

Dirty cache



Proučavanje naredbe `blockdev` pružilo nam je priliku da naučimo što je prljavi bit. Korak dalje, pa ćemo se susresti i s pojmom "dirty cache". Keširanje je način da se ubrza rad računala. Umjesto zapisivanje na spore diskove, podaci se privremeno sklanjaju u bržu memoriju. No ako s jedne strane dobijamo na brzini, s druge strane unosimo komplikacije i povećavamo rizik da se dogodi nešto nepoželjno. Možemo li očekivati da će se sve odvijati automatski, bez našeg znanja i naše intervencije?

Cache memorija je svuda: procesori imaju svoj cache, diskovi također, a i poneki disk kontroleri, opskrbljeni vlastitim baterijama, da bi sačuvali keširane podatke u slučaju nestanka struje. Dodajmo tu i operacijske sustave, datotečne sustave i poneke aplikacije, prvenstveno baze podataka, koje također sklanjaju podatke "na brzinu", da bi ih kasnije trajno pospremili. Virtualizacija donosi još jednu razinu apstrakcije, povećavajući prometnu gužvu u kojoj su moguće različite nezgode.

Sjećam se jedne situacije iz ranih 90-tih. Trvrtka za koju sam radio kao programer prodala je aplikaciju knjigovodstvenom servisu. Aplikacija se izvršavala na Unixu za PC računala (ne sjećam se više da li je to bio SCO ili Interactive Unix), a knjigovođe su radili na terminalima, spojenim na server serijskom vezom, brzinom od 9600 bitova u sekundi. Za tekstualne terminale to je bilo dovoljno. Za razvoj smo koristili Faircomove alate "četvrte generacije", c-tree kao bazu, d-tree za kreiranje ekrana za unos podataka, r-tree za izradu izvještaja. Odabrali smo te alate jer smo za njih dobili izvorni kod u programskom jeziku C, pa smo mogli dodavati svoje funkcije.. Jedna od mojih prvih intervencija na zahtjev korisnika bila je zanimljiva: knjigovodkinja je unijela temeljnicu kojom je završila godinu i nakon toga pustila bilancu na matični printer. Bilanca je izašla bez te zadnje temeljnice. Zbunjeno mi je pokazivala da je temeljnica spremljena, mogla ju je otvoriti na ekranu da mi to dokaže. Zašto onda bilanca ne štima? Problem sam brzo riješio. Nekoliko sam puta kao root na komandnoj liniji napisao `sync`, da se isprazni cache, nakon čega sam pustio bilancu na ekran terminala, umjesto na printer. Bilanca je sada štimala. Ali zašto se sve to događalo? Zadnja temeljnica ostala je u cache memoriji, aplikacija za unos podataka mogla joj je pristupiti. No za reporting je pozivala drugu aplikaciju, koja je pretraživala samo bazu zapisanu na disku, nije znala pristupiti cache memoriji prve aplikacije. Kad smo to shvatili, puštali smo reporte iz osnovne aplikacije. U međuvremenu, naučio sam gospođe iz knjigovodstva da bilancu najprije pogledaju na ekranu, prije nego je puste na pišač. Ako ne štima, neka se odlogiraju i ponovo logiraju: time bi izazvali zapisivanje podataka iz cachea na disk. Zabavno, zar ne? Tada sam se prvi puta susreo s "tamnom stranom keširanja".

Ako su nam podaci dragocjeni, najsigurnije bi ih bilo odmah zapisati na tvrdi disk, da se ne zagube u nekom od cacheva. Ali diskovi su sporiji od radne memorije, a danas je brzina sve, donosi prednost na tržištu. Kada kupci biraju procesor ili na primjer DBMS, gledat će rezultate benchmarka. Rijetki će pomisliti da veća brzina nosi i veći rizik, da bi sporija baza podataka možda mogla biti sigurnija.

Naviknuti na brzinu, ponekad ćemo se iznenaditi kad se susretnemo s nekim sporim procesom. Na bivšem poslu začudio sam se kad sam prvi puta pokrenuo shutdown Windows servera. Trajao je preko 15 minuta! Mnogo pospremanja prije nego se računalo ugasi! No bolje da je tako, nego da se neki podaci zagube. Zapravo je sve bilo logično, na serveru je bilo instalirano više aplikacija, baza, preko mreže su bili montirani vanjski diskovni sustavi. UPS je bio programiran da pokrene gašenje svih servera kad se baterija napola isprazni. S obzirom na sporost zatvaranja procesa i spremanja podataka, poželio sam ranije pokrenuti shutdown servera u slučaju nestanka struje. Za svaki slučaj. A i UPS bi trebalo redovito servisirati i kalibrirati, jer mu performanse s vremenom padaju. No nisam uspevao uvjeriti upravu da treba servisirati UPS-ove, njima je to bio nepotreban trošak. Dok radi, radi...

Drugom prilikom podaci su izgubljeni kad je dotrajala baterija na SCSI kontroleru. Naručili smo novu, ali je isporuka trajala mjesec dana. Prije nego je nova baterija stigla, nestalo je struje.

Pa što je onda "dirty cache"? Jednostavno, to je međuspremnik iz kojeg podaci još nisu pospremljeni na trajnu memoriju. Ako redovito pratite članke na portalu, dosad ste već naučili kako natjerati cache da se "očisti".

Prije svega, na tradicionalan način:

```
# sync
```

Pa onda hakerski način, upisivanjem vrijednosti u datoteku u proc filesystemu:

```
$ sudo echo 3 > /proc/sys/vm/drop_caches
```

A tu je i jedan parametar ("opcija") naredbe blockdev, što nismo spominjali u prethodnim člancima:

```
$ sudo blockdev --flushbufs /dev/sdaX
```

Postoji li razlika među ovim postupcima? Na to nisam našao definitivan i argumentiran odgovor, samo sugestiju da se sync odnosi na cache datotečnog sustava, a da blockdev --flushbufs radi na nižoj razini, gdje se odvija I/O na sam uređaj. Zato bi možda bilo dobro, ako želite biti sigurni, najprije napraviti sync, pa onda neku od ovih ostalih naredbi.

A onda sam naletio na raspravu o tome da blockdev --flushbufs neće obaviti svoj posao kako treba kad se radi o virtualnom disku, odnosno ako na hostu vrtite osim "domaćinskog" operacijskog sustava još poneku virtualnu makinu. Pa se onda savjetuje da virtualne diskove ne držite na istom fizičkom disku na kojem je host operacijski sustav!

Čini se da nam je privremeno pospremanje podataka osim dobrobiti donijelo nove glavobolje i da ga ne smijemo uzeti zdravo za gotovo. Dakle u žurbi ne griješe samo ljudi, ponekad griješe i strojevi. Errare machinarum est!

sub, 2018-06-30 20:42 - Aco Dmitrović **Kuharice**: [Linux](#) [1]

Kategorije: [Operacijski sustavi](#) [2]

Vote: 5

Vaša ocjena: Nema Average: 5 (1 vote)

story_tag: [dirty cache](#) [3]

[privremena memorija](#) [4]

[blockdev](#) [5]

[sync](#) [6]

Source URL: <https://sysportal.carnet.hr/node/1814>

Links

[1] <https://sysportal.carnet.hr/taxonomy/term/17>

[2] <https://sysportal.carnet.hr/taxonomy/term/26>

[3] <https://sysportal.carnet.hr/taxonomy/term/252>

[4] <https://sysportal.carnet.hr/taxonomy/term/253>

[5] <https://sysportal.carnet.hr/taxonomy/term/220>

[6] <https://sysportal.carnet.hr/taxonomy/term/254>