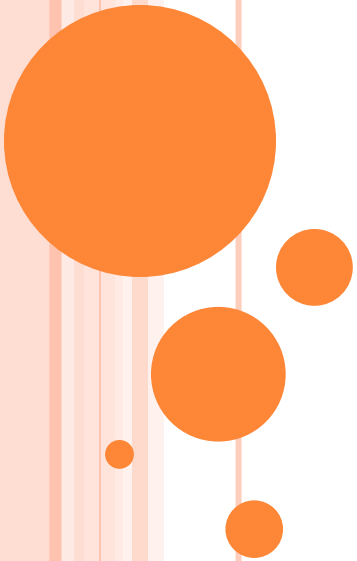


Moderní souborové systémy - ZFS

Richard Janča

Moderní souborové systémy - ZFS

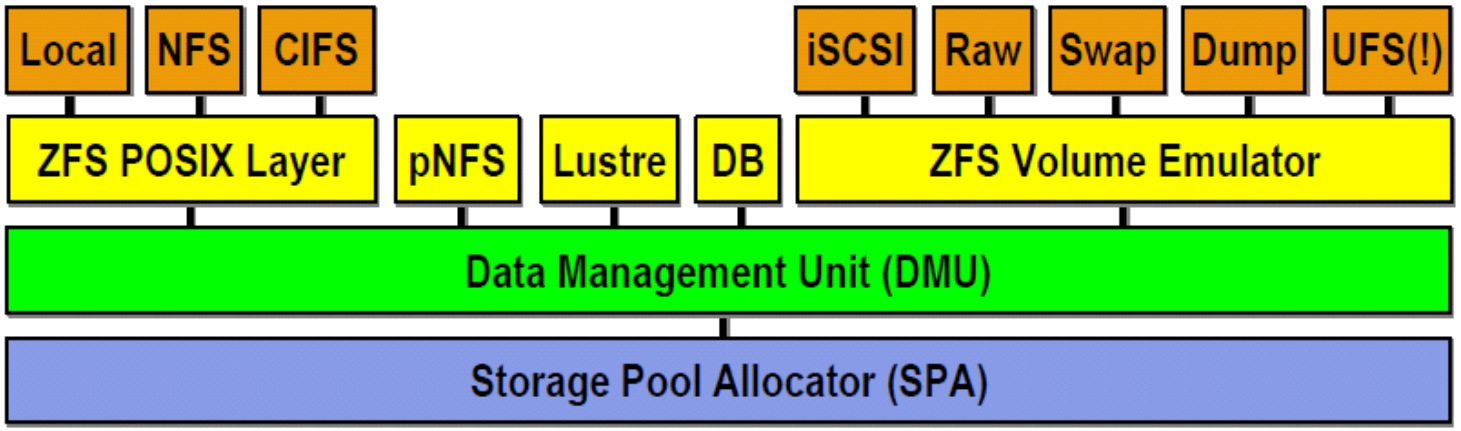
- ZFS- *Zettabyte File Systém*
- *128 bitový souborový systém*
- *Původně pouze pro Solaris*
- *Dnes již CDDL licence*
  - *FreeBSD*
  - *Solaris*
  - *Příprava pro Linux-problémy s CDDL licencí*
  - Mac OS X 10.6 Snow Leopard +

Vlastnosti

- ZFS umí uložit až 16 exabitů dat
- v jednom adresáři až 2^{56} souborů
- stráži integritu svých dat
- Proměnlivá velikost bloků
- Transparentní komprese
- umí tzv. self-healing dat (automaticky opravuje poškozená data)
- Lehké ovládání – zfs, zpool
- Využívá „snapshoty“ pro obnovu disků
  - na disku je teoreticky možné uchovat až 2 na 64 snapshotů
- chybovost dat je asi 0,01%

ZFS - Struktura

- SPA (Storage Pool Allocator)
- DSL (Dataset and Snapshot Layer)
- DMU (Data Management Layer)
- ZAP (ZFS Attribute Processor)
- ZPL (ZFS Posix layer)
- ZIL (ZFS Intent Log)
- ZVOL (ZFS Volume).

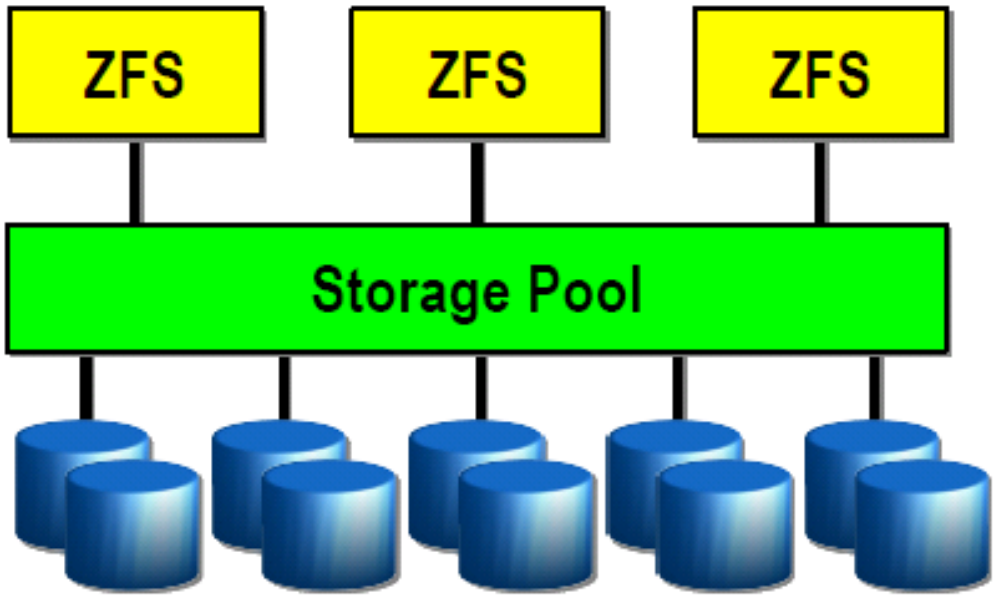


SPA – Storage Pool Alocator

- SPA zvládá všechny alokace bloků a IO.
- „Převádí“ všechna zařízení na vdev a poskytuje virtuálně adresované bloky pro DMU.
- SPA je tedy rozhraním pro alokování a uvolňování těchto virtuálně adresovaných bloků.
- Vdev
  - Celé disky, diskové oddíly, soubory

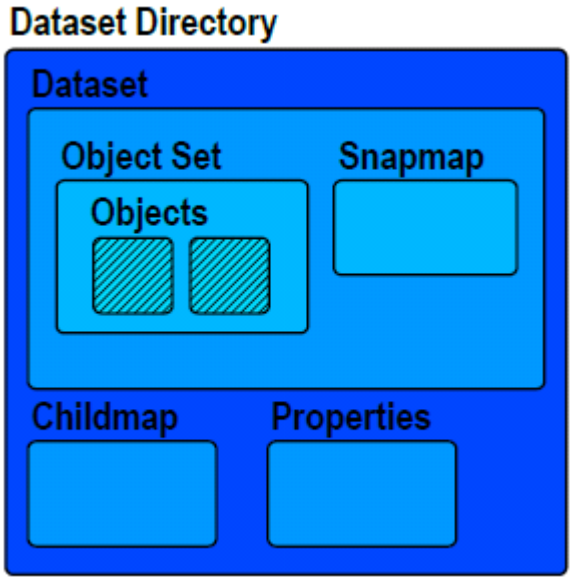
ZFS – Struktura

- ZFS – POOL
  - Nemá partition
  - Zvětšování/zmenšování – automaticky
  - Všechny uložiště přístupné



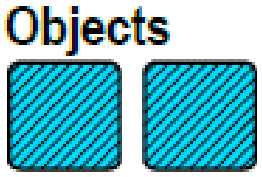
DSL - Dataset & Snapshot Layer

- Poskytuje mechanismus pro řízení „vztahů“ mezi:
  - filesystem, clone, snapshot, volume
- „vztahy“ se řeší nad object set jako Dataset a Dataset Directory



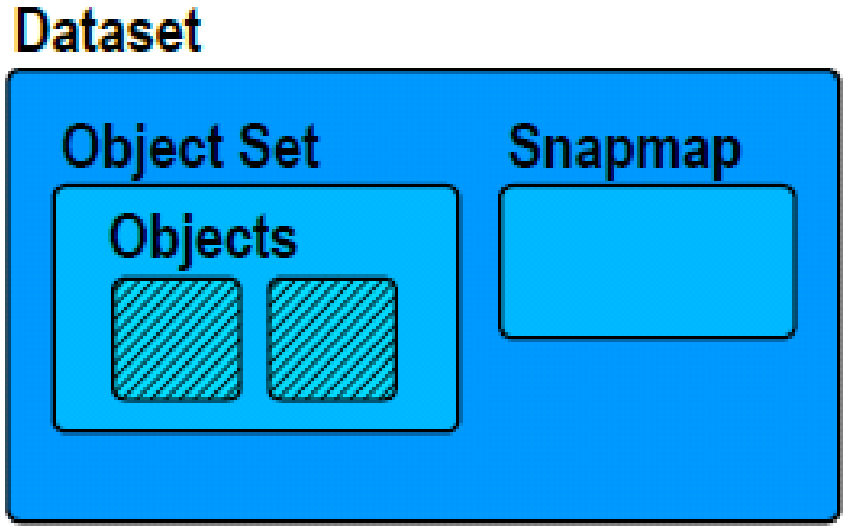
Objekty v DSL

- V podstatě vše v ZFS je object
- Dnode popisuje a organizuje sadu bloků které tvoří objekt
- Znode je representace souborů/adresářů na úrovni ZFS Posix Layer (ZPL)
- dnode+znode ≡ UFS i-node



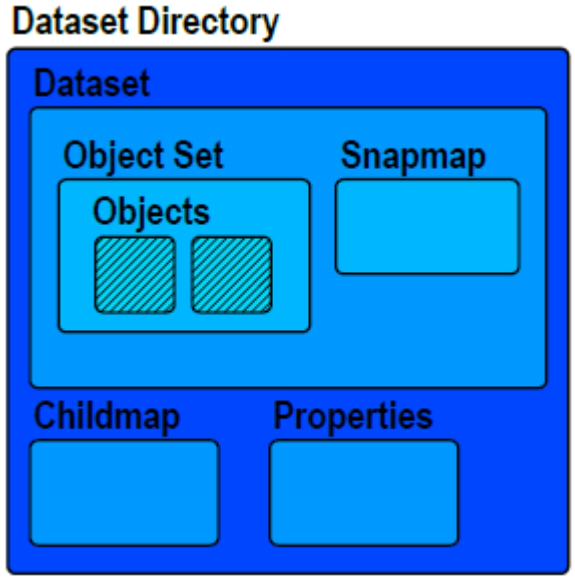
Datasets

- Zapouzdřuje objset a poskytuje:
  - Využití prostoru
  - Vztahy mezi snapshoty



Dataset Directory

- Spojuje Datasety
- Řeší vlastnosti například:
  - quotas, reservations, compression
- „vztahy“ Dataset
  - Rodič musí existovat aby existovalo dítě



DMU – Data Management Unit

- DMU – Data management unit
  - Dnode – definuje objekty
    - Typ, velikost
    - Seznam block-pointeru
  - DMU object set (dataset)
    - Obsahuje pole dnode
    - Provádí :
      - Snapshots, komprese, šifrování, end-to-end data integrita
  - DMU provádí transakce v rámci celého poolu
    - Soubory, bloky, objekty, síť

DMU – Data Management Unit

- DMU převádí instrukce od ZPL do transakčních příkazů.
- Lepší než posílat po jednom požadavek k zápisu, ZFS to řeší pomocí objektových transakcí které mohou být optimalizovány ještě před jakoukoliv diskovou aktivitou.
- Jakmile je toto splněno transakce jsou předány SPA a to již naplánuje a provede samotné I/O.
- Spolu s COW a checksum pro každý blok to značí nepotřebu žurnálová ní.

ZFS Attribute Processor

- ZAP – ZFS Attribute Processor
- ZAP je modul, který je umístěn v horní části DMU a operuje s tzv. ZAP objekty.
- ZAP objekty se používají k:
  - ukládání vlastností datové sady (dataset)
  - navigace filesystemovými objekty
  - ukládání vlastností poolu a další.

ZAP

- microZAP
  - Jeden blok(do 128k)
  - Jednoduché atributy (64 bit)
  - Omezená délka jména (50 bytes)
  - Odlehčené FatZAP
  - Používá se u omezeného počtu atributů.
- FatZAP
  - Spíše pro ZAP objekty s velkým množstvím atributů

ZFS Posix Layer / Volume

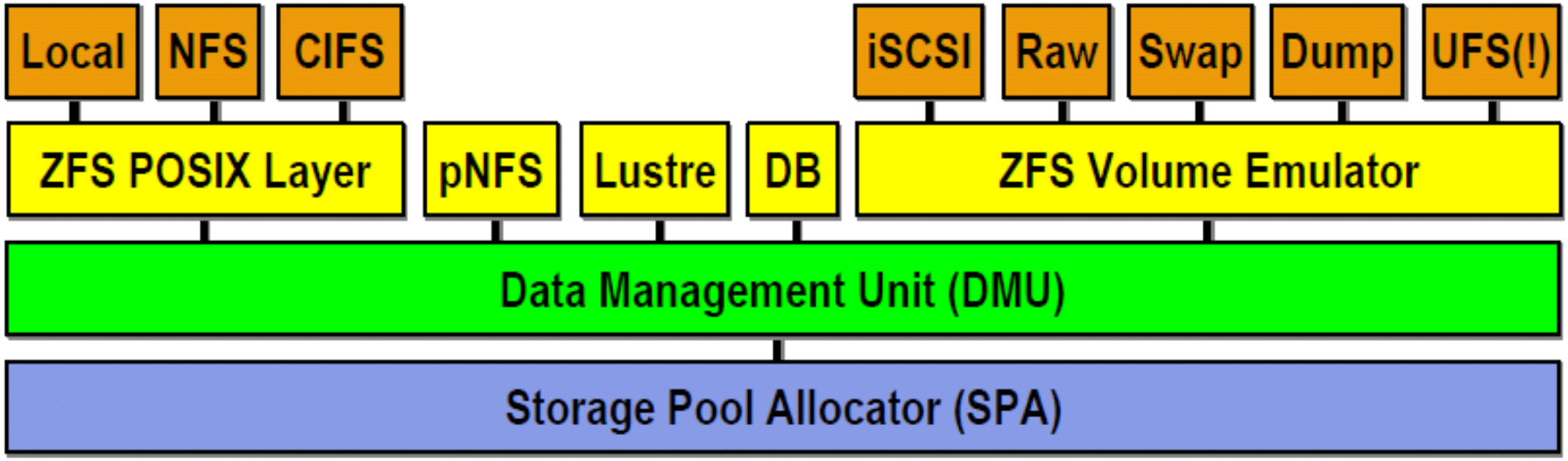
- ZFS Posix vrstva
  - Implementuje POSIX souborový system
  - Adresáře jsou ZAP objekty
  - Soubory DMU objekty

ZIL – ZFS Intent Log

- Zil ukládá záznamy transakcí systémových volání která jakkoliv mění souborový systém.
- Záznamy mají dostatek informací k tomu aby se daly zopakovat.
- Záznamy jsou uloženy v paměti kde čekají až DMU prohlásí že pool je stabilní a odstraní je.
- Transakce jsou ještě zapsány do stable logu a když dojde ke kernel panic či výpadku proudu všechny transakce jsou natrvalo zde.

ZVOL – ZFS Volume

- Mechanismus pro tvorbu logických svazků.
- Svazky jsou exportovány jako bloková zařízení a můžou být použity jako jakékoliv jiné blok. Zařízení.
  - iSCSI
  - UFS
  - Swap



Integrita Dat

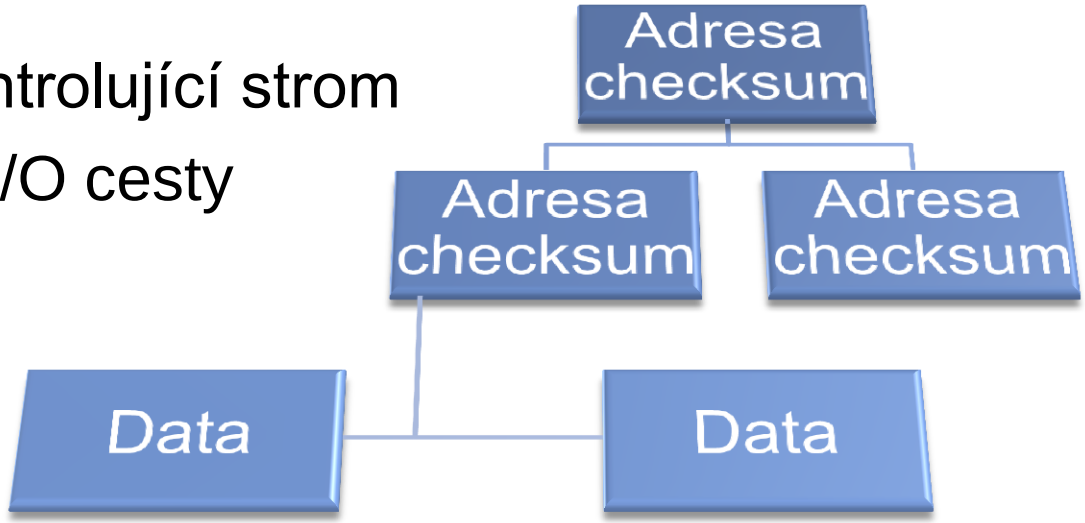
- Copy-On-Write
- Autentifikace dat
- Self-Healing
- Raid-Z

Copy on Write

- **Copy-On-Write** (COW), který stráži konzistenci dát.
- používá tzv. **block pointers,** teda ukazatele na bloky dát v rámci svého souborového systému.
- Jakmile je kopírování bloku dat úspěšně dokončení tak block-pointers se změní na bloky
- To zaručí konzistenci dat například při výpadku energie.

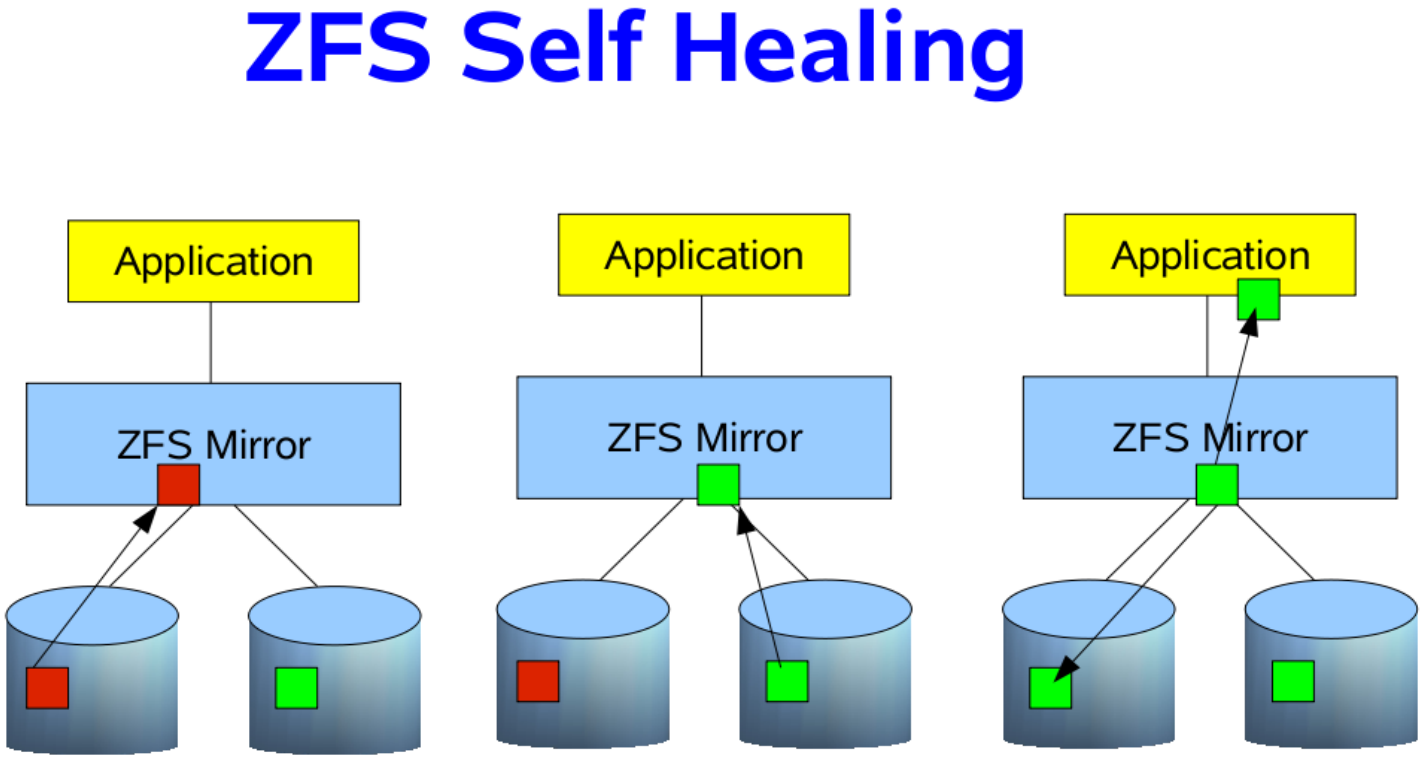
Autentifikace dat

- Autentifikace se provádí pomocí souhrnného součtu
- Checksum se ukládá v nadřazených blocích
  - Lepší protože při poškození dat není poškozen checksum
- Tvoří sebe-kontrolující strom
- Kontrola celé I/O cesty



ZFS SELF-HEALING

- ZFS **self-healing** je funkce, která při mirrorech dvou a více disků automaticky opravuje poškozená data. Kontrola probíhá pomocí checksum.



RAID-Z

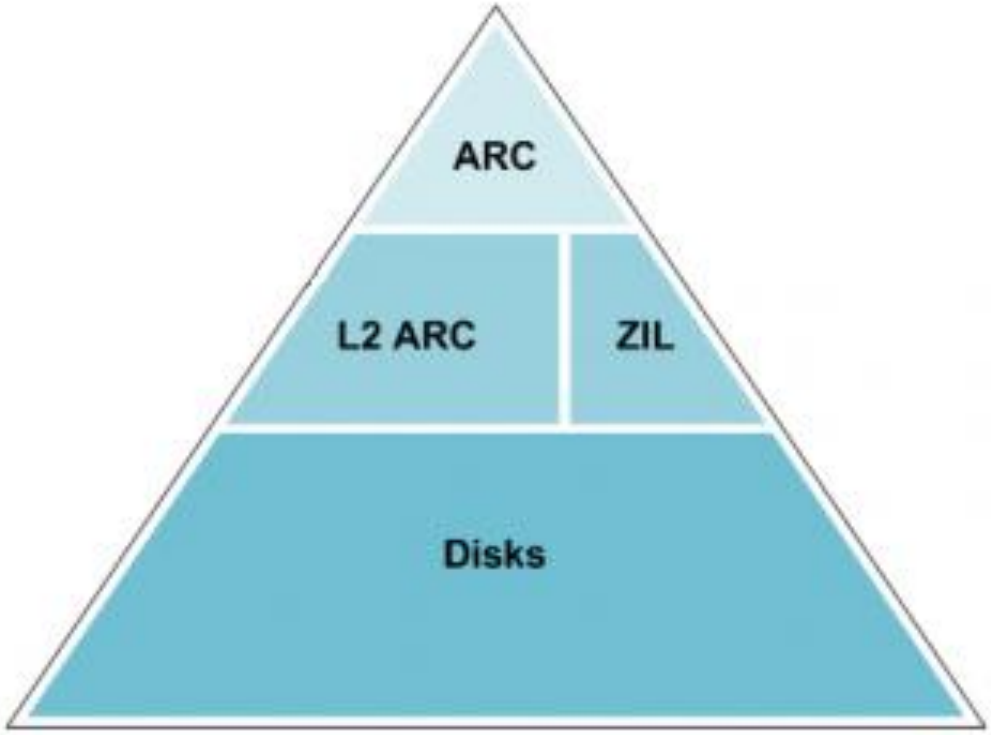
- Obdobný jako RAID-5
- Vyvarovává se chyby RAID-5 „write hole“ pomocí COW
- Používá dynamickou velikost bloku.
- RAID-Z je také rychlejší protože nikdy nedělá read-modify-write.

ARC - Adaptive Replacement Cache

- je rychlá hlavní vyrovnávací paměť pro operace s ZFS.
- Nachází se ve fyzické paměti.
  - Dynamicky mění svojí velikost vzhledem k dostupné paměti.
  - Když nemá systém dost paměti ZFS nějakou uvolní.
  - ARC zabírá cca ¾ celkové paměti
- **ZFS intent log** složí pro zaznamenání všech i zatím zamýšlených operací na discích.
  - Při Kernel panic či výpadku proudu existuje záznam co se dělo
  - Zabraňuje nekonzistenci

ARC

- ZIL – ZFS Intent Log
  - Zapisuje i zamýšlené diskové operace



New Model with ZFS

Záloha systému - Snapshots

- Read-only kopie file-systemu v konkrétním čase
- Může být vytvořen skoro okamžitě
- První Snapshot nemá skoro žádnou velikost (odkazuje na data ve FS)
- Po změně a vytvoření snapshotu č.2 má č.2 minimální velikost ale č.1 má velikost dat před změnou
- Každý snapshot je rozdílem předcházejícího

Záloha systemu - Clones

- ZFS **clone** je read/write kopie filesystemu vytvořená z snapshotu.
- Zabírá minimum místa na disku.
- **# zfs clone tank/solaris@monday tank/ws/lori/fix**

ZFS – PŘÍKAZY

- Vše se zjednodušuje na dva příkazy: zpool a zfs.
  - # zpool create prvni c1t2d0
  - # zfs create prvni/filmy
  - # zfs set mountpoint=/pub/video prvni/filmy
  - # zfs snapshot prvni/filmy@filmy_zaloha

```
# zfs list
NAME                    USED  AVAIL  REFER  MOUNTPOINT
pool                   21.7G  13.0G    18K  none
pool/root              6.72G  13.0G  4.99G  /
pool/root/tmp           165M  13.0G   165M  /tmp
pool/root/var           192M  13.0G   192M  /var
pool/samba             10.0G  10.0G  21.5K  /samba
....
```

```
# zfs list -t snapshot
NAME USED AVAIL REFER MOUNTPOINT
pool/test@zaloha 169M - 2.72G /test
pool/test@20101213 140M - 2.79G /test
....
```

```
# zfs list -r pool/test
NAME USED AVAIL REFER MOUNTPOINT
pool/test 277K 16.5G 277K /test
pool/test@zaloha 0 - 277K -
pool/test@zaloha2 0 - 5GK -
....
```

```
# zfs list -t all
NAME USED AVAIL REFER
pool 26.1G 8.58G 18K none
pool/root 6.11G 8.58G 5.08G /
pool/test@zaloha 169M -
....
```

KOMPRESE

```
# zfs set compression=gzip-9 pool/test
# zfs get -r compression pool/test
NAME PROPERTY VALUE SOURCE
pool/test compression gzip-9 local
```

```
# zfs get -r compression pool/test
NAME PROPERTY VALUE SOURCE
pool/test compression gzip-9 local
pool/test/A1 compression gzip-9 inherited from pool/test
pool/test/A2 compression gzip-9 inherited from pool/test
```

HOLD

```
# zfs hold keep pool/root@prvni_instalace
# zfs destroy pool/root@prvni_instalace

Cannot destroy ´pool/root@prvni_instalace´: dataset is busy
```