Tuning XFS

aneb

Cesta do pravěku

ivan.bibr@altworx.com

Něco z pravěku

- Vznik 1993 – SGI/IRIX
- GPL zdrojáky roce 1999
- Portace do kernelu 2001, od 2.6 v mainline
- Od ~2004 běžná součást distribucí
- CentOS/RHEL: default od v7, tzn. 2014
- Verzování XFS – kdysi, dnes kontinuální vývoj
- Pozor na staré informace na netu!
- Aktuální informace: https://xfs.wiki.kernel.org

ivan.bibr@altworx.com

Důležité vlastnosti

- Žurnálování – jeden z prvních v ml kernelu
- Paralelizace – výkon/propustnost roste s jádry
- ACL, kvóty (user/project) – někdy nutné
- Podpora velkých volumes, 8EB max na 64bit
  - Online resize FS
- Delayed allocation (fragmentace)
  - Online defragmentace v případě potřeby
- Nízké nároky na hardware, **stabilita!**

ivan.bibr@altworx.com

Nevýhody a chybějící vlastnosti

- Stará škola – just filesystem
- Snapshoty pouze přes LVM (pomalé)
  - Zálohy souborově, dle volume size
- Data nemají kontrolní součty (silent corruption)
- Neumí online kompresi
- Neumí deduplikaci (ani soubory ani bloky)
- Pomalá práce s metadaty: neplatí od ~2012
- Neumí copy on write: nějaká podpora od 2019?

ivan.bibr@altworx.com

Proč XFS a proč jeho tuning

- Geoinformatika:
  - Moc (specifických) dat – stovky TB až PB
  - Velké množství souborů - desítky/stovky mil.
  - Interní procesy = nemožný cloud
- Vlastní storage (hw/sw/backup) + 10GBe
- Velké nároky, malé peníze :)
- Logická volba: XFS (just filesystem)
  - Základní princip: KISS

ivan.bibr@altworx.com

Kdy tunit a proč, úskalí tuningu

- Nemáte problém = OK, máte problém = tuning
- Zlepšujte co je potřeba nebo dává smysl
  - Pozor na overtuning, osobní zkušenost :)
- Co lze získat:
  - Lepší využití prostředků (hardwaru)
  - Lepší datová propustnost a odezva
- Neexistuje univerzální tuning!
- Většinou nelze čekat násobné zvýšení výkonu!

ivan.bibr@altworx.com

Základní tuning

- Žádný tuning, taky tuning :).
  - Překvapivě dobré (odolné/bezpečné) řešení!
- Parametry do `/etc/fstab`:
  - `noatime,inode64,logbufs=8,logbsize=256k,noquota`
- Nejjednodušší tuning
  - Lepší hardware: dobrý poměr cena/výkon
  - SSD: overprovisioning (volné místo)
  - Separátní disky na data + mkfs na celý disk, případně dedikované servery dle usecase

ivan.bibr@altworx.com

Pokročilý tuning

- Potřebné znalosti (a zároveň okruhy tuningu):
  - Struktura XFS – data/metadata
  - Verze xfsutils/mkfs.xfs, on-disk-formatu
  - Vlastnosti hardwaru, tzn. disků/polí, keší ...
  - Způsob práce kernelu (VFS) + parametry
- Je nutné znát procesy, které budete ovlivňovat
  - Mnoho možností, vytřídit marginálie!
  - Příliš moho věcí lze pokazit!

ivan.bibr@altworx.com

Dobrá pravidla

- Dobrá volba: 80/20
- Co nejméně zásahů – předvídatelné výsledky
  - Spousta nastavení jde "proti sobě":
  - Větší bloky = větší propustnost ale i latence
- Nutné testy a zvážení přínosů:
  - Testy jsou ošemetné, pozor na podmínky!
  - Syntetické benchmarky: nevypovídající
  - Pečlivá systematická práce stojí moc času!

ivan.bibr@altworx.com

Hardware RAID a stripe size

- Více disků – rozložený zápis
- Redundance (1) a kontrolní součty (5/6) dat
- Atomická jednotka dat: stripe (XFS = chunk)
  - Ekvivalent bloku na fs (a důsledky).
  - Pozor na dezinterpretace!

- **Stripe Size**
  This parameter sets the size of segment written to each disk in a RAID 0, 1, 1E, 10, 5, 6, 50 or 60 logical drive. You can set the stripe size to 4 KB, 8 KB, 16 KB, 32 KB, 64 KB, 128 KB, 256KB, 512KB, or 1024KB.

Warning:

ivan.bibr@altworx.com

Hardware RAID – příklad

- Ukázkový příklad: 12 disků, RAID 6 + spare
  - Spare disk se nepoužívá (-1)
  - RAID 6: 2 checksumy (-2)
  - Celkem 9 datových disků
  - Stripe size 1M
- Parametry při tvorbě FS:
  - `mkfs.xfs … -d su=1024k,sw=9 …`
- Náš HW: až 300+ TB formátované (více disků)

ivan.bibr@altworx.com

Zvětšení hardware RAIDu

- Nevýhoda su/sw: pevná hodnota při mkfs
- Zvětšení raidu: přestane platit „zarovnání“ dat
- Lze řešit úpravou `/etc/fstab`
- „Archaické“ parametry sunit & swidth
  - **POZOR**, jsou v 512b blocích, pravěk!
- Náš příklad: zapojení spare do raidu
  - Místo 9 datových disků je 10
  - Fstab změna: `sunit=2048,swidth=20480`

Metadata na hardware RAIDu

- „Kazí“ propustnost, r/w seek mimo data (plotny)
  - Žurnál (zápis) a inody (čtení + zápis)
- Odložení žurnálu na SSD + zvětšení + lazy(?):
  - `mkfs.xfs … -l logdev=/dev/sdb,size=2020m\, lazy-count=1 …`
- Parametry do fstabu:
  - Přidat: `logdev=/dev/sdb`
  - Používat raději `/dev/disk-by-id/...`

ivan.bibr@altworx.com

Metadata na hardware RAIDu 2

- Inody – veškeré informace o souborech/datech
- Lze využít cache kernelu: vfs_cache_pressure
  - Tendence nechávat inody v RAM (seek)
  - `echo 0 > /proc/sys/vm/vfs_cache_pressure`
- Dávejte **VELKÝ POZOR!**

```
[redacted] ~]# free -h
              total        used        free      shared  buff/cache   available
Mem:           251G        119G        922M         57M        130G        128G
Swap:           31G         15G         15G
[redacted] ~]# echo 2 > /proc/sys/vm/drop_caches
[redacted] ~]# free -h
              total        used        free      shared  buff/cache   available
Mem:           251G        120G         46G         57M         84G        129G
```


ivan.bibr@altworx.com

Velké množství souborů

- Náročné na interní struktury FS (velké/složité)
- Lze zlepšit pomocí „allocations group“ (AG):
  - Separátní inodes, škálování/paralelizace
- Default 4 je málo, klidně lze i CPUx2
- Parametr pro mkfs:
  - `mkfs.xfs … -d agcount=32 …`
- Pomůže, ale není samospásné + má limity.

ivan.bibr@altworx.com

Velké množství souborů 2

- HW: výkonné SSD (NVMe), není co řešit
- I tak to má stále limity > 10M souborů
  - Pomalé operace: tvorba, kopírování, mazání
- Tip: mazání pomocí `find … | xargs -Px`
- Tip: lze vytvořit „disk-images“
  - Normální „velké“ soubory.
  - Rychlé přesouvání/ a mazání datových sad.
  - Pro náš usecase téměř ideální :)

ivan.bibr@altworx.com

Další možný tuning

- Větší inody: `mkfs.xfs -isize=512 ...`
- Lepší metadata: `-m crc=1,finobt=1`
  - Kontrolní součty pro metadata
  - Separátní strom pro free inodes
- Inline data – od 2018, lze „opatrně“ použít :)
- Obsolete: zarovnání oddílů (dělá parted).
- Spekulace: SSD a erase block size?
  - Vázáno na konkrétní fyzický disk, testy?

ivan.bibr@altworx.com

Usecase 1: velká datová pole

- Fotky (velké soubory), pouze v milionech
- Nutná velká propustnost (saturace 10GBe)
- `mkfs.xfs -isize=512 -m crc=1, finobt=1 -l logdev=/dev/sdb, lazy-count=1,size=2020m -d su=1024k, sw=30 /dev/sda`
- Parametry pro fstab: `logdev=/dev/sdb,noatime,inode64 logbufs=8,logbsize=256k,noquota`

ivan.bibr@altworx.com

Usecase 2: mapové dlaždice

- Miliony (10/100+) obrázků pro mapové služby
- Jednoúčelové servery se serverovými SSD
- Odděleno podle datových sad:
  - Sada = soubor s XFS na disku (LVM ne)
  - Pomalé generování (z podstaty, nevadí)
  - Rychlý přenos po síti
  - Rychlá výměna dat, rychlé smazání
  - Rychlost výdeje dat naprosto OK (SSD)

ivan.bibr@altworx.com

Hardware

- Servery Supermicro, vlastní návrhy dle užití
- Fileservery: jedno CPU, RAM podle potřeby
- Hardwarový RAID
  - Různý level podle potřeby: 1/5/6/60
  - Více jader, velká RAM na řadiči
- Serverové disky – rotační i SSD (workload!)
- Backup totéž, software: rsnapshot
- Velmi dobrá cena za uložení TB ročně.

ivan.bibr@altworx.com

Končíme, vypršel čas!

ŽÁDNÉ DOTAZY? VÝBORNĚ!

ivan.bibr@altworx.com