Skrivet av Erik Liljencrantz
2017-02-02 21:52 - Senast uppdaterad 2017-02-03 12:58

Stora roterande diskar kan kompletteras med en mindre SSD som cache på en server. Här testar vi hur det fungerar.

Vi lagrar allt fler och större filer på servrar och nätverksdiskar (NAS). För dessa lagringsplatser är roterande magnetiska diskar (hårddisk) det normala valet då kostnaden per gigabyte är såpass mycket högre för elektroniska diskar (SSD). Men de senare är väldigt mycket snabbare. Genom att kombinera dessa disktyper, kan man då få det bästa av två världar?

Under operativsystemet Linux som vi kör på våra servrar finns sedan några få år möjligheten att till en logisk volym (LVM) som kan vara speglad (RAID1) med mer för felsäkerhet även ansluta en cache-pool i form av en snabbare enhet.

För att testa riggas en utrangerad server med två SATA-anslutna diskar som /dev/sda och /dev/sdb. Den senaste versionen av CentOS 7.2 installeras på en LVM-volym med spegling (RAID1). Alla data skrivs därmed på bägge diskarna och vid läsning kan hälften av data läsas från vardera disken parallellt vilket därmed ger upp till dubbla läshastigheten.

Så läggs en SSD till som /dev/sdc. Kommandona är översiktligt:

+ skapa partition och ange typ 0x8e Linux LVM med:
fdisk /dev/sdc

+ skapa fysisk lagringsenhet med:
pvcreate /dev/sdc1

+ utöka befintlig volymgrupp med den nya partitionen (döpt till cl vid installationen av Linux):
vgextend cl /dev/sdc1

+ skapa meta- och data-utrymme för cachen på /dev/sdc1:
lvcreate -L 1G -n cachemeta cl /dev/sdc1 lvcreate -L 100G -n cachedata cl /dev/sdc1 där egentligen meta kan vara 1/1000-del i storlek mot data men här blev 1/100-del.

Med kommandot lvscan ser man de tidigare volymerna samt de två nya men ännu inaktiva cache-delarna:
ACTIVE '/dev/cl/root' [200,00 GiB] inherit ACTIVE '/dev/cl/home' [600,00 GiB] inherit ACTIVE '/dev/cl/swap' [8,00 GiB] inherit inactive '/dev/cl/cachemeta' [1,00 GiB] inherit inactive '/dev/cl/cachedata' [100,00 GiB] inherit

+ nu skapas själva cachepoolen genom att slå ihop meta och data: lvconvert --type cache-pool --poolmetadata cl/cachemeta cl/cachedata

+ till sist kopplas cachedata in som cache för /home-partitionen: lvconvert --type cache --cachepool cl/cachedata cl/home

Så här långt inga felmeddelanden. För att se hur diskarna arbetar körs: iostat -x 5 sda sdb sdc

i ett fönster vilket var femte sekund visar hur mycket diskarna läst, skrivit och nyttjandegrad.

Som test kopieras filträden /usr/bin och /usr/lib om ca 832 mb till sex kopior under /home på totalt drygt 5 000 mb. Man ser i utmatningen från iostat att data läses på hårddiskarna och skrivs både på SSD och hårddiskar. Kopieringen går ganska långsamt då hårddiskarna måste växla mellan läs och skriv med nära 100% nyttjandegrad medan SSD:n som bara skriver lite data ligger på neråt 10% nyttjande. För denna operation har vi inte tjänat något, men förhoppningsvis inte förlorat så mycket - men kopierade data finns även cache:ade på SSD:n när kommandot kört klart.

Nästa steg är att testa upprepade läsningar av de kopierade filerna med time grep -r erik . och här visar iostat att SSD:n leverar data i upp till 120 mb/s. Totalt tar genomsökningen av drygt 5 000 mb ca 58 sekunder (ett snitt lite under 100 mb/s).

Samma genomsökning av de ursprungliga filträden under /usr/bin och /usr/lib om 832 mb tar ca 30 sekunder, men sexdubblar man den tiden får man som jämförelse 180 sekunder - tre gånger långsammare än /home med SSD-cachen!

Genom SSD-cachen går läsningen av en stor datamängd ca 3 ggr snabbare med LVM:s cache-funktion på en SSD inkopplad!

Dock: vid upprepad genomläsning av /usr/bin och /usr/lib så dessa 832 mb finns i diskbuffert i RAM-minne går det på endast 0,5 sekunder - som sexdubblad motsvarar 3 sekunder eller ca 10 ggr snabbare än SSD-läsning! Här anar man att SSD-cachning må kunna ge lite bättre fart på återkommande läsningar, men en enklare lösning kan vara mer RAM-minne som kan användas till blixtsnabba diskbuffertar istället.

Stärkt av den inledande framgången skapas två nya volymer för meta och data så även root-partitionen kan cache:as. Motsvarande som ovan, men under namnet rcachemeta och rcachedata, så anlutning till root-partitionen med: lvconvert --type cache --cachepool cl/rcachedata cl/root

Nu infinner sig dock problem vid boot. Drivrutinen för dm-cache ingår inte i den initrd-fil som skapades vid installationen av Linux. Efter omstart i rescue-läget - som verkar innehålla ALLA drivrutiner och till slut kör i runlevel 5 visar kommandot: lsinitrd /boot/initramfs-3.10.0-514.6.1.el7.x86_64.img | grep drivers/md att katalogen drivers/md saknar dm-cache och några drivrutiner till. Med dracut byggs en ny intird för den senaste kernel-versionen: dracut -f /boot/initramfs-3.10.0-514.6.1.el7.x86_64.img 3.10.0-514.6.1.el7.x86_64 och kör man lsinitrd-kommandot på denna fil visas rader som: ...usr/lib/modules/3.10.0-514.6.1.el7.x86_64/kernel/drivers/md/dm-cache.ko ...usr/lib/modules/3.10.0-514.6.1.el7.x86_64/kernel/drivers/md/persistent-data/dm-persistent-data.ko vilket ser mer rätt ut.

Omstart med reboot så startar servern med senaste kerneln och cachepool inkopplad även för root-partitionen!

Ytterligare tester bekräftar slutsatserna ovan men vissa operationer verkar tyvärr gå klart mycket långsammare! Att kopiera en stor fil på 3 500 mb från root-partitionen (med SSD-cache) till home-partitionen (också med SSD-cache) når en effektiv hastighet på ca 20 mb/s vilket är riktigt långsamt. Nu är cachen inställd så skrivningar görs direkt även på de underliggande mekaniska diskarna (write-through). Det finns möjlighet att ändra så skrivningar först görs enbart på SSD (i cachen) och senare skrivs även på hårddisk (write-back) vilket skulle öka prestanda väsentligt, men det ökar också risken för dataförlust vid ett SSD-haveri om data ännu inte hunnit skrivas även på hårddiskarna.

Vår slutsats blir dock att svaret på om det är värt att koppla in SSD för cache av hårddiskar beror på situation och ekonomi:

- skall det vara riktigt snabbt, använd enbart SSD (backa upp på hårddiskar) - till högre kostnad per gigabyte
- är det stora datamängder och mest läsningar kan SSD-cache i LVM definitivt höja prestanda på ett kostnadseffektivt sätt
- ett alternativ kan vara att öka mängden RAM-minne så mer av diskarna cachas i många gånger snabbare RAM-minne

Hoppas denna tekniska artikel om LVM och SSD-cache intresserade någon. **Vi hjälper Dig gärna med enklare lagringsfrågor också :-)**

Fotnot: då det är en äldre utrangerad server med äldre SATA-anslutningar når inte SSD:n den hastighet den egentligen kan göra - med modern SATA3 i en ny server skall SSD kunna ge drygt 500 mb/s mot hårddiskens ca 100-150 mb/s. Så finns M.2 eller för server hellre U.2, men det är en annan historia.