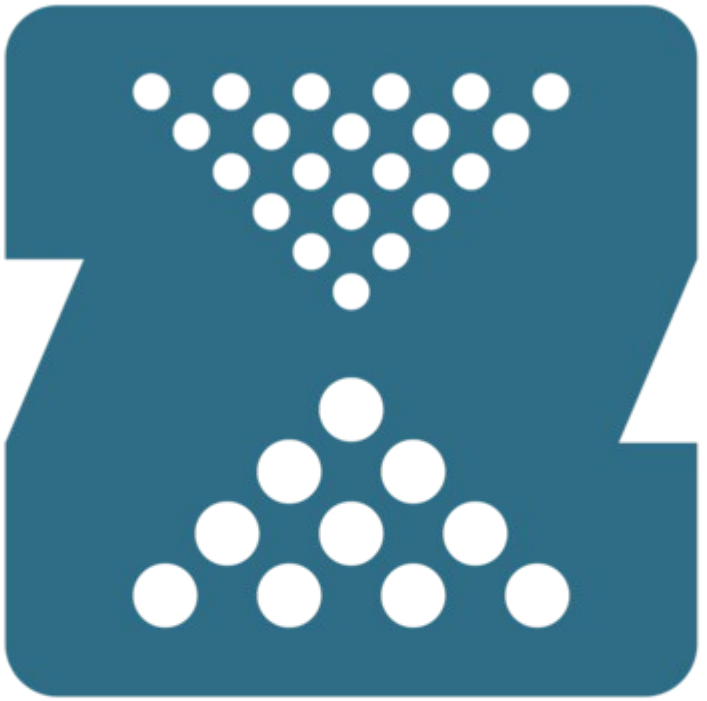
ZFS na Linuxu



OpenZFS

whoami

- Pavel Šnajdr
  - Fulltime SysOP
    - S Linuxem dělám v produkčním nasazení už 6 let
    - Cokoliv pod aplikační vrstvou je moje – OS, jádro, kontejnery, FS a perf
  - Předseda spolku vpsFree.cz
    - Hosting VPS na neziskovém principu
    - Kdo neznáte nebo chcete vědět víc – stánek
  - Relbit CTO
    - Evia Project (http://eviaproject.org)
      - PaaS Operační systém pro provoz PHP/MySQL bez pracného adminování
      - Koho baví dělat ty samé admin tasky pořád dokola

vpsFree.cz

OpenZFS

About talk

- Problémy, které ZFS řeší
- Historie a současný vývoj
- ZFS on Linux
- ZFS design
- Vlastnosti ZFS
- Nasazení ve vpsFree.cz
- Q&A

vpsFree.cz

OpenZFS

Návrhové cíle ZFS

- Škálovatelnost
  - Musí obsloužit obrovské objemy dat
  - Terabajt je nic, jsou k vidění PB pole
  - Stamiliony souborů, adresářů
- Integrita dat = nechci přijít o data
  - Ochrana proti náhodnému poškození dat na disku i cestě mezi diskem a OS
  - Ztráta napájení
- Běžný hardware
  - CPU výkonu je dost, HW RAID ztrácí opodstatnění
  - “ZFS loves cheap disk”
- Šetřit psychické zdraví administrátorů
  - Důraz na maximální jednoduchost použití a redukci práce admina

Zjednodušení storage modelu

- Storage dneška == minimálně 3 vrstvy
  - RAID, volume manager a filesystem
- Vrstvení přináší umělé problémy
  - Fixní velikost bloku, zarovnaní bloků
  - Online změny v konfiguraci jsou problém
    - FS shrink nemožný
  - Jedna vrstva neví o druhé
    - Snapshoty

Historie ZFS

- 2001 Sun Microsystems, J. Bonwick & M. Ahrens
- 2005 ZFS otevřeno spolu s OpenSolarisem, CDDL
- 2006 Produkční nasazení zákazníkům – Solaris 10u2
  - Tzn. ZFS je testováno ohněm už 8 let
- 2010 Oracle kupuje Sun, ruší OpenSolaris, illumos fork
  - ZFS pod CDDL žije dál vlastním životem
  - “Solaris family” se rozutekla ke startupům pracovat na Illumosu
- 2013 OpenZFS project
  - M. Ahrens a B. Behlendorf

OpenZFS

- OpenZFS (http://open-zfs.org)
  - Komunity okolo ostatních OS portovaly ZFS k sobě
  - Hrozilo, že by se implementace rozcházely
  - Deštník, pod kterým se sdílí kód mezi implementacemi
    - Illumos
    - FreeBSD
    - ZFSonLinux
    - MacZFS
  - Komerční zájem vyvíjet OpenZFS existuje
    - Nexenta, Delphix, Joyent, ClusterHQ, ...

Konkurence ZFS

- NetApp (WAFL)
  - Closed-source pro jejich storage řešení
- btrfs
  - GPL => Linux-only
  - Mladý a málo otestovaný pro různé typy zátěže
  - Poněkud méně admin-friendly
  - Nemá alternativu ARC, ZVOL, zfetch, ...
  - Má potenciál do budoucna

vpsFree.cz

OpenZFS

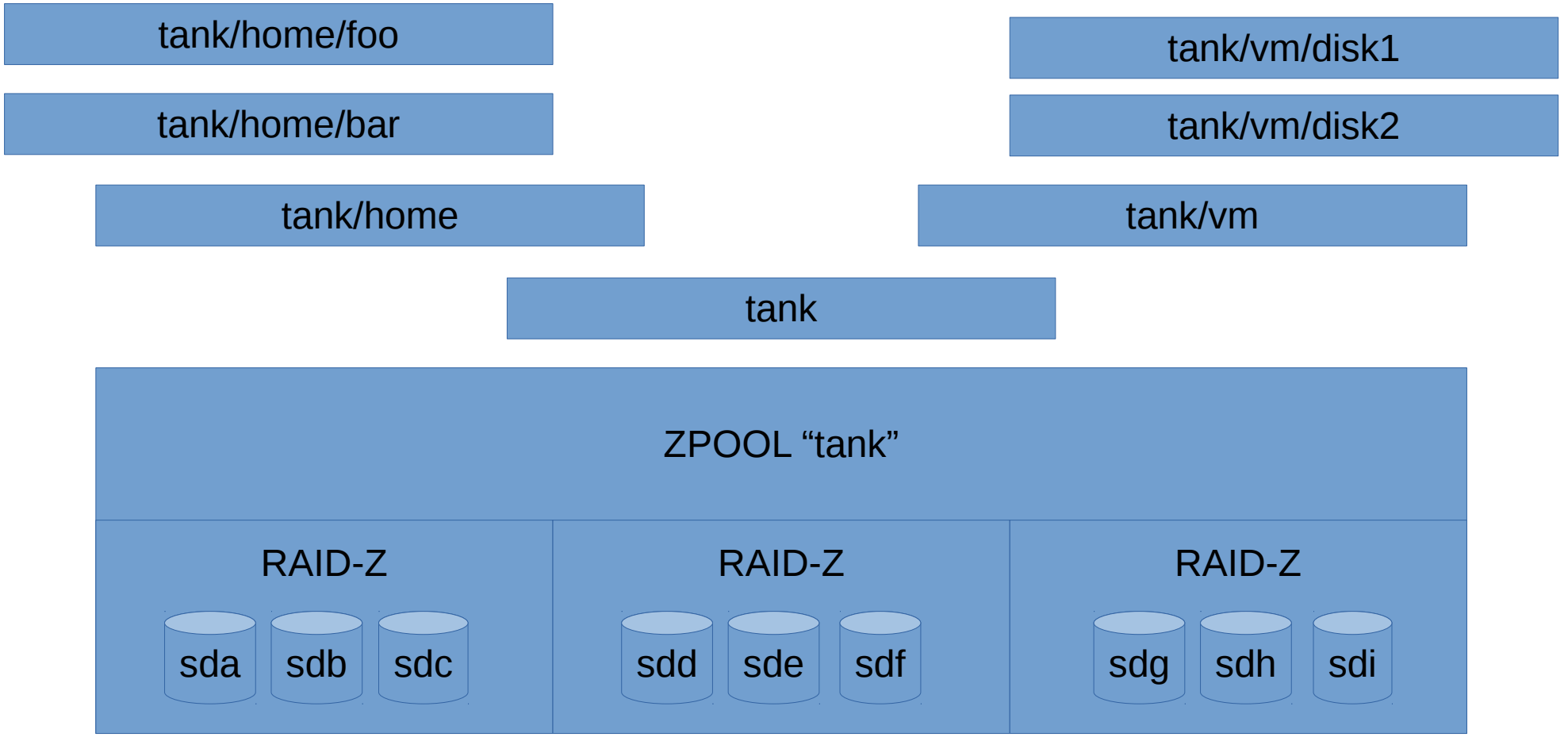
ZFS

- 128 bit design
  - Teoretické limity: sky's the limit
    - 2^{78} bytes/pool
    - 2^{64} bytes/filesystem
    - 2^{64} devices/pool
    - 2^{64} pools/system
  - Ext4 maximum 2^{32} inodes
    - 2^{48} entries/directory

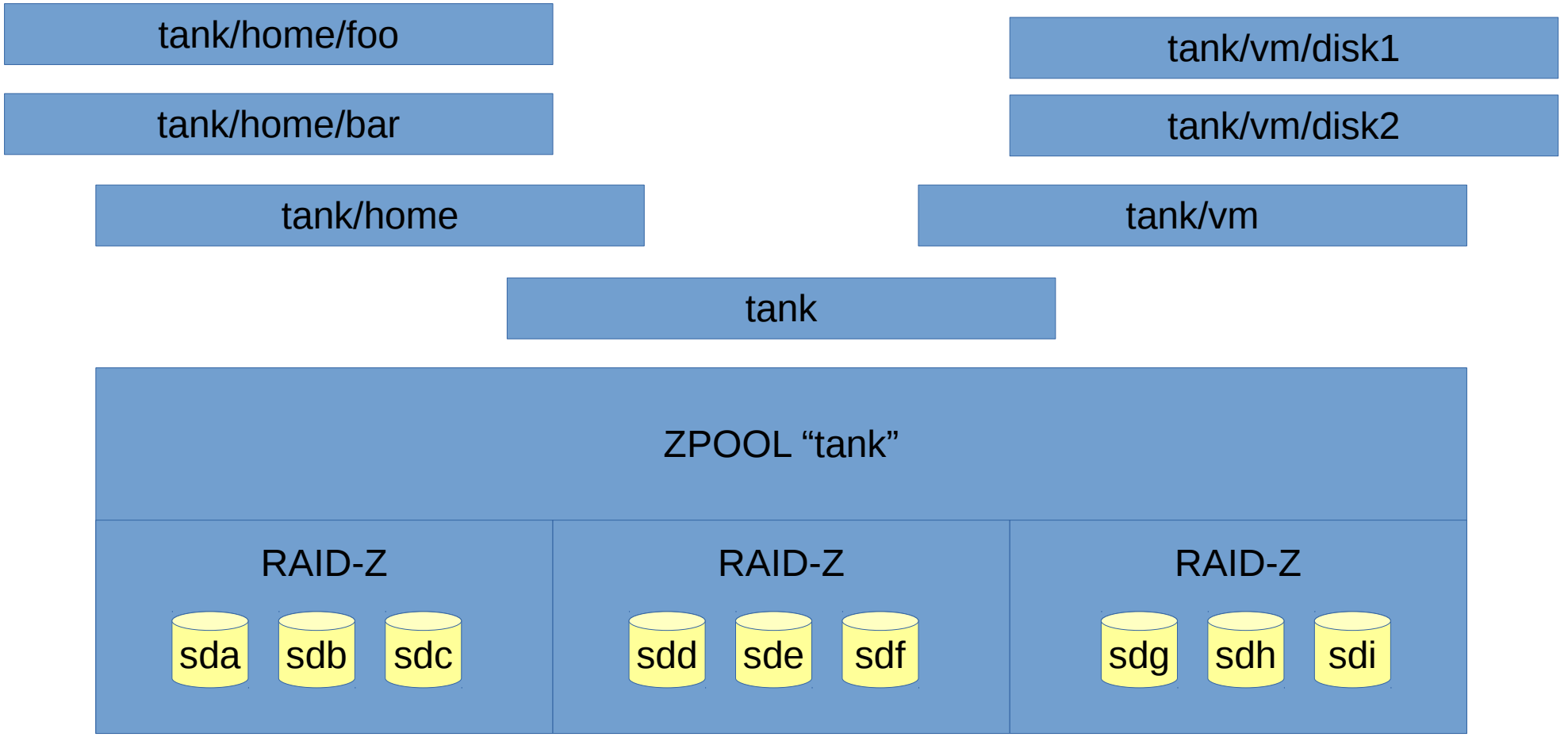
Pooled storage

- zpool
  - Kombinuje FS a volume manager v jednom
  - Physical VDEV: disk, soubor
  - Virtual VDEV: mirror, raid-z (-z2, -z3)
    - any write = full stripe write, no read-modify-write
  - VDEVy jde přidávat za běhu, ale odebrat nikdy
    - ZFS neumí realokaci bloků
  - Dynamická velikost bloku
- Datasety
  - Hierarchická struktura s dědičností nastavení
  - Filesystem
  - ZVOL

Pooled storage



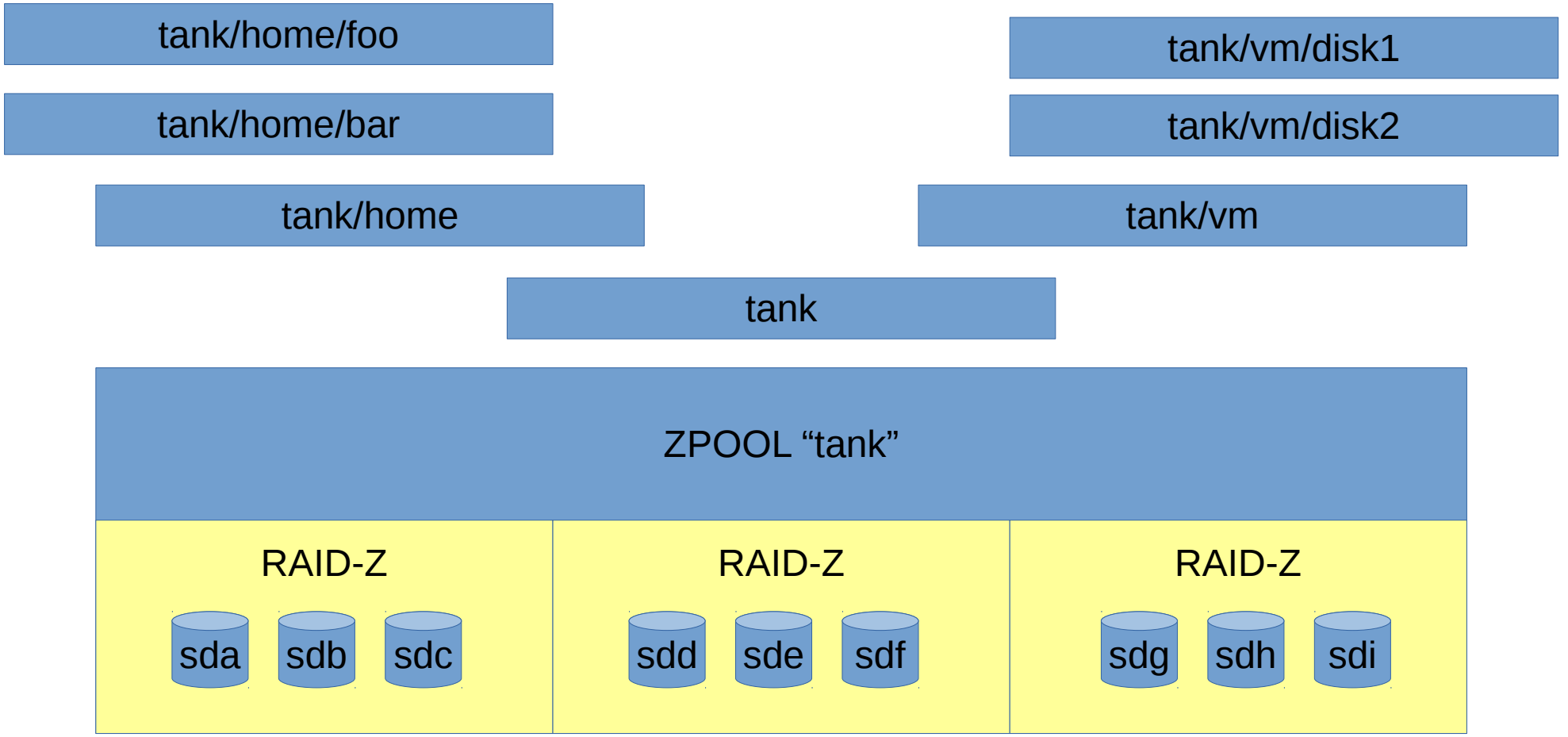
Pooled storage



vpsFree.cz

OpenZFS

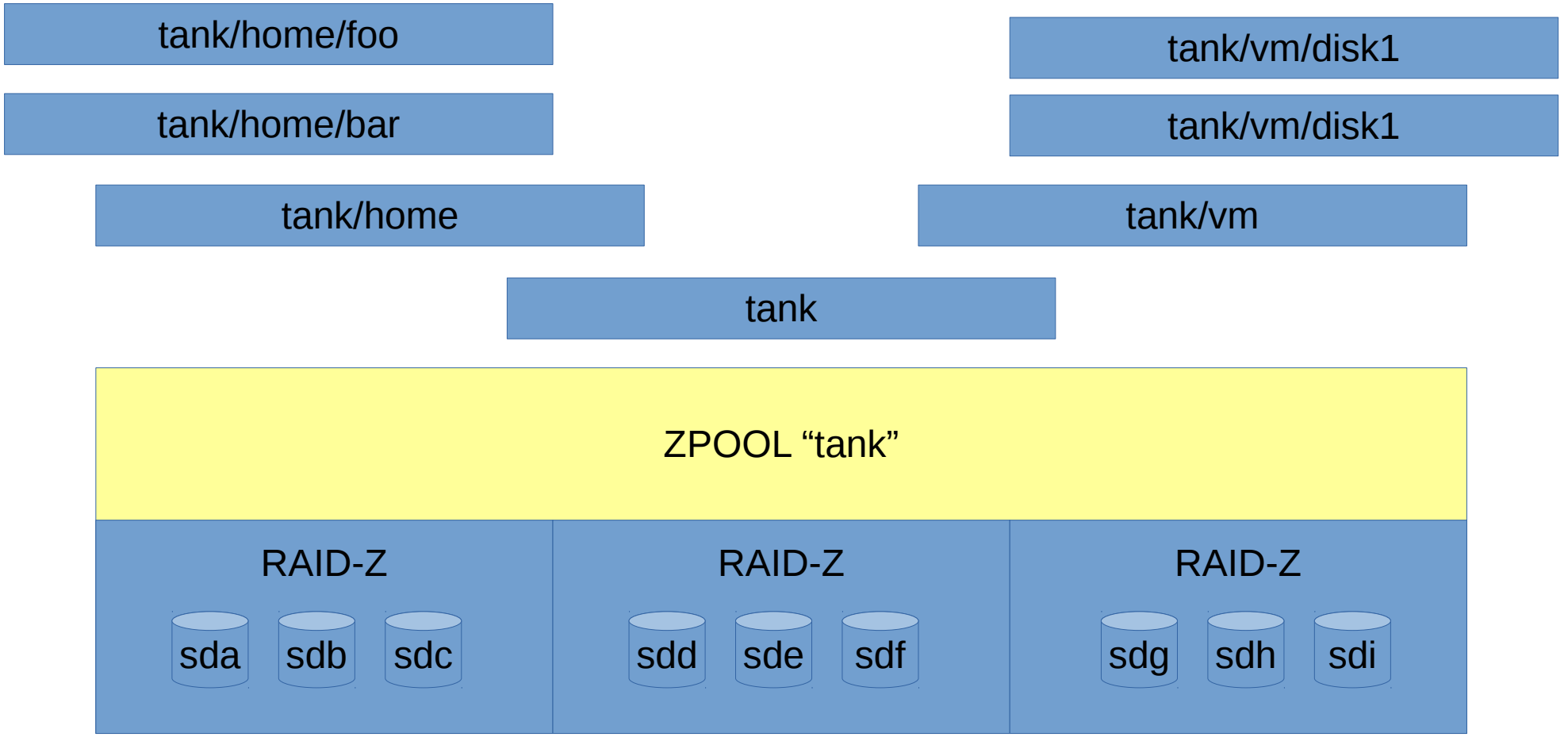
Pooled storage



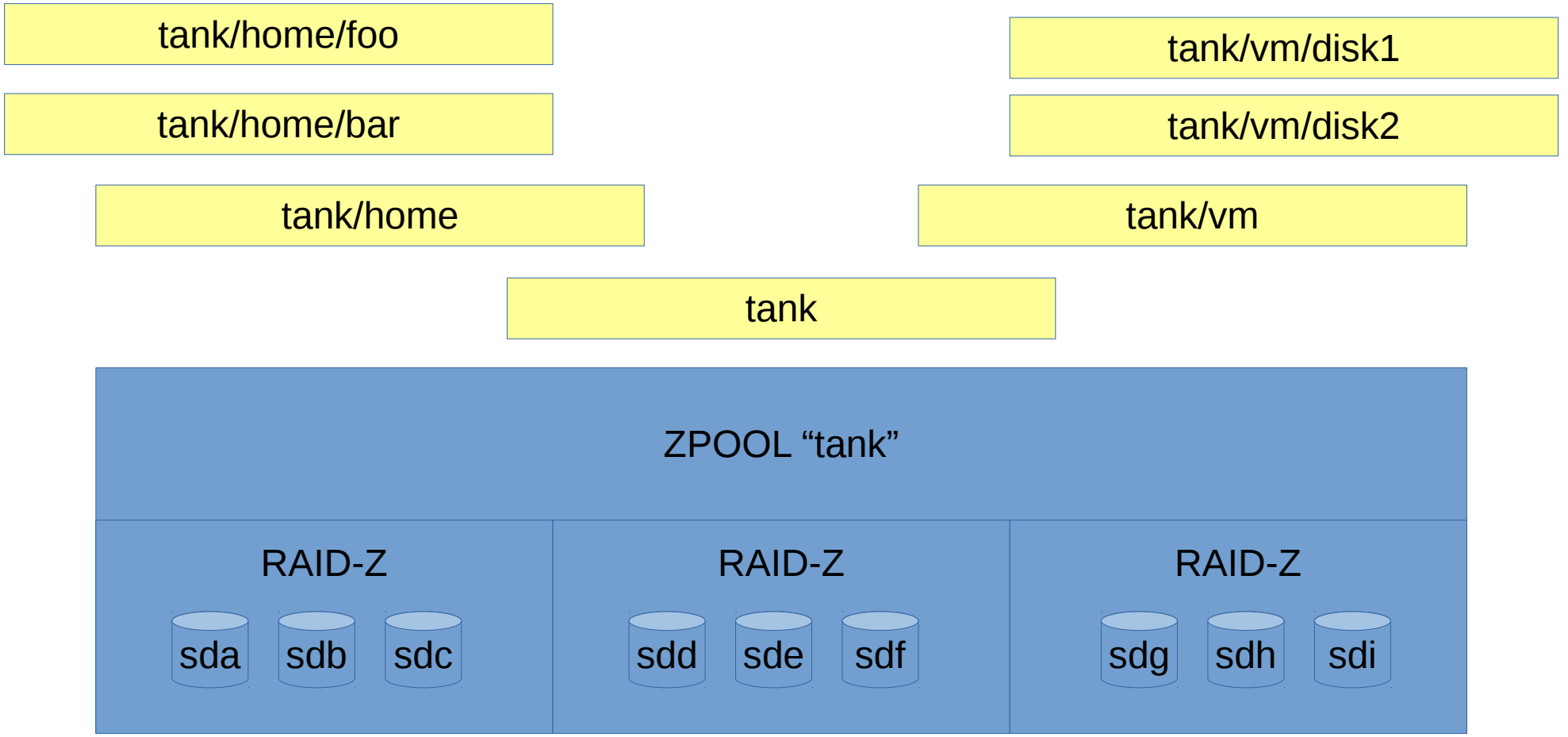
vpsFree.cz

OpenZFS

Pooled storage



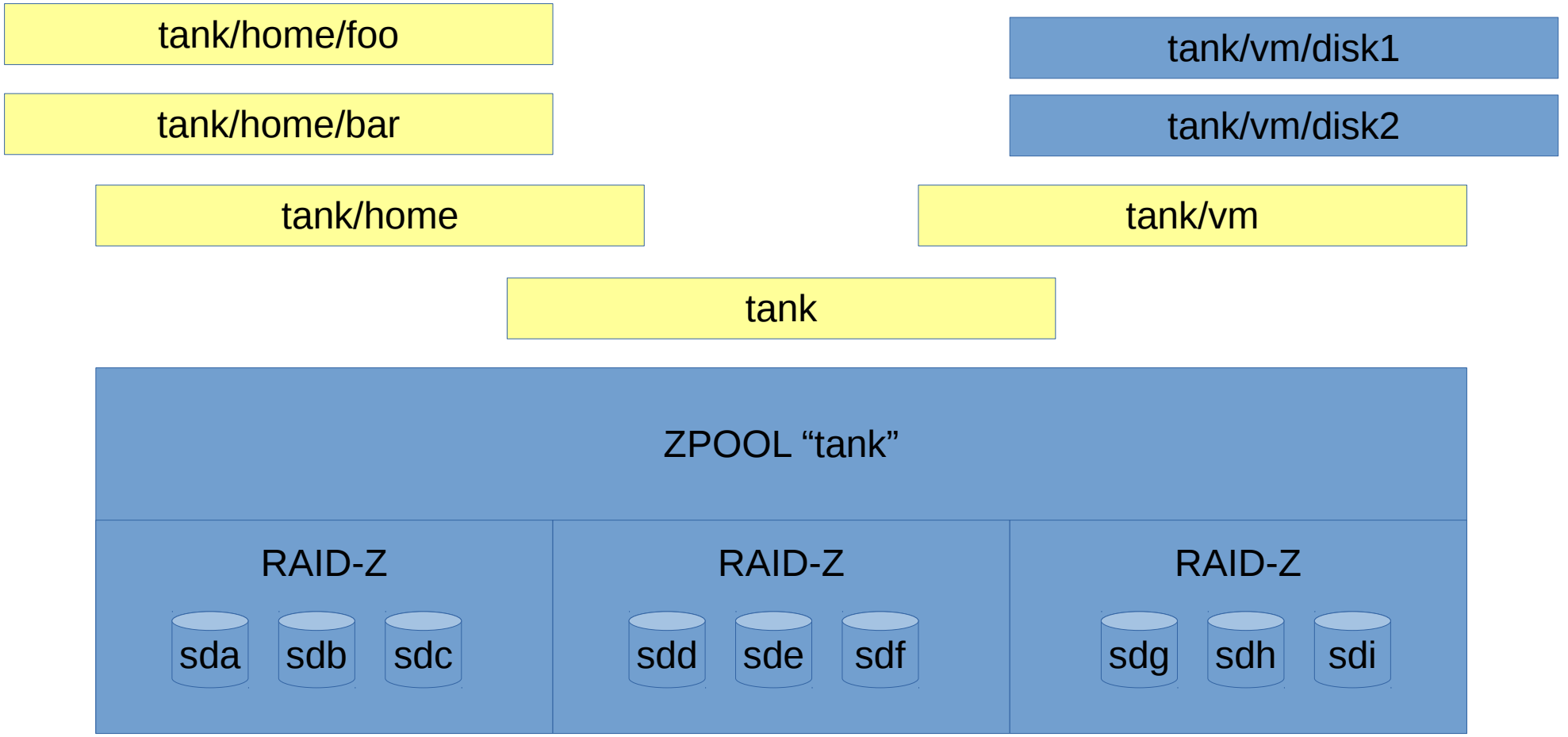
Pooled storage



vpsFree.cz

OpenZFS

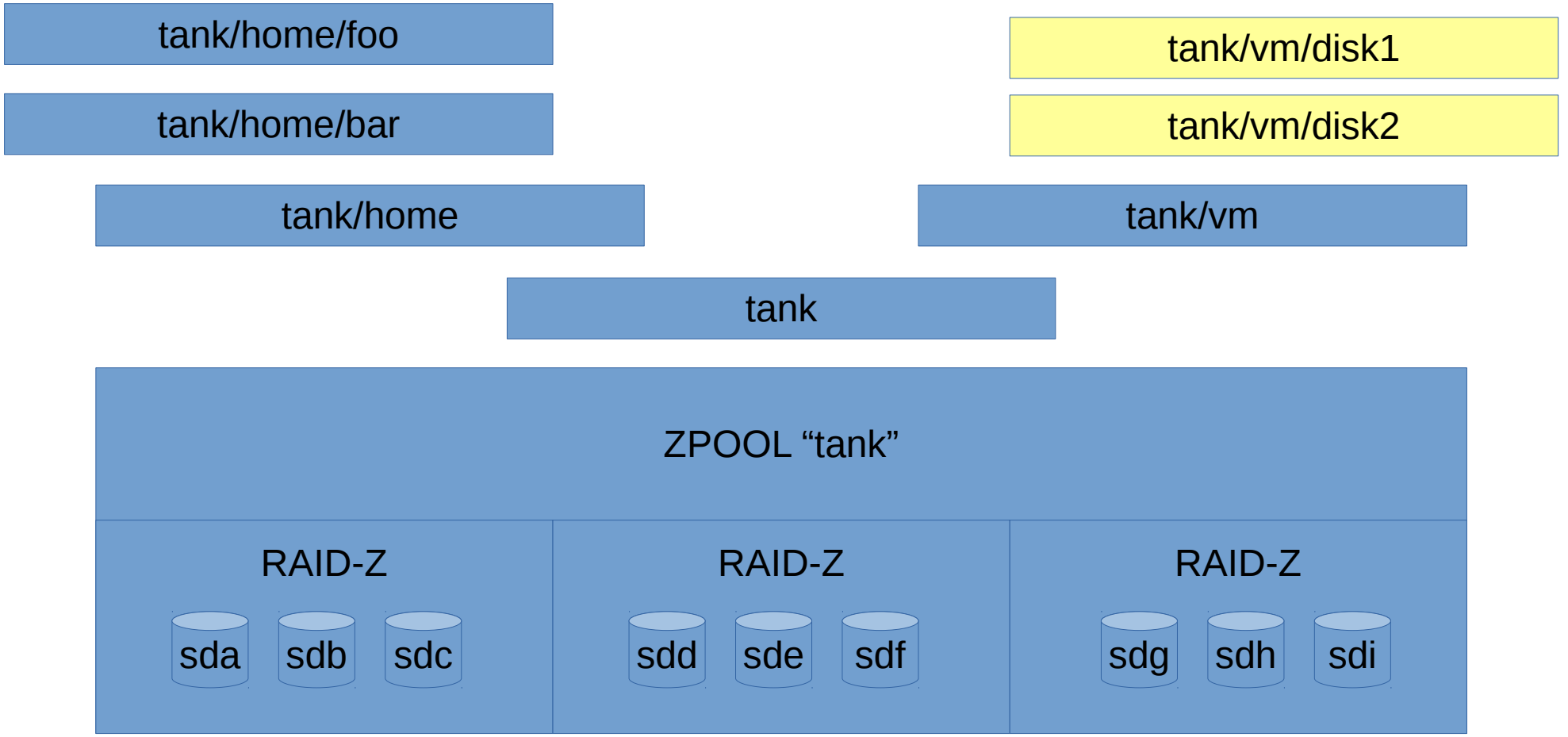
Pooled storage



vpsFree.cz

OpenZFS

Pooled storage

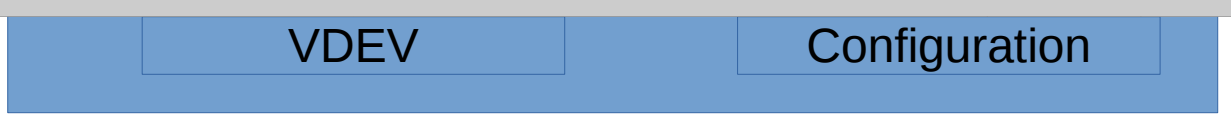


vpsFree.cz

OpenZFS

ZFS Architecture

- Architektura ZFS kodu
- 3 hlavni vrstvy, ktere jsou uzce provazane
- Nebudu je popisovat do detailu, hodi se vedet, ze existuji a jakou maji na sebe navaznost



OpenZFS

ZFS Architecture

- Pooled storage layer
  - Vytváří adresní prostor nad disky, z kterého lze alokovat bloky pro objekty z vyssi vrstvy
  - Stará se o redundanci uložení dat a obsluhu fyzického hardware
  - Je zodpovědná za cachování

VDEV

Configuration

OpenZFS

ZFS Architecture

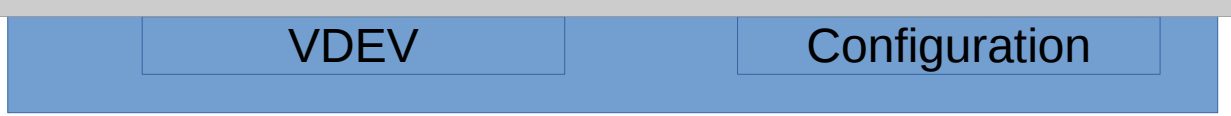
- Transactional object layer
- Bloky dat tvoří strom objektů
- Druhy objektů – soubor, adresář, dataset
- Změny objektů jsou transakce, dokončené jsou po úspěšném uložení na disky

ZFS Architecture

- Transakce se skládají v čase za sebe do txg
  - Potom jsou periodicky zapisovány na disky
  - Po dokončení txg se přepisuje uberblock
  - txg číslovány - “kolikátá txg od vzniku poolu”
  - Každý objekt si eviduje číslo txg, kdy byl zapsán

ZFS Architecture

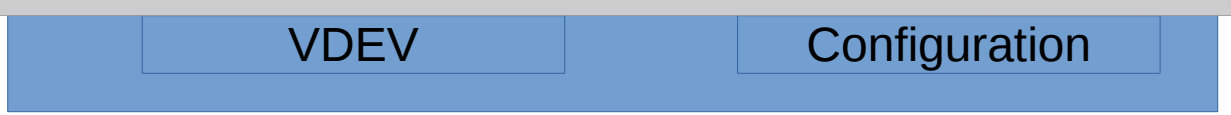
- Poslední vrstvou jsou rozhraní do uživatel. prostoru
- ZPL mapuje ZFS objekty na POSIXový FS
- ZVOL poskytuje rozhraní blokového zařízení
- /dev/zfs umožňuje manipulaci se ZFS v jádře



OpenZFS

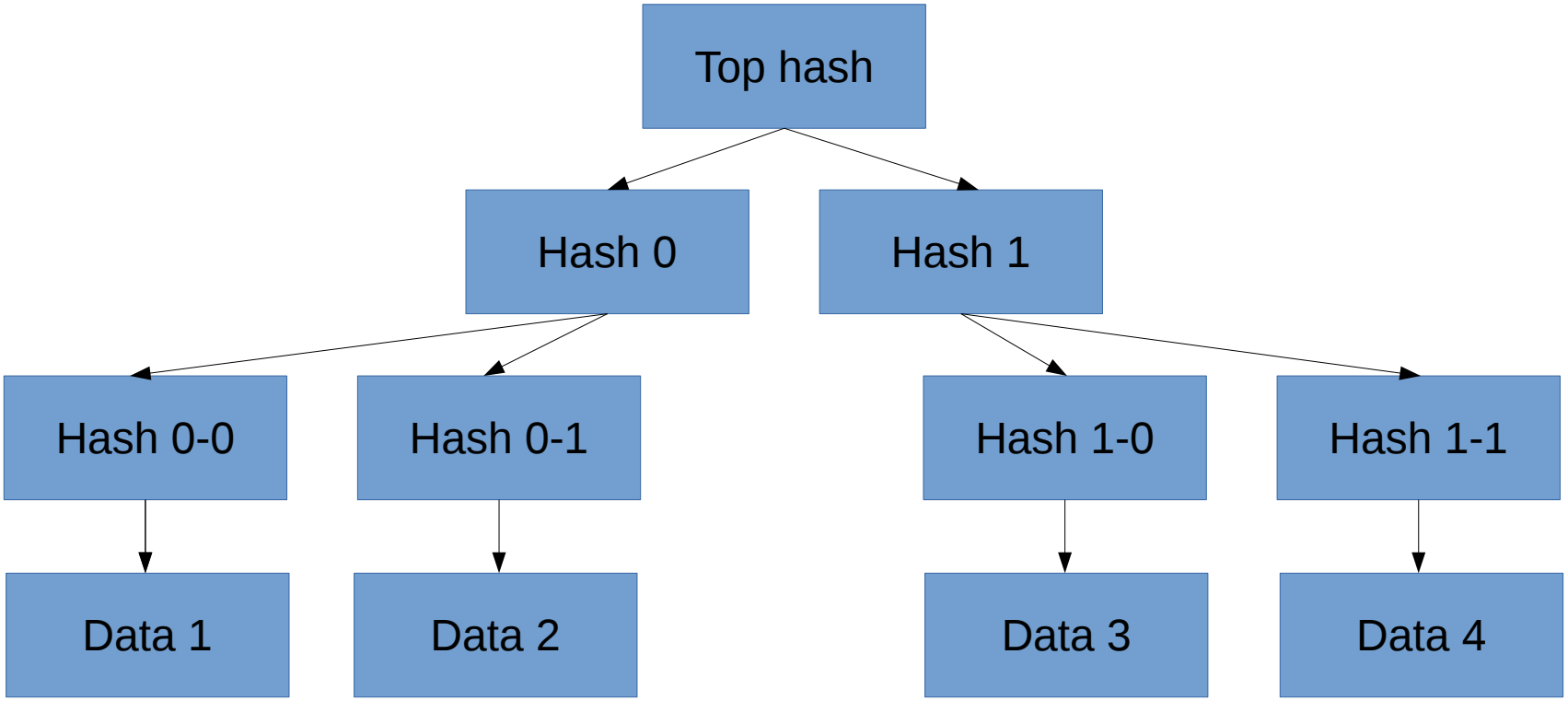
ZFS Architecture

- V uživatelském prostoru s /dev/zfs pracuje libzfs
- Kterou používají zfs commandline utility



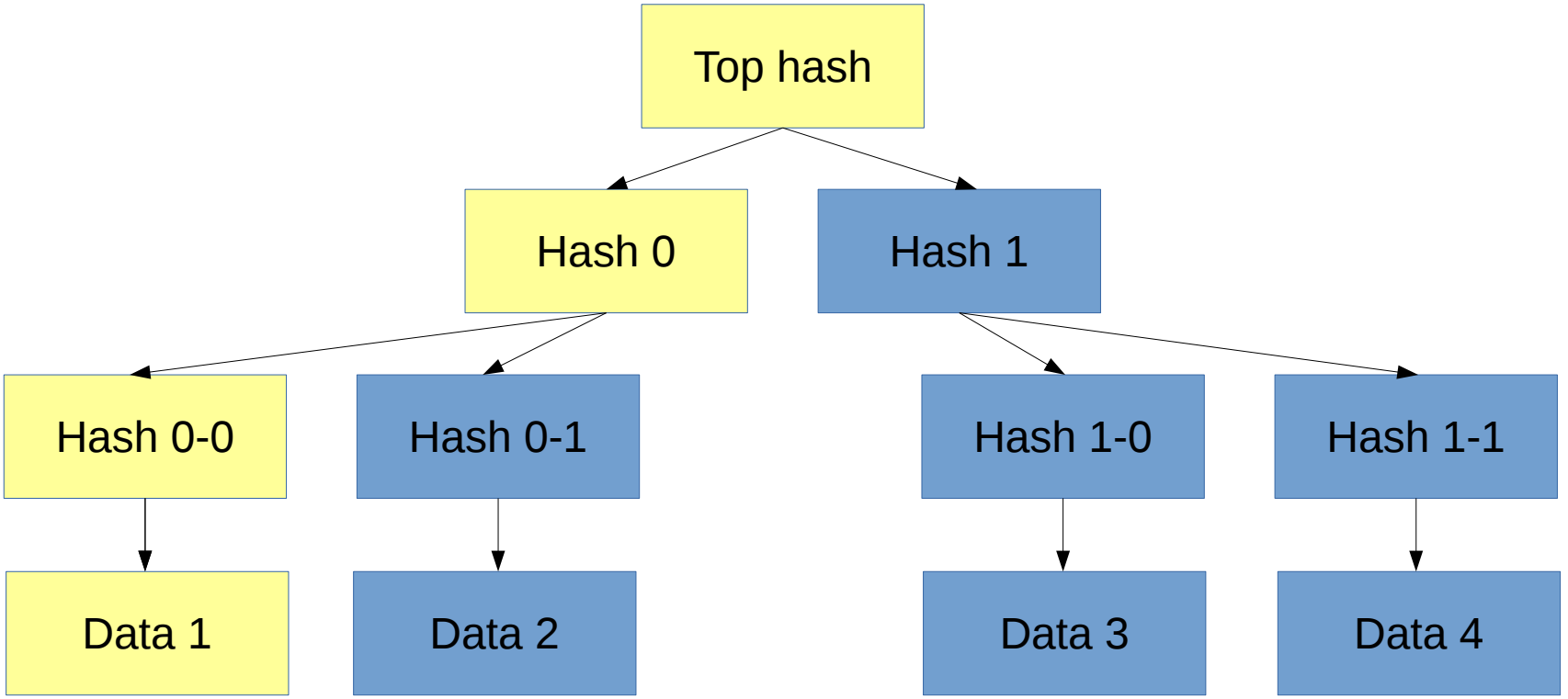
OpenZFS

Merkle tree



- Jak ZFS zajišťuje integritu ukládaných dat? Používá checksumy.
- Checksum datového bloku je uložený vždy v nadřazeném nepřímém bloku
- Kontrola při každém čtení, pokud nesedí, čte se z jiného disku, když je odkud

Merkle tree

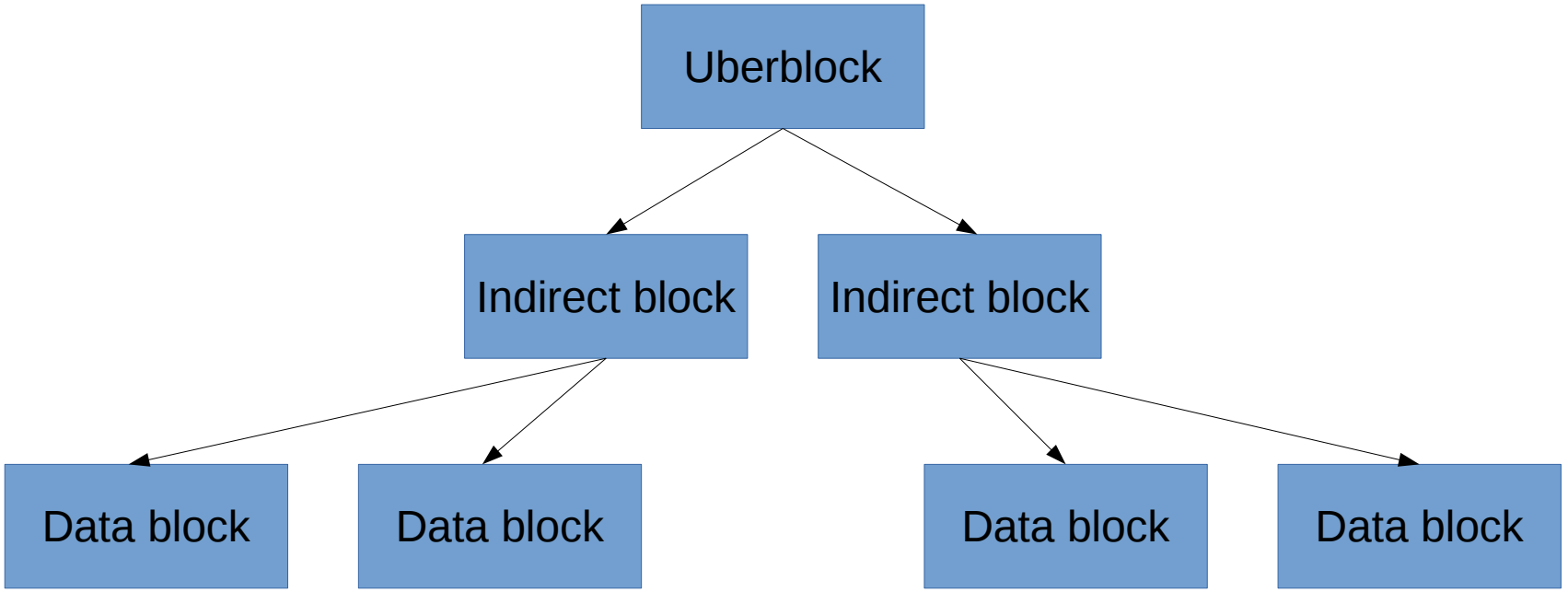


- Při změně datového bloku se musí přepočítat všechny nadřazené checksumy
- 256 bit checksum (fletcher2, fletcher4, SHA256)

vpsFree.cz

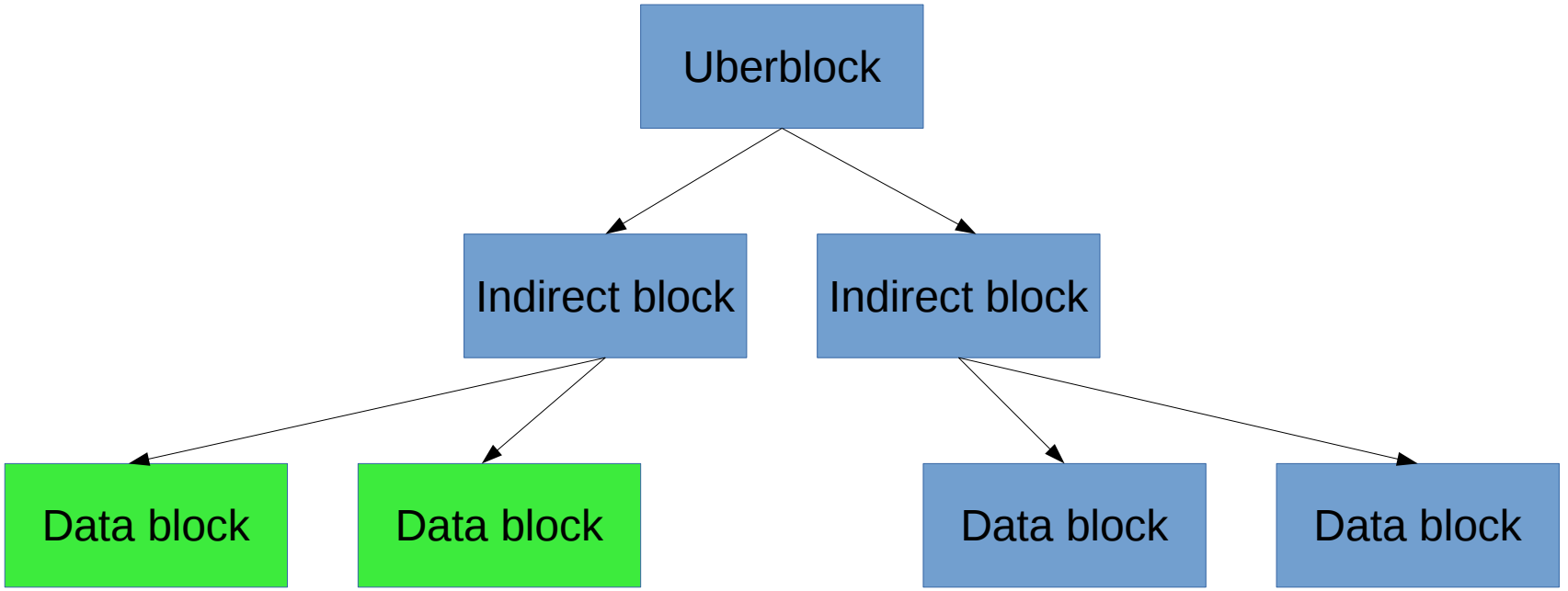
OpenZFS

Transactional CoW



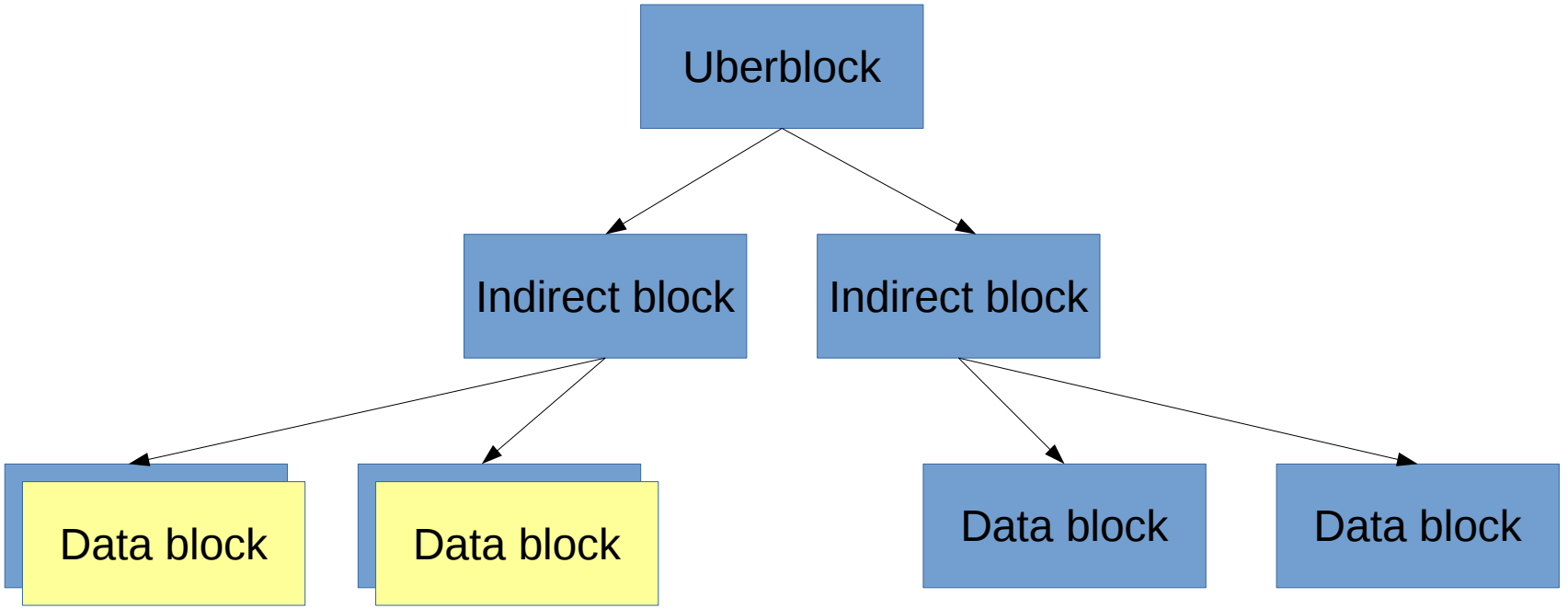
- Copy on write = při změně dat se bloky nepřepisují, ukládají se na nové místo
- Když přimícháme transakce a Merkle tree, dostaneme přesný obrázek, jak ZFS pod kapotou pracuje

Transactional CoW



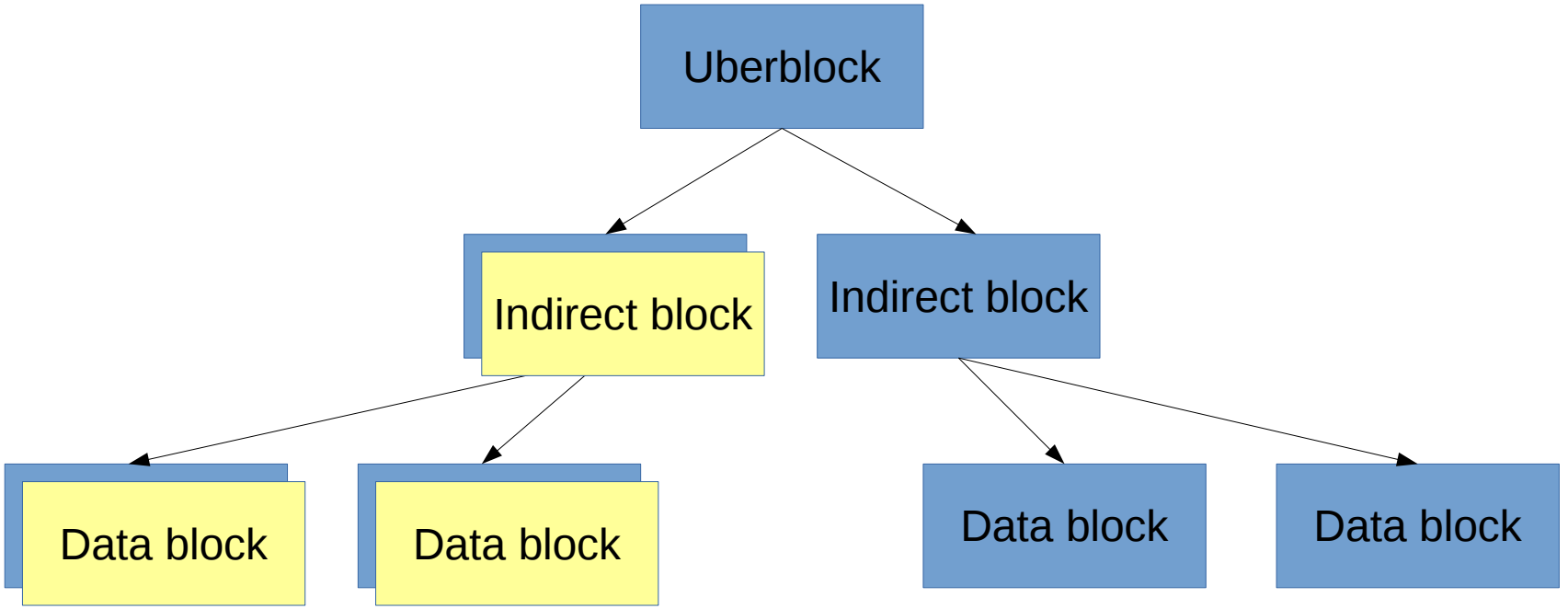
- Řekněme, že chci změnit následující dva bloky dat

Transactional CoW



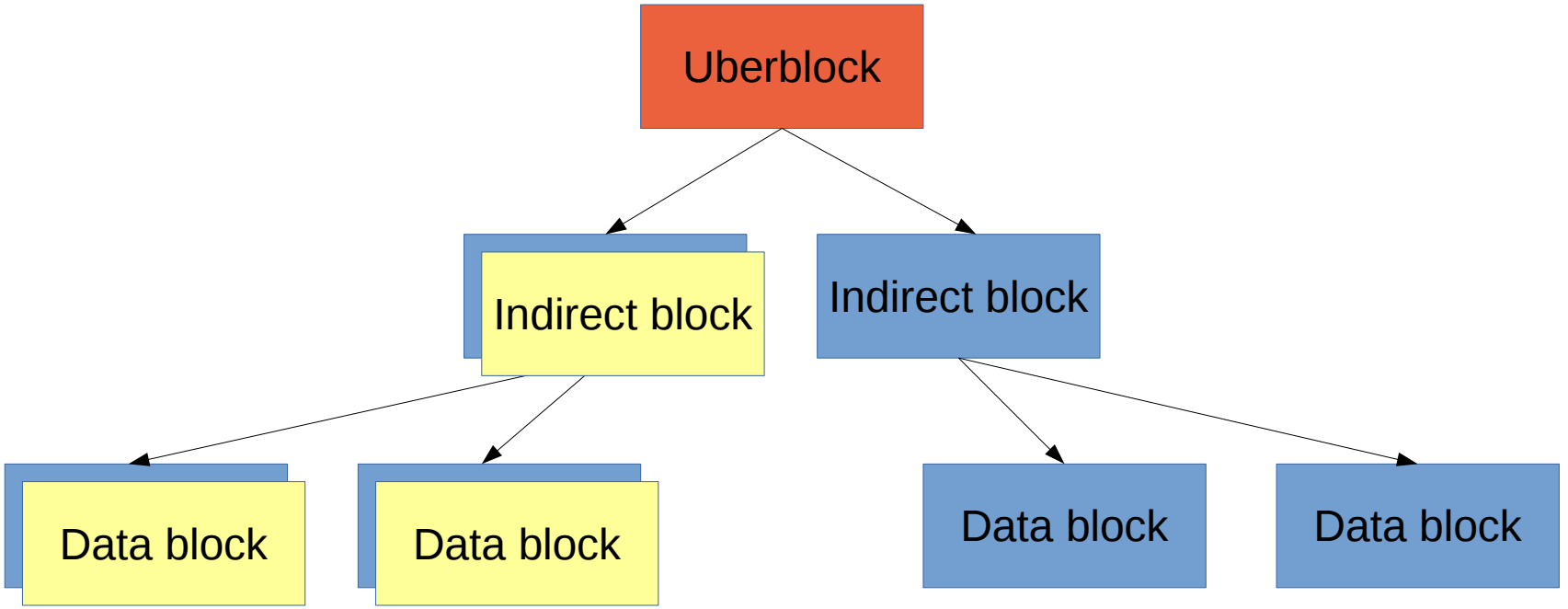
- Nejdřív zapíšeme jejich novou verzi na nové místo na disku

Transactional CoW



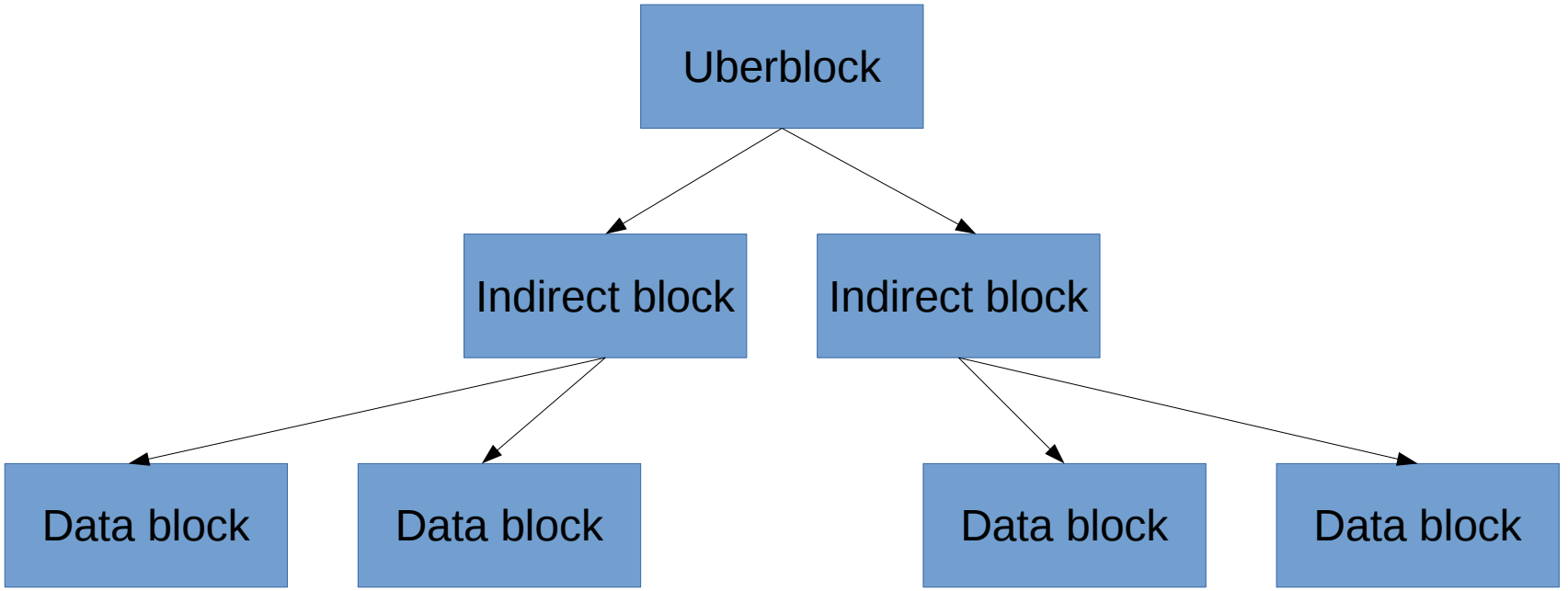
- Uložíme jejich checksumy do bloku, který na ně ukazuje a to opakujeme, dokud nenarazíme na vrchol stromu
  - Uberblock

Transactional CoW



- Potom v jednou atomickou operací na konci úspěšného syncu txg přepíšeme uberblock

Transactional CoW



- Původní bloky můžeme uvolnit, pokud se nemají stát součástí snapshotu a začíná se odznova
- Takhle je ZFS 100% času konzistentní na disku a nepotřebuje fsck

vpsFree.cz

OpenZFS

ZFS on Linux

- http://zfsonlinux.org
- Projekt sponzorovaný LLNL
  - IBM Sequoia
    - ZVOL pro data a metadata Lustre FS
  - 2008 první release, chybí podpora FS, jenom ZVOL
  - 2013 Brian Behlendorf oznamuje ZoL 0.6.2

- Komunita okolo ZoL stále roste
- Github https://github.com/zfsonlinux/zfs

vpsFree.cz

OpenZFS

Portování ZFS na Linux

- ZFS je univerzální a kompaktní balík kódu
  - Dá se sestavit pro kernel i user space
- Linux není Solaris
  - SPL (“Solaris Porting Layer”)
  - Správa paměti - Linux VM subsys se hodně liší
  - Linuxu specifické změny
    - IO reordering dělá Linux
    - /dev/zfs, ZVOL a VDEV komponenty obsahují drobné změny
    - ZPL (POSIX layer) je kompletně přepsaný
  - Všechen ostatní kód je sdílený v rámci OpenZFS projektu
    - Naprostá většina ZFS logiky

ZFS on Linux

- Současná stabilní verze je 0.6.3
  - Stabilní pro produkční nasazení
  - Podpora pro Linux 3.14
  - Podporuje POSIX ACL a SELinux
  - ZFS Event Daemon
    - Mail v případě úmrtí disku, autoreplace, autoexpand
  - Balíky pro RHEL klony, Debian, Ubuntu, Arch, Gentoo...

- ZoL projekt ma automatizované testování
  - Příliš mnoho distribucí, příliš mnoho verzí jádra

ZFS on Linux

- Největší “problém”:
  - GPL není kompatibilní s CDDL
    - ZFS nebude nikdy začleneno do Linuxu
    - Podobná pozice jako nVidia blob drivery
    - Nemůže využívat GPL-only export symboly
      - Nemožnost legalní integrace s uprobes
      - Pro ZVOL chybí /sys/block/ položky
      - .zfs/snapshot nemůže používat automounter
      - ...

ZFS on Linux

- Rozpracovaná zlepšení
  - AIO support (0.6.4)
  - ZVOL rework (0.6.4)
  - Persistent L2ARC (0.6.4)
  - iSCSI integration (0.7.0)
  - ARC pagecache integration (0.7.0)
  - TRIM support (0.7.0)
  - Fallocate support (0.7.0)
- Do budoucna
  - O_DIRECT, reflink

ZFS Features: CLI

- 2 příkazy obslouží všechno
  - zpool, zfs
- Sémantika použití kopíruje záměry administrátora
  - # zpool create tank mirror sda sdb
    - Vytvoří zpool s mirrorem sda, sdb
    - Vytvoří a namountuje root dataset pod /tank
  - # zfs create -o quota=60G -p tank/foo/bar
    - Vytvoří rekurzivně foo i bar datasety i mountpointy a namountuje
    - Nastaví kvótu tank/foo/bar na 60 GB
- Integrace s NFS, CIFS, iSCSI (ZoL 0.7.0)
- Nepoužívá /etc/fstab ani /etc/exports

ZFS Features: snapshots

- Snapshot
  - Per dataset, možnost rekurzivních snapshotů
  - Point in time backup
  - Constant time operation
  - Readonly, přístupné v tajném skrytém .zfs/snapshot
- Clone, Rollback, Diff
- Send/receive
  - Serializace a deserializace dat datasetu
  - Rekurzivní i inkrementální send/recv pro snadné zalohovaní
  - Použitelné pro asynchronní (geo)replikaci
  - Rsync milionů malinkých souborů je historií

ZFS Features: caching

- ARC = Adjustable Replacement Cache
  - 2 LRU seznamy – MRU, MFU
  - MRU: most recently used
  - MFU: most frequently used
  - Mnohem vyšší hitrate při variabilní zátěži
- L2ARC
  - Level 2 ARC, určeno pro SSD
  - Dostává evicted data z ARC
  - Pomalý warm-up, persistent L2ARC (0.6.4)
- Hierarchický storage model
- Nastavitelné per dataset
  - Primarycache/secondarycache: all/none/metadata

ZFS Features: zfetch

- Inteligentní prefetch mechanismus pro soubory
  - Přednačte další jeden blok, pokud se použije, načte další dva... až 256
  - Detekuje lineární čtení dopředu i dozadu
  - 8 streamů na jeden soubor

ZFS Features: ZIL

- Synchronní zápis a CoW nejdou dobře dokupy
- sync() straší ze záhrobí
- ZIL = “ZFS Intent Log”
  - Existuje vždy, write-only po většinu času
  - Oblasti v poolu dedikované pro ZIL
- Dedicated SLOG device
  - Pro SSD s vysokým write IOPS
  - Databáze: TPS++
  - sync() už není problém
  - Může být mirrored
- Nastavitelný logbias per dataset
  - Latency/throughput, zajímavé pro pole s vyšší propustností než SSD

ZFS Features: compression

- Transparentní komprese
  - Nastavitelná per dataset
  - lzjb, gzip, zle, lz4
  - lzjb/lz4: nejlepší poměr cena/výkon
  - Pro drtivou většinu nasazení se vyplatí
  - CPU výkonu je většinou dostatek
  - Šetří propustnost disků i sběrnic
  - Při zisku < 12.5% se blok uloží nekomprimovaný
  - Blok plný nul se rovnou ignoruje

ZFS Features: dedup

- Deduplikace na blokové úrovni
  - Nastavitelná per dataset
  - Volitelná verifikace, jinak stačí checksum
  - Pokud je aktivní, vynucuje SHA256 checksumy
- DDT zabírají paměť
  - Doporučuje se 1 GB RAM / 1 TB dat
  - Zpomalí systém pokud DDT přetékají
- Vyplatí se jenom pro specifické workloady
  - Pro všechny ostatní je tu komprese

ZFS Features: resilver/scrub

- Scrub
  - Rutinní check konzistence dat (bitrot!)
- Resilver
  - Obnova degradovaného poolu
- Rekurzivní průchod celým stromem
  - Rozdíl od lineárního přístupu běžných RAIDů
  - Prochazí jenom užitečná data, ne volné místo
- Scrub/resilver IO má nejnižší prioritu
  - Ano, ZFS má i svůj IO scheduler

ZFS Features: ZVOL

- Block device nad ZFS
  - # zfs create -V 20G tank/myzvol
    - => /dev/zvol/tank/myzvol
- Sparse ZVOL
- iSCSI integrace v CLI (ZoL 0.7.0)
- Ideální pro fullvirt VM

Nasazení ZFS ve vpsFree.cz

- Komunitní VPS hosting
  - Kontejnerová virtualizace, kontejner je adresář
    - Cílem je efektivně využívat společné HW prostředky
    - Spousty malých souborů k zálohování
    - Velmi variabilní zátěž, nestačí naivní LRU cache
  - Jak projekt rostl, rostl i hardware
    - První stroj – 8 GB RAM, ani ne 15 kontejnerů
    - Dnes nejdeme pod 256 GB
      - Cílem je ~100 kontejnerů per server
      - Za žádnou cenu ale nechceme obětovat výkon

vpsFree.cz

OpenZFS

Nasazení ZFS ve vpsFree.cz

- Původní řešení
  - MD-RAID10, LVM, flashcache, Ext4
  - Journal bottleneckem
  - Random writes zabíjí IO
  - Stačilo pár MySQL serverů na jednom node...
  - Vzájemné vylévání si cache...
  - Zálohy trvaly věčnost...
  - FSCK po pádu systému na náladě nepřidá...
  - vzquota...

vpsFree.cz

OpenZFS

Nasazení ZFS ve vpsFree.cz

- Přechod na ZFS
  - Červenec 2013, ZoL 0.6.2 – první testovací node
  - Porodní bolesti nemalé (správa paměti pod Linuxem)
  - Okamžitě viditelný rozdíl
    - ARC téměř 100% hitrate, CoW – serializace random writes
    - Zkrácen čas zálohování na polovinu
  - Start systému s 90ti VPS do 10ti minut
  - Červenec 2014: ZFS only
  - Zbývá dodělat zálohování přes send/recv
  - Zpřístupnění vlastností ZFS do kontejneru

vpsFree.cz

OpenZFS

Nasazení ZFS ve vpsFree.cz

```
[root@node1.brq.vpsfree.cz]
 ~ # zpool status
  pool: vz
 state: ONLINE
  scan: scrub repaired 0 in 0h5m with 0 errors on Wed Sep  3 19:47:07 2014
config:

    NAME        STATE     READ WRITE CKSUM
    vz          ONLINE       0     0     0
      mirror-0  ONLINE       0     0     0
        sdc     ONLINE       0     0     0
        sdd     ONLINE       0     0     0
      mirror-1  ONLINE       0     0     0
        sda     ONLINE       0     0     0
        sdb     ONLINE       0     0     0
      mirror-2  ONLINE       0     0     0
        sdg     ONLINE       0     0     0
        sdh     ONLINE       0     0     0
      mirror-3  ONLINE       0     0     0
        sdi     ONLINE       0     0     0
        sdj     ONLINE       0     0     0
    logs
      mirror-4  ONLINE       0     0     0
        sde3    ONLINE       0     0     0
        sdf3    ONLINE       0     0     0
    cache
      sde5      ONLINE       0     0     0
      sdf5      ONLINE       0     0     0

errors: No known data errors
```

Nasazení ZFS ve vpsFree.cz

```
[root@node1.brq.vpsfree.cz]
 ~ # zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
vz                  172G  6.97T   803M  /vz
vz/dump              30K  6.97T    30K  /vz/dump
vz/private          171G  6.97T    50K  /vz/private
vz/private/1009    31.9G  28.1G  31.9G  /vz/private/1009
vz/private/1071     891M  6.97T   891M  /vz/private/1071
vz/private/1151    35.9G  24.1G  35.9G  /vz/private/1151
vz/private/1162     493M  6.97T   493M  /vz/private/1162
vz/private/1320    18.4G  41.6G  18.4G  /vz/private/1320
vz/private/1322    21.0G  39.0G  21.0G  /vz/private/1322
vz/private/1846    49.1G  10.9G  49.1G  /vz/private/1846
vz/private/2527    8.18G  6.97T  8.18G  /vz/private/2527
vz/private/2866     355M  6.97T   355M  /vz/private/2866
vz/private/3042      48K  60.0G    30K  /vz/private/3042
vz/private/3453    3.20G  56.8G  3.20G  /vz/private/3453
vz/private/3477     211M  59.8G   211M  /vz/private/3477
vz/private/3479    1.01G  59.0G  1.01G  /vz/private/3479
vz/root              53K  6.97T    53K  /vz/root
vz/template          30K  6.97T    30K  /vz/template
```

vpsFree.cz

OpenZFS

Nasazení ZFS ve vpsFree.cz

```
snajpa@snajpaws:~$ for n in $VPSFREE_NODES; do echo -en "$n\t"; ssh $n zfs get -Ho
value compressratio vz; done

node1.brq.vpsfree.cz  1.29x
node2.brq.vpsfree.cz  1.53x
node1.prg.vpsfree.cz  1.43x
node2.prg.vpsfree.cz  1.34x
node5.prg.vpsfree.cz  1.25x
node6.prg.vpsfree.cz  1.56x
node7.prg.vpsfree.cz  1.36x
node8.prg.vpsfree.cz  1.28x
node9.prg.vpsfree.cz  1.41x
node10.prg.vpsfree.cz 1.67x
node1.pgnd.vpsfree.cz 1.39x


[root@backuper.prg.vpsfree.cz]
 ~ # zpool list storage
NAME      SIZE  ALLOC   FREE  EXPANDSZ    CAP  DEDUP  HEALTH  ALTROOT
storage  89.4T  62.2T  27.2T         -    69%  1.00x  ONLINE  -
```

vpsFree.cz

OpenZFS

Workshop

- Koho ZFS zaujalo, má možnost si ho rovnou osahat
- Best practices
- 15:00 učebna 348

vpsFree.cz

OpenZFS

Shrnutí

- “ZFS: The Last Word in Filesystems!”
- “Nejlepší filesystem pro servery roku 2014+”
- Linuxový port je použitelný a stabilní
  - Máte mašinu, kde ext4 nestačí? Chcete ZFS.

vpsFree.cz

OpenZFS

Shrnutí

- “ZFS: The Last Word in Filesystems!”
- “Nejlepší filesystem pro servery roku 2014+”
- Linuxový port je použitelný a stabilní
  - Máte mašinu, kde ext4 nestačí? Chcete ZFS.

- Otázky?
  - snajpa@snajpa.net

vpsFree.cz

OpenZFS