adresář)
  - Rezervovaná oblast používaná namísto virtuální paměti
- In-line log
  - Místo pro žurnálování změn metadat
  - Vždy umístěn za prac. prostorem pro *fsck*

Alokační skupiny (AG)

- K rozdělení agregátu na části
- Snaha držet související data blízko u sebe (=diskové bloky a inody souborů ve stejné AG)
- Snaha dát nesouvisející data do jiných AG, aby mohla být později související data ve stejné AG)
- Až 128 AG v agregátu
- Každá AG musí mít aspoň 8192 bloků nebo být velká 32 MB

Mount

- Nejdřív je třeba vytvořit diskový oddíl pomocí fdisk (např. /dev/hdb3)
- **mkfs.jfs /dev/hdb3** – vytvoří soubor. systém JFS s logem uvnitř diskového oddílu
  - Výchozí velikost bloku je 4096 B
- Pro mount použít argument jfs
- **mount –t jfs /dev/hdb3 /jfs**
- odmontování – **umount /jfs**

Mount (2)

- externí log – zlepšuje výkon, protože změny logu jsou ukládány na jiný diskový oddíl
- potřeba mít dva nevyužité diskové oddíly
- **mkfs.jfs –j /dev/hdb1 /dev/hda6**
- **mount –t jfs /dev/hda6 /jfs**
- možno nezapisovat do žurnálu
  - argument **nointegrity**
  - pro vyšší výkon, např. když se obnovuje svazek ze záložního média

jfsutils

- Sada utilit pro práci s JFS
- jfs_debugfs
  - Pro provádění nízkoúrovňových operací na JFS
  - Náhrada dat umístěných na uvedené pozici bloku danou hodnotou
  - Zobrazení záznamů adresářů v daném inodu
  - Zobrazení superbloku, superbloku žurnálu, …

jfsutils (2)

- jfs_fsck
  - Zkontroluje a opraví souborový systém
  - Přehraje log
  - Mělo by být použito pro nenamontovaný nebo pro READ ONLY fs (jinak hrozí poškození fs)
  - Spuštěn automaticky při bootování
- jfs_fscklog
  - Extrahuje obsah logu z dané jednotky a zapíše jej do souboru
  - Popř. jej pouze zobrazí

jfsutils (3)

- jfs_logdump
- jfs_mkfs
- jfs_tune
  - Konfigurace FS
  - Změna lokace žurnálu:
    - mkfs.jfs –J journal_dev /dev/hda2
    - jfs_tune –J device=/dev/hda2 /dev/hdb1

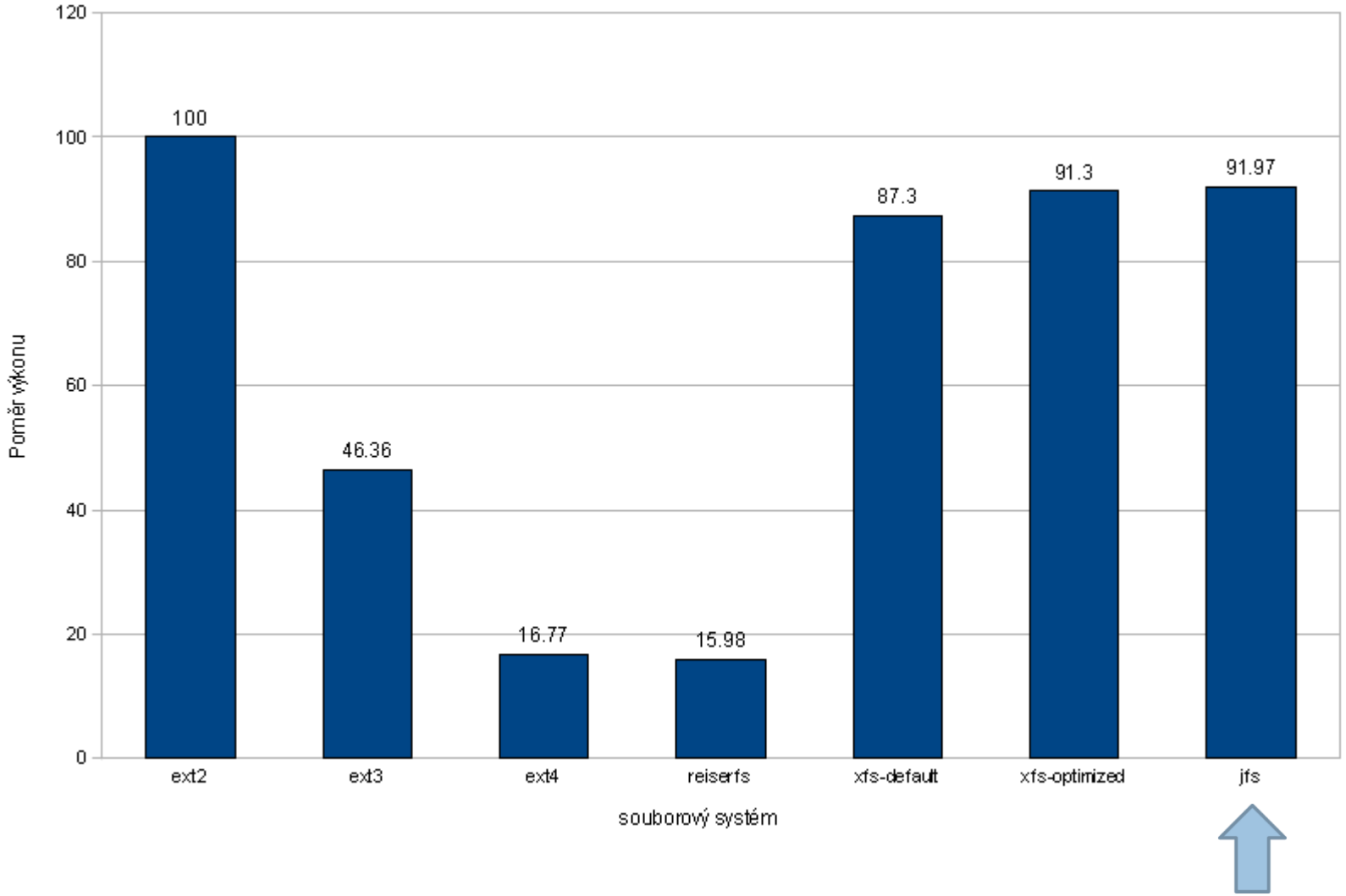
Výkon

- Ze všech FS nejméně zatěžuje procesor
- Ověřeno, že se dosáhne lepších výsledků, pokud jádro Linuxu používá deadline plánovač
- Rychlé vytvoření FS a montování/odmontování
- Rychlý při práci s velkými soubory
- Relativně pomalý, pokud pracuje najednou s mnoha soubory
- Překvapivě rychlý na SSD discích

Benchmarky

- Měření výkonu PostgreSQL
- http://www.iprint.sk/mereni-vykonu-postgresl-na-ruznych-systemech-souboru/
- HW: Intel(R) Core(TM)2 CPU 6320; 8GB DDR2-800; HDD SAMSUNG HD103UJ
- SW: CentOS 5.5 64b (2.6.18-194.11.1.el5.centos.plus); postgresql-8.4.4
- IO plánovač: deadline

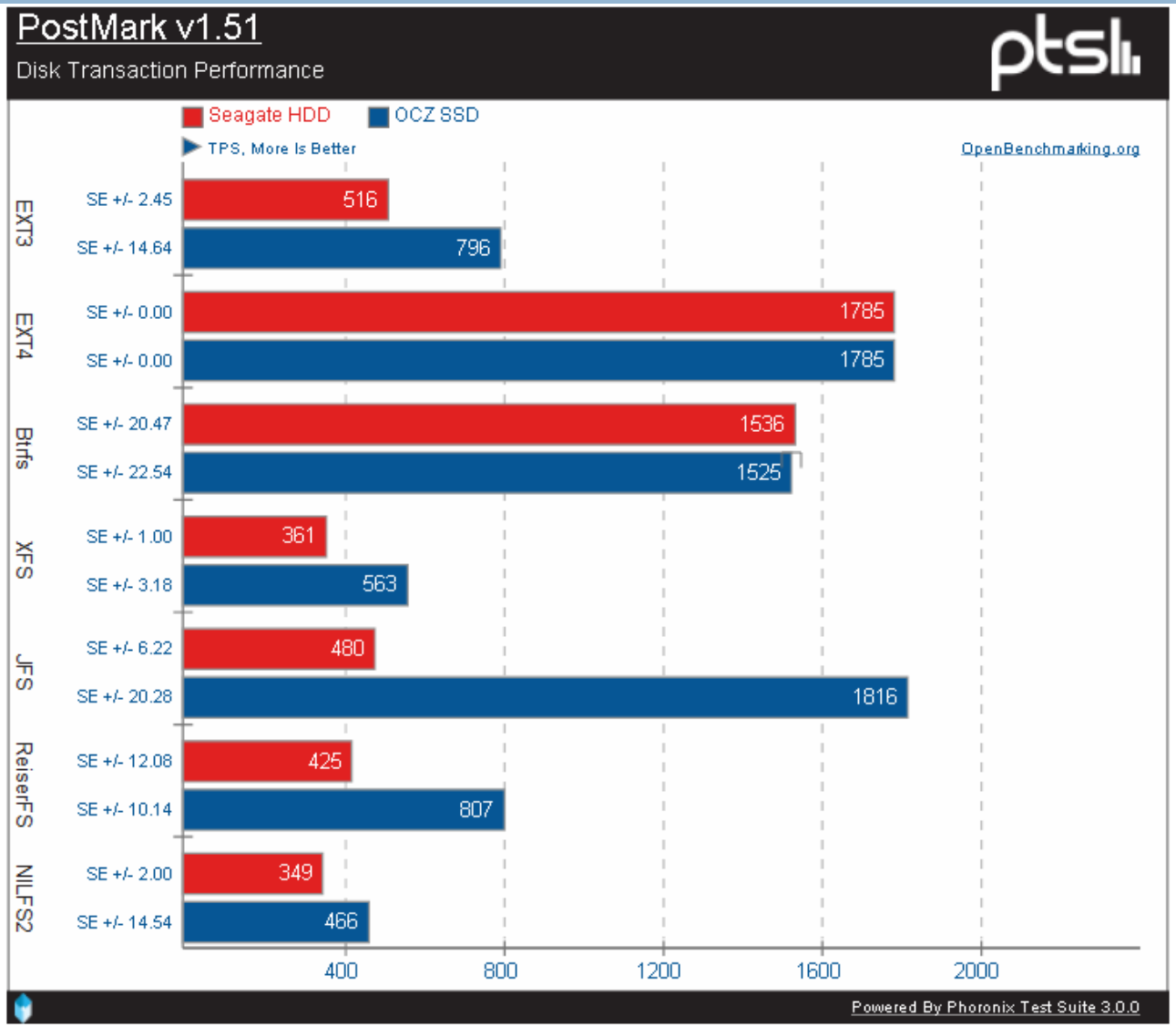
Měření výkonu PostgreSQL



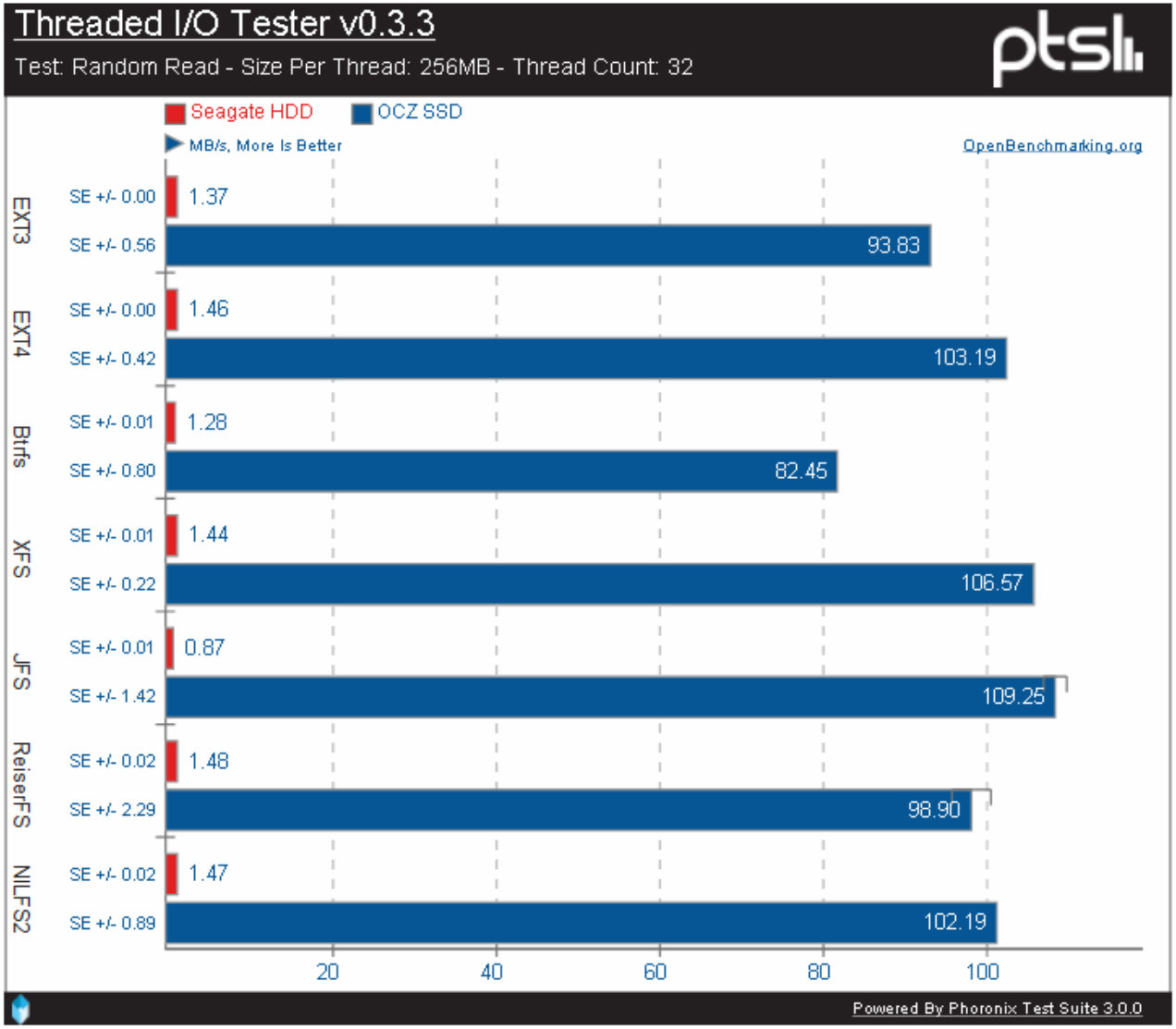
Benchmarky (2)

- http://www.phoronix.com/scan.php?page=article&item=linux_2638_large
- Březen 2011
- Jádro Linuxu 2.6.38
- Na DB serverech opět vyniká (PostgreSQL, SQLite)
- Počet diskových transakcí/s
- Náhodné čtení
- Náhodný zápis

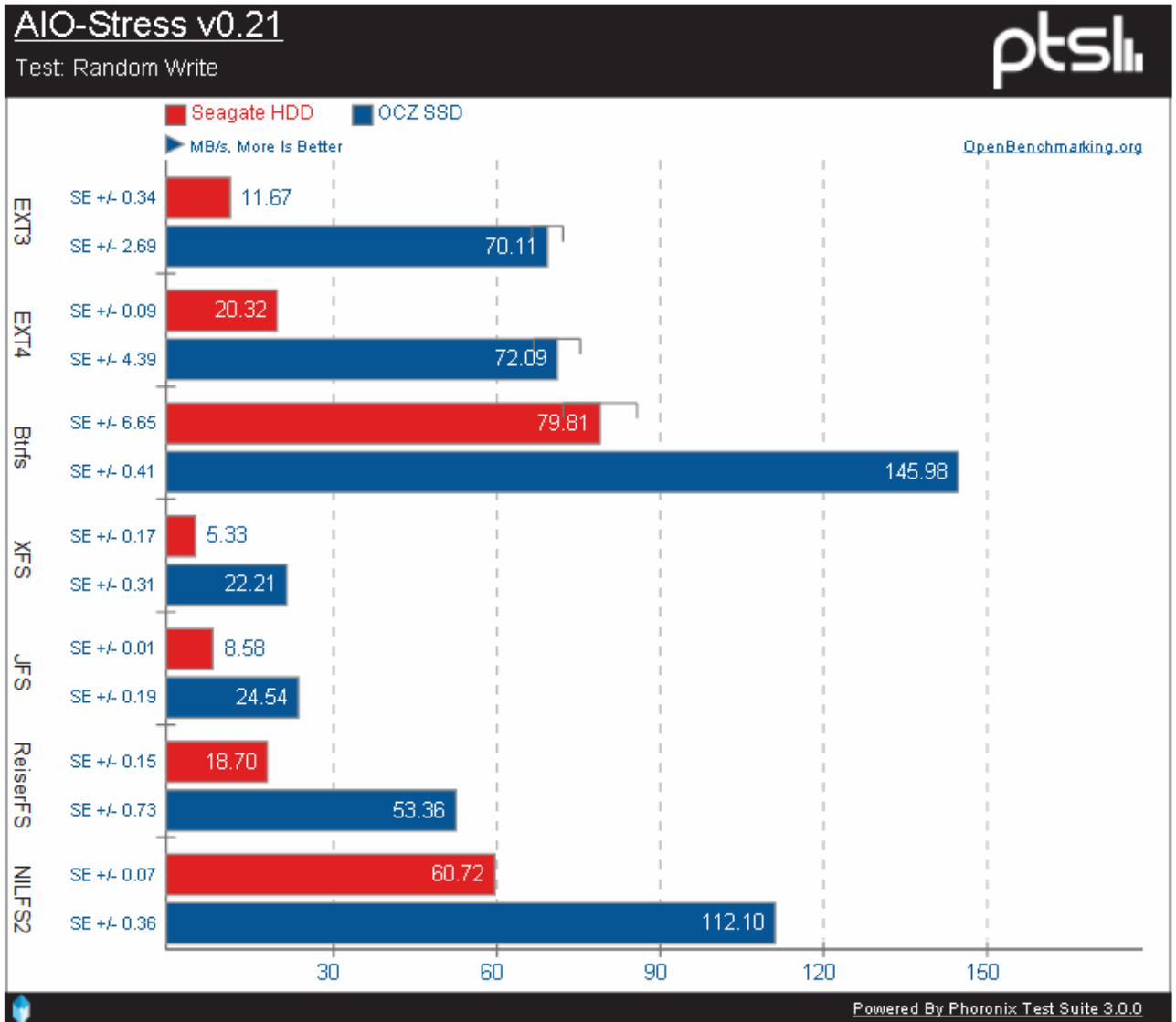
Počet diskových transakcí/s



Náhodné čtení



Náhodný zápis



Zdroje

- http://jfs.sourceforge.net/project/pub/jfslayout.pdf
- http://jfs.sourceforge.net/project/pub/jfslog/jfslog.pdf
- http://jfs.sourceforge.net/project/pub/jfs.pdf
- https://wiki.archlinux.org/index.php/JFS_Filesystem
- http://www.sabi.co.uk/Notes/linuxFS.html
- http://www.softpanorama.org/Internals/Filesystems/jfs.shtml
- http://linuxgazette.net/issue55/florido.html
- http://commandlinemac.blogspot.com/2008/12/30-days-with-jfs.html
- http://www.debian-administration.org/articles/388
- http://www.linux-mag.com/id/1802/
- http://students.mimuw.edu.pl/SO/Projekt03-04/temat5-g7/4.jk-jfs/jfs.html
- http://en.wikipedia.org/wiki/JFS_(file_system)
- http://www.linux-mag.com/id/7518/
- http://www.phoronix.com/scan.php?page=article&item=linux_2638_large&num=1
- http://www.iprint.sk/mereni-vykonu-postgresl-na-ruznych-systemech-souboru/

Otázky