Slobodan Tanasić

Text manager – indeksiranje i pretraživanje teksta

Text manager je namenjen brzom pretraživanju velike količine tekstualnih podataka korišćenjem indeksa, koji obezbeđuju značajne uštede u vremenu, lako proširivanje baze i fleksibilnost u zadavanju upita. Upiti po kojima se pretraživanje vrši se formiraju navođenjem reči i fraza uz primenu Bulovih operatora. Osnovne oblasti njegove primene su arhivski ili bibliotečki sistemi koji se često dopunjuju.

Ključne reči: indeksiranje teksta, pretraživanje teksta, rasuto adresiranje, upitni jezik.

Uvod

Prilikom rada sa tekstom u velikim arhivskim sistemima (sudstvo, biblioteke...) nailazi se na niz problema, pre svega, u pristupu tim podacima, jer se dužine datoteka mere desetinama ili stotinama MB.

Ako tražimo konkretan podatak nekim od opšte prihvaćenih algoritama za pretraživanje teksta kao Bojer-Mur ili Knut-Moris-Prat, dobićemo veoma loše rezultate – i po pitanju brzine i po pitanju tačnosti. Naime, pomenuti metodi zasnivaju na prostom poređenju onoga što tražimo sa tekstom koji je u bazi. Tako će svaka nova pretraga obrađivati sav tekst ispočetka, što je presporo za sistem koji će se svakodnevno koristi, a pošto se radi o najobičnijem poređenju karaktera, mnogi delovi teksta mogu biti preskočeni ako se naša formulacija izraza ne slaže doslovce sa onim u bazi. U ovim slučajevima, sekvencijalno pretraživanje velike količine teksta je neefikasno i neupotrebljivo, pa se mora primeniti drugačiji pristup – – indeksiranje.

Pojam indeksiranja se često vezuje samo za baze podataka. Tekstualni fajlovi se ne mogu smatrati „klasičnim bazama podataka, jer ne sadrže slogove fiksne dužine, već se u njima javlja neki broj različitih reči, ali se

Slobodan Tanasić (1979), Valjevo, Daničićeva 17, učenik 2. razreda Valjevske gimnazije

mnoge ponavljaju po nekoliko hiljada puta. Ipak, na njima se mogu primeniti neke slične tehnike. Indeks je fajl dosta manji od orginalne baze u koji se zapisuju neke dodatne informacije pomoću kojih se kasnije ubrzava pretraživanje. Zapravo, njegovim konsultovanjem dobijamo informacije o tekstu bez njegovog neposrednog čitanja. Sve datoteke koje pretražujemo moramo unapred indeksirati, i to je vremenski najzahtevniji proces. Nakon toga, željeni podaci se nalaze veoma brzo.

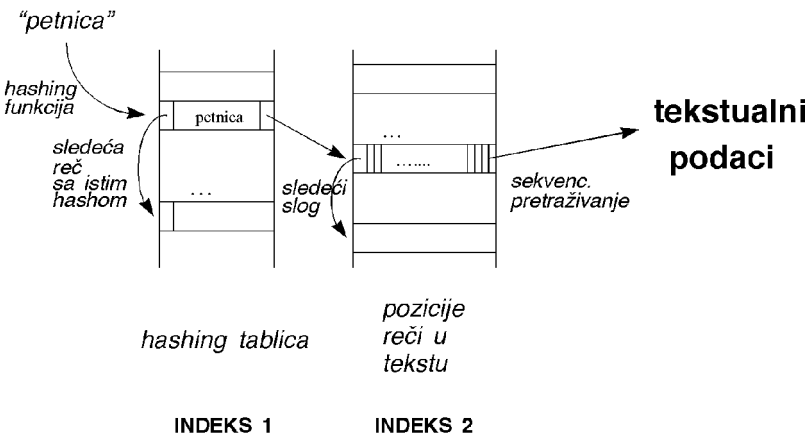
Opis rada

Text manager je zamišljen kao tzv. ,,full text search sistem, odnosno, pretraživanje je moguće vršiti po svim rečima, gotovo bez ograničenja. Ipak, ako se radi o velikim tekstualnim datotekama, realno je pretpostaviti da neće baš svi podaci iz njih biti neophodni, a ,,preskakanje nekih vodi do znatno manjih indeksa. Zbog toga je ostavljena mogućnost da se neke često korišćene ili nepotrebne reči zanemare. Text manager poseduje i relativno jednostavan upitni jezik čija je glavna namena da olakša pretragu u slučajavima kada korisnih ne može precizno da odredi traženi podatak. Upotreba indeksa omogućuje veoma laku obradu takvih izraza. Konačno, indeksne datoteke su organizovane tako da je nove podatke moguće dodati bez reindeksiranja postojećih, što omogućava nesmetan rast baze. Tako je obezbeđena i fleksibilnost u održavanju arhiva. Pri tome, indeksi ostaju dovoljno mali, tako da dok se tekst nalazi na nekom sporijem memorijskom medijumu većeg kapaciteta (CD-ROM, strimeri), oni mogu biti smešteni na hard-disku – a to neposredno utiče i na brzinu pretraživanja.

Struktura indeksnih datoteka

Ceo mehanizam indeksiranja zasnovan je na pretvaranju reči u brojnu vrednost, tzv. hashing-u. Mehanizam hashing-a obezbeđuje brz pristup uz relativno laku implementaciju, ali zahteva postojanje hash-tabele fiksne dužine, i odmah nameće problem kolizije ključeva (reči sa istom hash-vrednošću). Radi uštede prostora, koriste se dve datoteke: prva je hash-tablica, i koristi se za pristup drugoj datoteci u kojoj su podaci o konkretnom pojavljivanju reči u bazi; algoritmi i indeksiranja i pretraživanja se svode na rad sa njih dve. Da bi se rešio problem kolizije, koristi se ,,odvojeno lančanje – ako je pozicija u tabeli zauzeta, odvoji se nova na kraju fajla, a u zauzetu se doda pointer na nju. Na taj način se formira lanac reči sa istim hash-vrednostima. Pošto je za arhivske podatke veoma značajna mogućnost rasta indeksa, to je omogućeno ,,ulančavanjem i druge datoteke.

Rad sa indeksnim datotekama je šematski prikazan na primeru dodavanja nove reči u indeks (slika 1).



Slika 1. Šematski prikaz rada sa indeksnom datotekom

Algoritam pretraživanja je sličan – umesto zapisivanja podataka, oni se čitaju iz indeksa.

Da bi se dužina indeksa delimično smanjila, informacije koje se pamte u drugoj datoteci samo približno određuju položaj reči. Informacije o položaju reči se kodiraju binarnm nizom, u kome N-ti bit setovan označava da se reč pojavljuje u N-tom „delu baze, pri čemu se svi fajlovi u bazi fiktivno podele na nekoliko približno istih delova koji se zatim numerišu. Problem nastaje ako se u uslovu pretrage nalazi više reči u frazi, a one moraju biti nađene baš u tom obliku. Presekom (tj. AND-ovanjem) binarnih nizova dobijenih za svaku pojedinačnu reč dobijamo mesta u bazi na kojima se ta fraza možda pojavljuje; dalje je neophodno izvršiti sekvecnijalno pretraživanje, ali znatno manjeg dela teksta. Konkretno, Text manager-u se služi Bojer-Mur algoritmom.

Mane ovakvog pristupa se ogledaju, pre svega, u velikoj dužini indeksnih datoteka za male tekstove. Sa porastom baze hash-tabela se polako popunjava velikim brojem različitih reči tako da će nakon određenog vremena prestati sa rastom. Podaci u drugoj datoteci su linearno zavisni količine podataka koji se indeksiraju, jer dužina binarnog niza zavisi od broja „delova baze.

Sve u svemu, ovaj princip indeksiranja, zajedno sa keširanjem često korišćenih slogova (koje je takođe implementirano) daje performanse koje je teško dostići nekim drugim metodima. Problemi koji mogu nastati korišćenjem hash funkcija su brojni, i uglavnom se svode na odabir one sa najboljom raspodelom, ali su razmotreni u literaturi [1], i van su opsega ovog rada.

Upiti

Upiti se koriste za definisanje uslova pretrage. Text manager upitni jezik je jednostavan, ali se pomoću njega mogu formirati veoma složeni izrazi. Najkraće rečeno, on omogućuje kombinovanje fraza, tj. delova teksta koji se moraju naći onako kako su uneti, Bulovim operatorima AND, OR i NOT (svaki sa očiglednim značenjem). Za ograničavanje fraza se

koristi jednostruki navodnik ( ). Svaki od operatora ima svoj prioritet, pa je dozvoljeno i korišćenje zagrada, npr:

Istrazivacka stanica Petnica AND ((racunari OR informatika) AND NOT psihologija))

Prvi korak je proveravanje sintakse ovakvog izraza, i njegova konverzija u postfiksu noraciju (RPN) koja je pogodnija za dalju obradu. Interpretiranje upita se sada svodi na čitanje informacija o položaju fraza u tekstu iz indeksa i primenu osnovnih bit-operacija. Kao rezultat se dobija niz potencijalnih pojavljivalja, pa se do konačne liste pogodaka dolazi sekvencijanim pretraživanjem tih delova baze. Postupci prevođenja infiksne u postfiksnu notaciju i interpretiranja RPN izraza su detaljno opisani u [1].

Rezultati testova

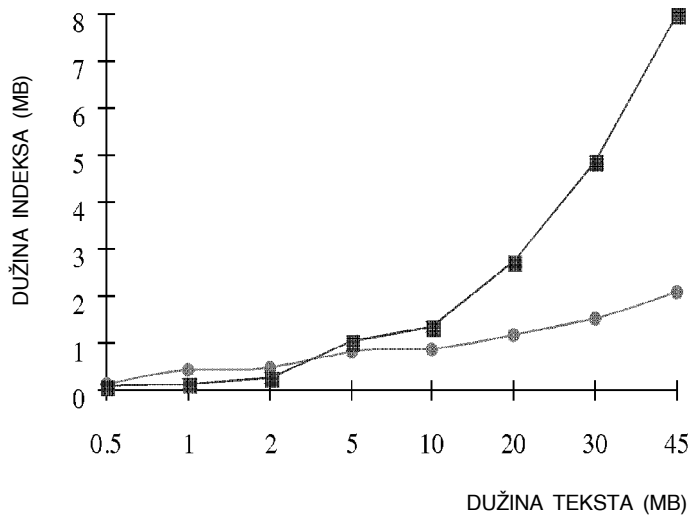
Izvršeni testovi brzine indeksiranja i pretraživanja za različite dužine teksta i indeksnih datoteka su, u celini, potrvdili moje pretpostavke. Program je testiran pod Linux-om (verzija kernela 1.2.13), na 486DX4/100 računaru sa Quantum diskom od 1.7GB. Za merenje vremena je korišćen utility program time . Indeksirano je ukupno 45 MB različitih tekstualnih podataka, mahom README, MANUAL i DOC fajlova, kao i dve knjige u elektronskoj formi.

Brzina indeksiranja date su u tabeli 1. Za različite dužine teksta, indeksi su svaki put ponovo generisani. Kada se podaci dodaju na postojeću bazu, to je najčešće u vidu nekoliko novih dokumenata čija je dužina najviše stotinak KB; ta operacija je gotovo transparentna za korisnika, tako da posebno merenje vremena u tom slučaju nije neophodno.

Utrošeno vreme je, uglavnom, linearno zavisno od veličine teksta koji se indeksira. Brzina od 150KB teksta za sekundu (ili 1 MB za nepunih 7 sekundi) je dovoljno dobra za dnevno ažuriranje baze (dodavanjem par novih fajlova), ali proces reindeksiranja ipak traje primetno vreme.

Tabela 1. Brzine indeksiranja

Dužina indeksiranog teksta (MB)	Korisničko vreme (s)	Sistemsko vreme (s)	Ukupno vreme
0.5	2.21	0.27	00:02.51
1	5.31	0.36	00:05.95
2	10.94	0.96	00:12.49
5	26.96	12.88	00:57.32
10	51.61	21.46	01:46.72
20	119.65	44.44	03:12.98
30	266.97	68.78	06:34.54
45	495.72	183.68	12:25.57



Slika 2. Krtive rasta indeksnih datoteka.

Neravnomerna zavisnost dužine indeksa za male dužine teksta je posledica strukture indeksnih datoteka, tako da one mogu zauzimati više od stvarno potrebnog prostora. To je cena koja se plaća mogućnošću dodavanja podataka u indeks, ali gubici retko prelaze par procenata. Prosek je 30% orginalnog teksta, ali za veće baze ta brojka pada na 15-20%.

Brzina pretraživanja je testirana na fajlovima ukupne veličine 45 MB (indeksne datoteke su težile oko 7MB). Najveći problem bio je izabrati podesne fraze za test – na kraju sam odlučio oko 40 različitih fraza klasifikujem po broju pogodaka.

Tabela 2. Brzina pretraživanja

Broj pogodaka	Broj testiranih fraza	Prosečno vreme pretraživnja (s)
0-100	11	1.19
100-200	7	3.20
200-500	12	11.81
500-1000	9	18.16
1000-2000	5	26.75

U vreme pretrage uračunat i ispis rezultata (izlaz je bio preusmeren u fajl). Pošto veliki broj pogodaka ne znači uvek i veliki utrošak vremena (npr. svi pogoci su u jednom fajlu), raspodela fraza je izvršena tako da se ipak primeti razlika u srednjem vremenu. Ukupno prosečno vreme bilo je ispod 15 sekundi.

Zaključak

Rezultati testova su opravdali očekivanja i pokazali da je Text manager sasvim funkcionalan program. U radu sa arhivskim podacima, za oko 25% više prostora dobija se velika brzina pretraživanja uz mogućnost proširenja tekstualne baze. Vreme potrebno za takva dopunjavanja je svega nekoliko sekundi.

Primarni pravac u daljem razvoju programa je prelazak na klijent-server organizaciju, gde bi postojao tekst-server koji smešta tekst, vrši indeksiranje i odgovara na upite klijenata. Moguće je unaprediti upitni jezik dodavanjem novih konstrukcija, kombinovano sa novim indeksnim datotekama (npr. lista tema, autora...).

Literatura

[1] Jocković, Miroslav B. 1992. *Uvod u strukture podataka*. Beograd: Institut za nuklearne nauke „Boris Kidrič", Vinča

[2] Floyd, Edwin T. 1990. *An Existential Dictionary*. Dr. Dobb s Journal, (November 1990):

Slobodan Tanasić

Text manager – Text Indexing and Searching Program

Text manager is intended for fast searching of large textual databases using indexes, that provide important savings of time, easy database expansion and flexibility in compounding a query. Queries used in searches are created by combining words and phrases, that should be found, with Boolean operators. Main areas of its usage are archive and library systems often fill in with new data.

Keywords: text indexing, full text search, hashing, query language.