Storage...

co je nového (SSD!) ...
a co se zatím nepovedlo rozbít:-)

Milan Brož
mbroz@redhat.com

LinuxAlt 2010, Brno

redhat

RAID – plán (kernel 2.6.37+)

- **RAID v kernelu...**
  - MD (multiple device) – RAID0,1,5,6,10 ...
  - DM (device-mapper) – RAID0,1, (5)
  - Btrfs?
- **MD/DM**
  - MD wrapper – device-mapper target
  - V DM pouze stripe (RAID0)
- **mdadm ~~dmraid~~**
  - Externí metadata pro fake RAID
  - Již umí Intel DDF (využití libdmraid formátů?)
- **LVM2**
  - Využije target interface k MD

redhat

Co je nového? (jen velmi malý výběr)

- **DM/LVM Snapshot**
  - návrhy pro thin provisioning
  - snapshot merge (2.6.33)
- **LVM2 / DM / LUKS používá udev**
  - spojení dvou rozdílných světů :-)
  - další interakce (udisks, systemd, ...)
  - požadavky na škálovatelnost (tisíce LV)
- **Defaultní zarovnání na 1MB**
  - fdisk, parted, mdadm, lvm2, LUKS
  - Pokud device topology neříká jinak (RAID)
- **DRBD (2.6.33)**
  - „síťový RAID1“ - high availability (2 uzly)

redhat

LVM snapshot (příklad)

- **spojení snapshotu zpět do LV**

```
# lvcreate -s -n lv_snap -l 100%FREE vg_test/lv
...
# lvconvert --merge vg_test/lv_snap
  Merging of volume lv_snap started.
  lv: Merged: 0.0%
  Merge of snapshot into logical volume lv has finished.
  Logical volume "lv_snap" successfully removed
```

- **virtual size**

```
# lvcreate -s --virtualsize 4t -l 100%FREE -n lv_snap vg_test
Logical volume "lv_snap" created

# lvs -a vg_test
LV                VG      Attr   LSize  Origin            Snap%
lv_snap           vg_test swi-a- 60.00m [lv_snap_vorigin]   0.01
[lv_snap_vorigin] vg_test vwi-a-  4.00t
```

redhat

Bariéry (fs) [kernel < 2.6.37]

- **Pořadí zápisů se může měnit**
  - IO plánovač, optimalizace, řízení toku, cache
- **Bariéra**

  „všechny zápisy musí být dokončeny před tím, než se provedou operace následující za bariérou“

  - ~ není to totéž, co paměťová bariéra
  - zajišťění konsistence po ***neočekávaném*** pádu
    - často vázané na operace FS (žurnál), fsync()
    - zajištění pořadí operací
    - vynucení zápisu na disk (ne do cache)
  - … a … je to neefektivní a pomalé
    (~mount FS bez bariér versus bezpečnost dat při pádu)

FLUSH / FUA [2.6.37+]

- **Zkusme to jinak...**
  - *To ale neznamená nutně lépe:-)*
  - ~ explicitní řízení writeback cache
  - dva příznaky IO operace

- **FLUSH**

  „před započetím operace se vyprázdní cache na disk“

- **FUA (Force Unit Access)**

  „ukončení oznámeno až jsou data skutečně na disku“

- zajištění pořadí si musí hlídat FS

- všechny klíčové blokové ovladače FLUSH/FUA respektují

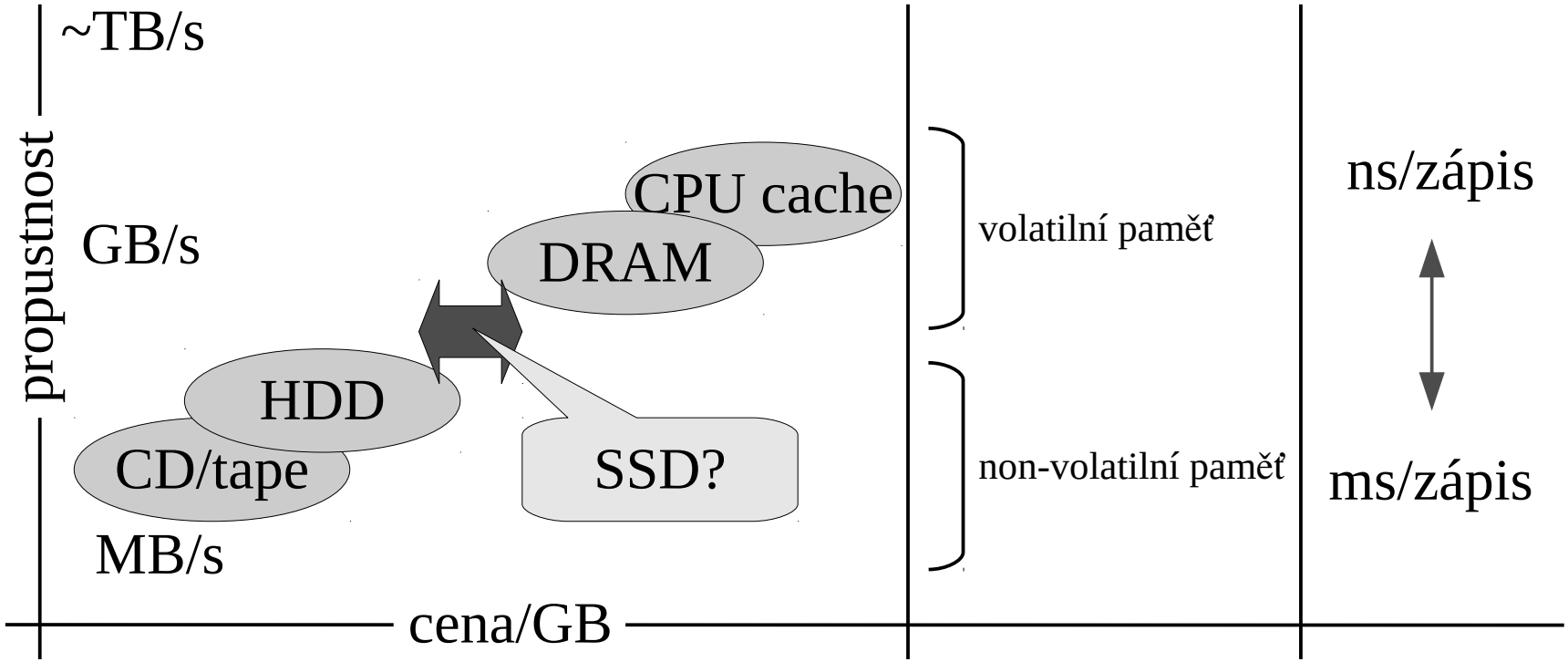
redhat

TRIM, Discard [2.6.36+]

informace (zejména od souborového systému) pro storage, že bloky již neobsahují data

- **TRIM**
  - v terminologii ATA
  - SCSI UNMAP ...
- **Discard**
  - Linux terminologie
  - FITRIM ioctl, IO queue discard flag
- **Kde to má smysl?**
  - *Thin provisioning*
  - *SSD*

SSD (Solid State Drive)

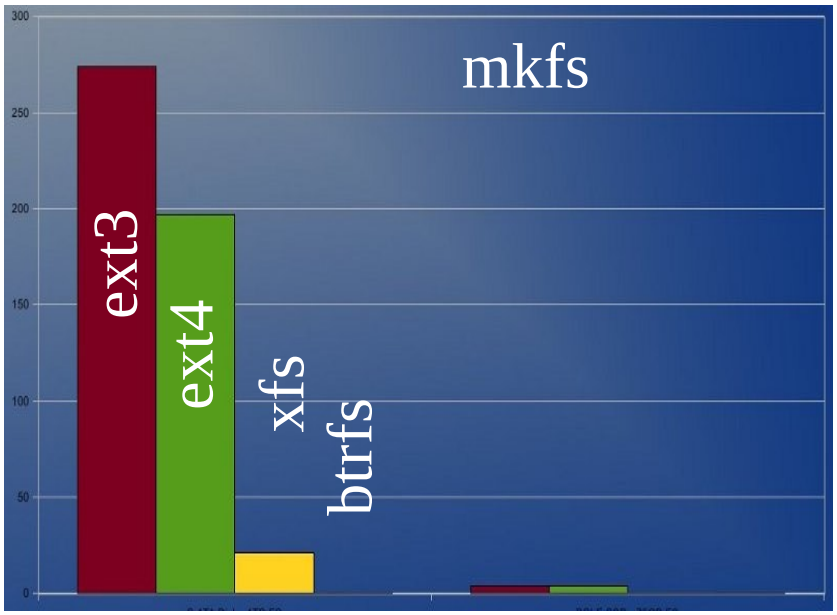


SSD (Solid State Drive)

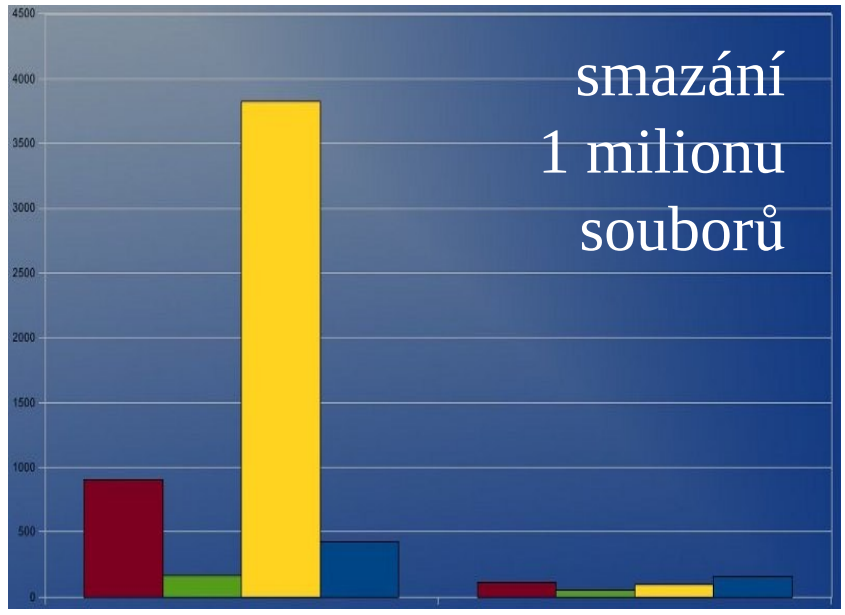
- **SSD – HDD interface**
  SATA / SAS
  "rotating rust" – původně pro disky

- **direct attached SSD**
  blokové zařízení
  přímo na sběrnici (PCI-E)

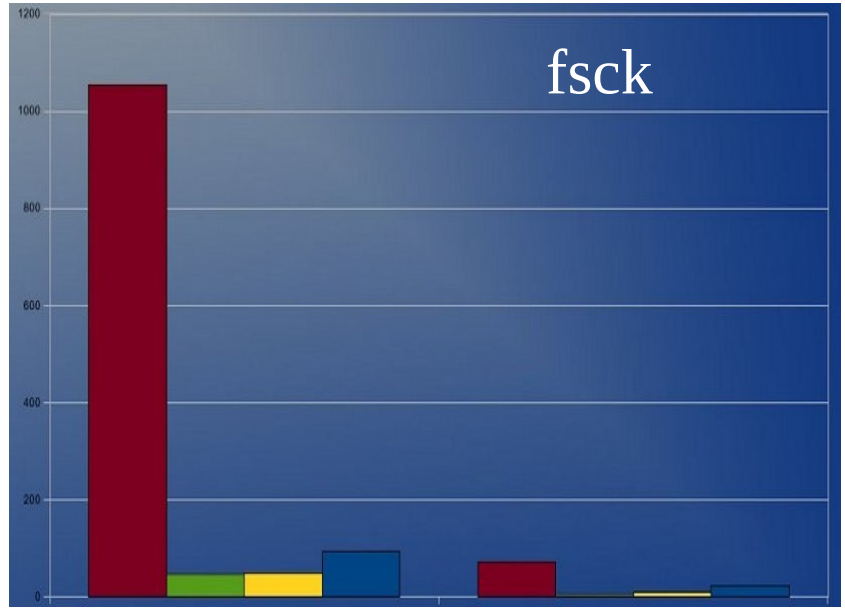
SSD rychlost (Ric Wheeler o škálování FS – Linuxcon 2010)



SATA HDD 1TB FS | PCI-E SSD 75GB FS



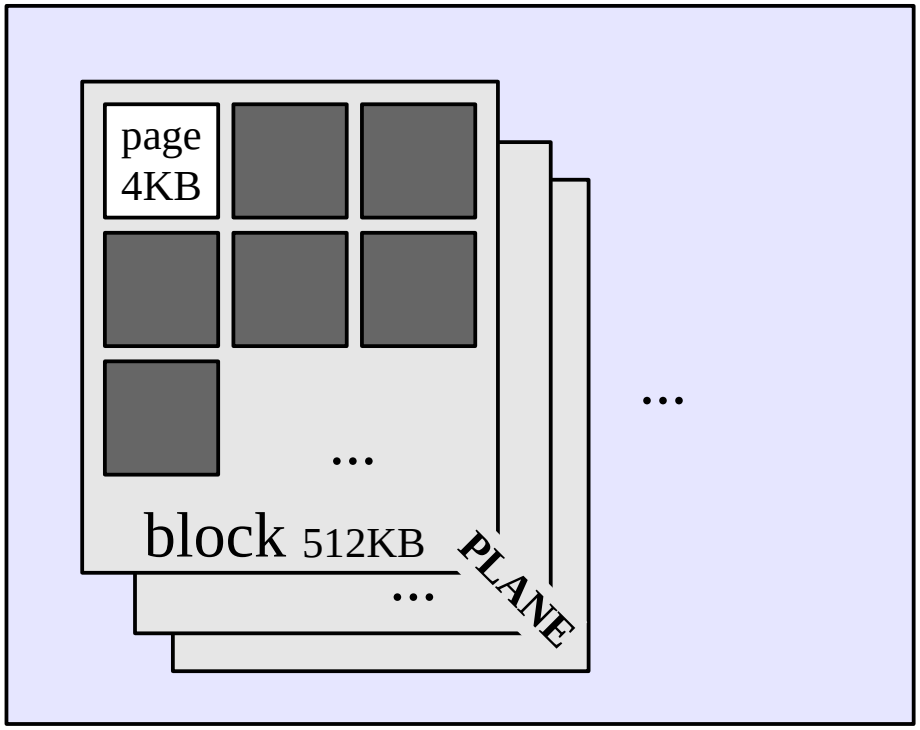
SATA HDD 1TB FS | PCI-E SSD 75GB FS



Koláčové grafy nesmí nikdy chybět ...

redhat

SSD architektura



Příklad
- stránka (page) **4KB**
- blok **512KB** *(128 stránek)*
- plane 512MB *(1024 bloků)*
- ... (zarovnání!)

- Rychlost
  - minimální seek time
  - velmi rychlé čtení
- Atomické operace
  - čtení/zápis – stránka (4KB)
  - vymazání – blok (512KB)

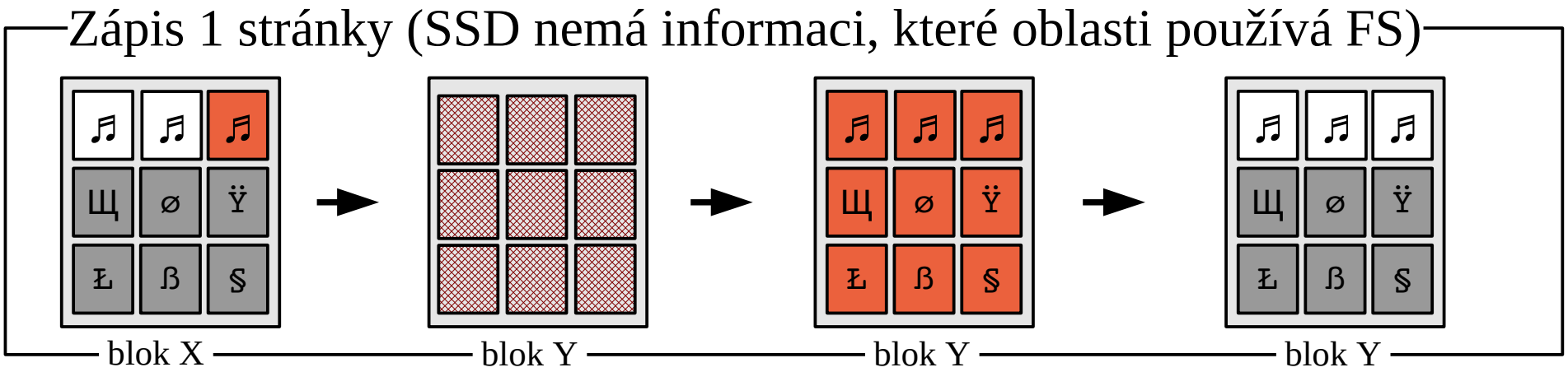
  Přepis stránky
    - *smazání bloku*
    - *znovuzapsání stránek*
- wear leveling
  - omezený počet přepsání
- Interní fragmentace
  - ≠ fragmentace FS

redhat

Není SSD jako SSD

- **Rozdílné HW parametry**
  - Single / Multi / 3bit Level Cell (MLC / SLC / 3BPC)
  - paralelní planes, cache, ...
  - určuje pouze **maximální** výkonnost
- **Zásadní rozdíl je ve firmware**
  - zcela může změnit parametry SSD
  - Garbage Collector (GC)
  - podpora TRIM
  - velikost oblasti vyhrazené pro wear leveling
  - různé (pseudo-)optimalizace zápisů

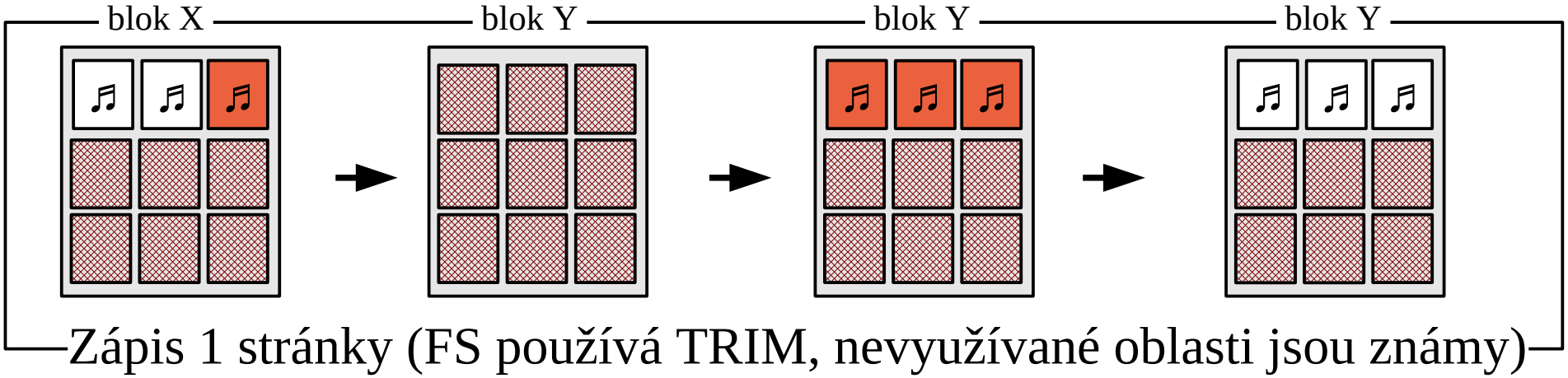
SSD – proč je problém se zápisy



X ≠ Y (wear leveling)

vymazání bloku

zápis stránek



redhat

Podpora TRIM v Linuxu

- **vázané na operace FS**
  - FS obvykle uvolňuje bloky **v malých rozsazích**
    v tomto případě TRIM operace fs zpomalí!
  - **dávkové zpracování** (jednou za čas)
    větší rozsahy, omezení na minimální velikost k uvolnění
- **FITRIM ioctl podpora** [2.6.37+]
  - *ioctl(FITRIM, { start, len, min_len })*
  - nezávislé na konkrétním FS

Nástroje na testování:

- **test-discard** http://sourceforge.net/projects/test-discard/
- **fstrim** http://sourceforge.net/projects/fstrim/

redhat

Podpora TRIM v Linuxu

- **blokové zařízení**
  - SSD, MTD, RAM-based
  - základní device-mapper [2.6.36]
    (linear, stripe, mpath) – pro LVM2 a multipath
  - pro dm-crypt je TRIM bezpečnostní problém
- **souborové systémy**
  - ext4, btrfs, gfs2, nilfs, fat [2.6.36+]
  - plánuje se xfs (a další)
- **e2fsprogs (ext4)**
  - discard před mkfs
  - zrychlení inicializace inode tabulky
    (pokud storage vrací 0 pro nealokované bloky)
  - plánuje se fsck (TRIM nealokovaných bloků)

redhat

SSD / TRIM v Linuxu

- **nová technologie**
  - ne vše je ideální
  - chyby, nekompatibility
    včetně problémů s firmware s TRIM
  - je třeba ještě hodně testování

- **nutné změny v blokové vrstvě a FS**
  limity blokové vrstvy a FS
  neumožňují plně využít potenciál SSD

  - evoluce jako u síťových driverů (1Mb/s ... 10Gb/s)
  - škálování do milionů operací za sekundu
  - FS: miliarda souborů

  ...

1965

Bryant Series 4000

~ $0.0006 per byte

2010

FusionIO ioXtreme

~ $0.0000000011 per byte