

Napredni algoritmi i strukture podataka

Write Ahead Log



Univerzitet u Novom Sadu
Fakultet Tehničkih Nauka

Write Ahead Log

- ▶ mmap je veoma koristan alat za rad sa I/O
- ▶ mmap je sistemski poziv, što znači da brigu oko sinhronizacije i Swap-a prepuštamo onome ko to radi dobro i efikasno — OS
- ▶ Izbegava stvaranje dodatne kopije bafera u memoriji (za razliku od Standardnog IO-a)
- ▶ Iz perspektive programera, čitanje pomoću mmap-irane datoteke izgleda kao normalna operacija pokazivača i ne uključuje dodatne pozive
- ▶ Dosta se koristi u dizajnu baza podataka

*Pretpostavka je da radi isto i na Windows-u, ili Windows ima svoj neki ekvivalent (kao npr. ceo svet UUID, Windows GUID)

Segmentirani WAL

- ▶ Možemo da nekako probamo da podelimo WAL datoteku
- ▶ Jedan WAL možemo podeliti u više segmenata
- ▶ Za svaki segment možemo da specificiramo veličinu
- ▶ Kada se WAL pokrene, on treba da preskenira *wa/* direktrijum, i da pokupi lokacije segmenata
- ▶ Jedna opcija je da pokupimo lokacije (putanje) do segmenata
- ▶ Ovo radimo da bi mogli lakše da im pristupimo — nema potrebe da ponovo skeniramo *wa/* direktorijum

- ▶ **ALI** vvek treba da znamo redosled segmenata!
- ▶ Ovo nam pomaže da znamo šta smo pročitali kada radimo pretragu
- ▶ Segmenti neće biti učitani u memoriju odmah
- ▶ Iz praktičnih razloga možemo da učitamo samo poslednji segment
- ▶ Segmentacijom WAL-a moramo obezbediti jednostavan način za mapiranje offset-a WAL-a (ili rednih brojeva) u segmente
- ▶ Na primer, ime svakog segmenta se dobija spajanjem unapred poznatog prefiksa (npr wal) i offeta (ili rednog broja segmenta) — npr: wal_0001.log

- ▶ Ako je potrebno da pročitamo neki segment, moramo ga naći po identifikatoru na disku, i pretražiti sapise u njemu
- ▶ Kada se WAL pokreće, potrebno je da preskeniramo taj folder i da vidimo koliko segmenata ima
- ▶ Nakon toga, možemo da formiramo strukturu unutar WAL-a sa putanjama i offset-ima za vaki segmet, radi lakšeg kasnijeg rada i pronalaska
- ▶ U memoriju WAL-a, umožemo učitati samo poslednji segment

Brisanje delova WAL-a

- ▶ Treba nam mehanizam koji će reći WAL mašineriji koji deo segmenata je bezbedno obrisati
- ▶ WAL zna putanju do *wal* direktorijuma, odakle da briše segmente — ova odluka se pokazala korisna :)
- ▶ *Low-Water Mark* ideja daje najniži indeks ili **low water mark**, pre koga se segmeti mogu obrisati
- ▶ Kratko rečeno, to je indeks u WAL-u unapred definisan, koji pokazuje koji deo WAL-a može da se obriše
- ▶ Uglavom možemo obrisati sve osim poslednjeg segmenta

Zadaci

- ▶ Nastaviti rad od prošlog termina, ili implementirati novi log koristeći isti format
- ▶ WAL treba da bude podeljen na segmente, svaki segment treba da sadrži n zapisa, gde n birate sami
- ▶ Napraviti folder *wal* i u njemu čuvati segmente WAL-a
- ▶ Segmente imenovati u formatu *prefx_offset.log* (npr. *wal_00001.log*, *wal_00002*)
- ▶ WAL prilikom pokretanja treba da preskenira *wal* folder i da zabeleži sve segmente i njihove lokacije, a u memoriju samo da učitava zadnji segment
- ▶ Prilikom pravljenja novog segmenta koristiti *mmap* OS poziv. Samo trenutno aktivan segment koji se popunjava treba da koristi *mmap* mehanizam
- ▶ WAL treba da sadrži **low water mark** index, tj. do kog indeksa je bezbedno brisati segmente. Ako nam ostane samo poslednji segment, njega treba preimenovati tako da je on prvi segment
- ▶ Napraviti funkciju koja koristi **low water mark** i iz *wal* foldera briše dopuštene segmente