FIZIČKA ORGANIZACIJA PODATAKA

Gordana Pavlović-Lažetić
Matematički fakultet, Beograd
šk. 2018/19.

Fizička organizacija podataka

- Fizička reprezentacija podataka
- Metode pristupa podacima i metode obrade podataka
- Cilj: Minimizacija broja pristupa disku pri radu sa podacima

- Različite fizičke organizacije pokazuju se manje ili više efikasnim u različitim aplikacijama i pri raznim radnim opterećenjima

- Strukture podataka i algoritmi: pregled različitih fizičkih organizacija
- Sekvencijalna organizacija
- Indeks-sekvencijalna organizacija
- Indeksna organizacija
- Direktna (heš) organizacija

Strukture podataka i pristupne metode

- Unutrašnji nivo ANSI/SPARC arhitekture sistema baza podataka

- Veliki broj *fizičkih slogova* različitih tipova
- Sa tačke gledišta SUBP, grupisani u *datoteke* koje imaju jedinstvene identifikatore

- Fizički slog odgovara logičkom slogu (n-torci) koji predstavlja, ima jedinstveni identifikator i sastoji se od niza bajtova
- Informacija o tipu fizičkog sloga, dužini sloga u bajtovima, oznake i vrednosti *fizičkih polja*.

- Zadatak SUBP je da zahteve koje korisnik postavlja na konceptualnom nivou nad relacijama i *n*-torkama preslika, na unutrašnjem nivou, u zahteve nad datotekama i fizičkim slogovima

Strukture podataka i pristupne metode

- Zahtevi koje SUBP može da proizvede na unutrašnjem nivou su sledeći:

1. pronaći fizički slog *s* iz datoteke *d*;
2. izmeniti fizički slog *s* u datoteci *d*;
3. dodati novi fizički slog datoteci *d* i dobiti njegov identifikator;
4. izbrisati fizički slog *s* iz datoteke *d*;
5. kreirati novu datoteku *d*;
6. ukloniti datoteku *d*.

Strukture podataka i pristupne metode

- SUBP u svojstvu korisnika operativnog sistema koristi funkcionalnost *upravljača tekama* (engl. file manager), a upravljač tekama svoju funkcionalnost izgrađuje koristeći funkcije *upravljača diskom* (engl. disk manager)

- ***Upravljač diskom*** ima informacije o svim fizičkim adresama na disku, i odgovoran je za sve fizičke ulazno/izlazne operacije sa diskom.

- Za ***upravljač diskom*** fizički slog predstavlja samo niz bajtova na specifičnim adresama.

- ***Upravljač tekama*** "vidi" disk kao skup blokova fiksne veličine koji se zovu *stranice*.

Strukture podataka i pristupne metode

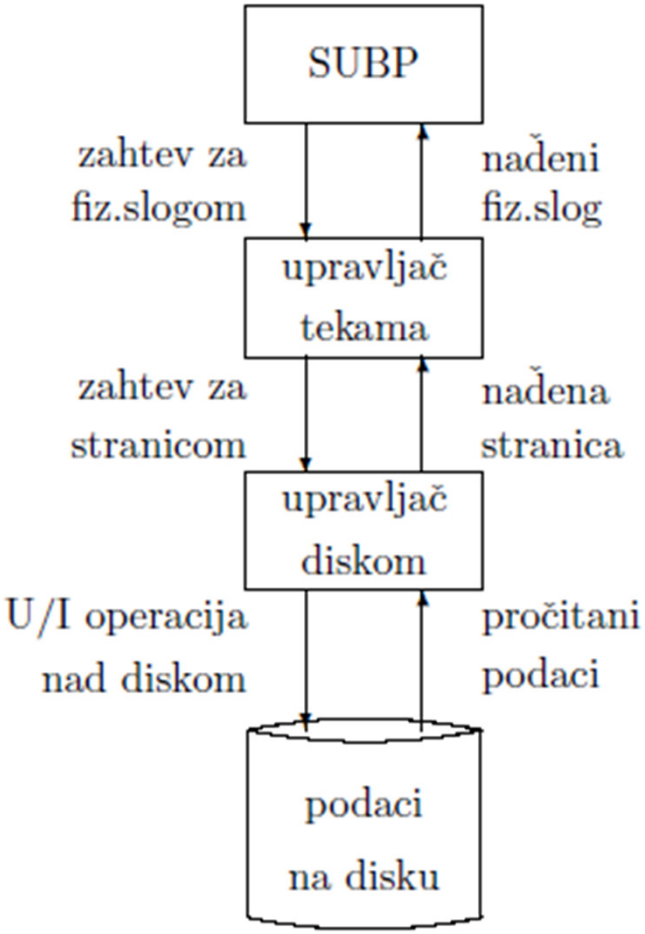
- Stranice su numerisane tako da linearno uređenje skupa stranica odgovara njihovom fizičkom redosledu.
- Veličina stranice je obično izražena brojem bajtova (4K- 32K).

- Stranice su logički grupisane u *grupe stranica* od kojih svaka ima svoj redni broj (tj. identifikator).

- Stranice (odnosno grupe stranica) su, s tačke gledišta upravljača tekama, dalje grupisane u disjunktne *stranične skupove* (engl. page sets).
- Svaki stranični skup ima jedinstveni identifikator.

- Jedan stranični skup može da sadrži fizičke slogove jednog ili raznih tipova, odnosno može da uključi i fizičke slogove koji, s tačke gledišta SUBP, pripadaju različitim datotekama.

Strukture podataka i pristupne metode



Strukture podataka i pristupne metode

- ***Upravljač diskom*** preslikava stranicu u fizičke adrese na disku koje odgovaraju toj stranici

- ***Upravljač tekama*** može da postavi (a ***upravljač diskom*** da izvrši) zahteve:
  - 1. pretražiti stranicu *p* iz straničnog skupa *st*;
  - 2. izmeniti stranicu *p* u straničnom skupu *st*;
  - 3. dodati novu stranicu straničnom skupu *st* i dobiti broj te stranice, *p*;
  - 4. ukloniti stranicu *p* iz straničnog skupa *st*.

- SUBP mora da "zna" za postojanje stranica / veći broj njih izgrađuje svoj sopstveni upravljač tekama, da bi izborom odgovarajuće fizičke reprezentacije povećao efikasnost pristupa podacima

Fizička reprezentacija relacije

- Fizička reprezentacija jedne relacije uključuje *datoteku podataka* (DP), koja sadrži fizičke slogove koji odgovaraju podacima koji se "vide" na logičkom nivou (npr. *n*-torkama relacije).

- Da bi se postigla efikasnost obrade podataka u datoteci podataka, u zavisnosti od prirode podataka i vrste obrade, fizička reprezentacija može da uključi i druge datoteke
  - *datoteku promena*,
  - *datoteku prekoračenja*
  - *indeksne datoteke*

- Uz odgovarajuće algoritme obrade podataka, dolazi se do različitih fizičkih organizacija podataka – sekvencijalne, indeks-sekvencijalne, indeksne, direktne.

Efikasnost fizičke organizacije

- Izražava se vremenom obrade traženog fizičkog sloga
  - vreme pristupa grupi stranica,
  - vreme pristupa stranici unutar grupe
  - vreme donošenja odabrane stranice sa diska u unutrašnju memoriju (vreme upisa stranice iz unutrašnje memorije na odabranu poziciju)
  - vreme obrade sloga u unutrašnjoj memoriji.
- Vreme pristupa grupi stranica i vreme pristupa stranici unutar grupe zavisi od karakteristika uređaja (brzina pomeranja glava diska)
- Vreme čitanja, odnosno upisa stranice smatraju se jednakim za sve stranice na disku.

- Jedina mera efikasnosti pojedine fizičke organizacije, u centralizovanom slučaju, biće broj stranica koje je potrebno upisati, odnosno pročitati sa diska, u realizaciji upita pri toj fizičkoj organizaciji.

Primer

- Neka datoteka podataka sadrži *n* slogova grupisanih u stranice, od kojih svaka može da primi po *m* slogova, dakle *n/m* stranica.
- Neka je vreme čitanja, odnosno upisa pojedine stranice jednako *t*.
- Neka je potrebno obraditi sve slogove iz te datoteke podataka.
- Ako se slogovima pristupa serijski, za obradu svih slogova potrebno je vreme (*n/m*)**t*, a vreme obrade pojedinačnog sloga ((*n/m*)**t*)/*n* = *t/m*.
- Ako postoji mogućnost direktnog pristupa slogu (po nekom svojstvu tog sloga), i ako slogovi nisu grupisani po tom svojstvu, za obradu cele datoteke podataka potrebno je vreme *n*t*, tj. vreme obrade pojedinačnog sloga pri ovakvom pristupu je *n*t/n* = *t*.

Primer

- Zaključak: kada je potrebno obraditi veliki broj slogova iz datoteke podataka, vreme obrade po slogu, pa i celokupno vreme obrade svih slogova daleko je manje u slučaju da se slogovima pristupa sekvencijalno – u fizičkom redosledu stranica, nego u slučaju da je pristup stranicama slobodan ("preko reda").

Indeksna datoteka

- Indeksna datoteka, u oznaci ID, sa logičke tačke gledišta odgovara funkciji koja za datu vrednost iz domena skupa atributa određuje skup slogova u datoteci podataka DP sa datom vrednošću tih atributa (ili neki nadskup tog skupa slogova).
- $A \subseteq \{A1, A2, \ldots, An\}$ indeksna datoteka ID predstavlja logički sledeće preslikavanje:

$$\mathrm{ID} : dom(\mathrm{A}) \rightarrow \mathcal{P}(\mathrm{DP})$$

Indeksna datoteka

- kod indeks-sekvencijalne reprezentacije
  - nad primarnim ključem relacije i predstavlja (logički) tabelu koja sadrži uređene parove oblika
  - (redni-broj-stranice, maksimalna-vrednost-ključa-u-stranici)

- indeks kod indeksne reprezentacije
  - nad bilo kojim atributom relacije ili kombinacijom atributa
  - logički, predstavlja tabelu u kojoj je određenoj kombinaciji vrednosti indeksnih atributa pridružen skup rednih brojeva stranica u kojima se nalaze slogovi sa tom kombinacijom vrednosti indeksnih atributa
  - sadrži uređene parove oblika
  - (vrednost-indeksnih-atributa, {r-br-stranice1,..., r-br-stranice*k* })

Algoritam direktnog pristupa

- Za pristup slogovima datoteke podataka: algoritam direktnog pristupa –“heš” algoritam (engl. hash)
- Transformiše vrednost nekog atributa (atribut direktnog pristupa, heš atribut) u redni broj stranice sloga sa datom vrednošću tog atributa
- Algoritam direktnog pristupa zadaje se analitički
- Heš funkcija: funkcija *f* koja preslikava domen atributa direktnog pristupa (ili proizvod domena skupa atributa direktnog pristupa) u datoteku podataka, tj. u odgovarajući stranični skup.

Algoritam spoljašnjeg sortiranja balansiranim spajanjem

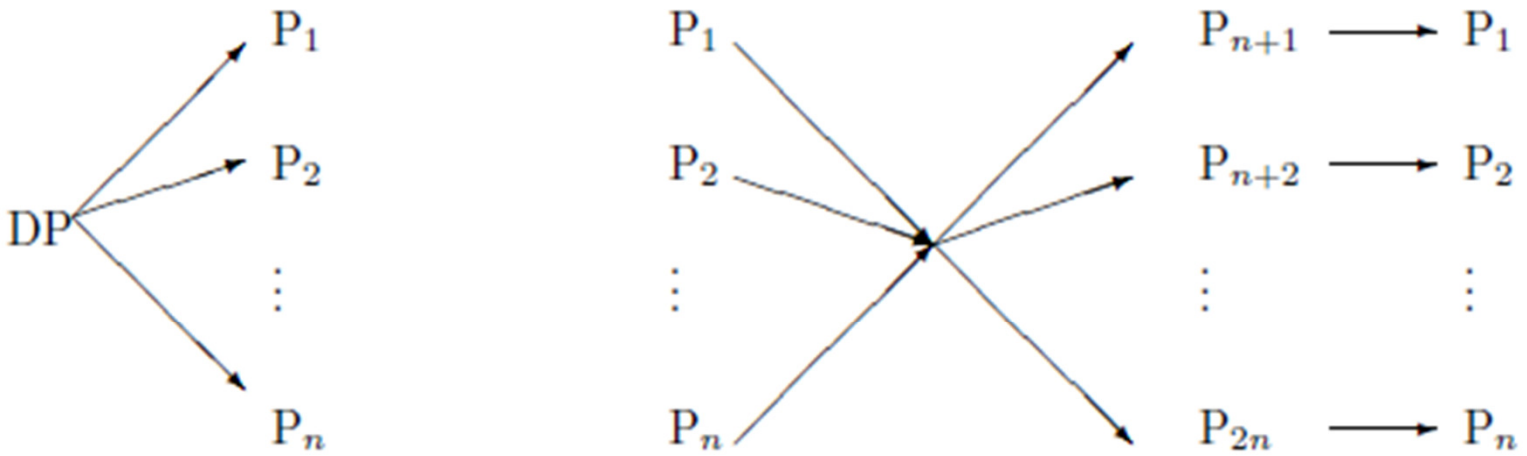
- Neuređena sekvencijalna datoteka podataka DP na disku – sekvencijalan upis, pristup
- Atribut sortiranja A (ili kombinacija atributa)

- Algoritam sortiranja:
  - unutrašnje sortiranje delova datoteke DP i
  - spoljašnje spajanje tih sortiranih delova

- Dva skupa sekvencijalnih pomoćnih datoteka, P1, P2, . . . , Pn i Pn+1, Pn+2, . . . , P2n.

- U unutrašnjoj memoriji moguće je sortirati po m slogova

Algoritam spoljašnjeg sortiranja balansiranim spajanjem



a) početna distribucija

b) spajanje, distribucija i prepisivanje

Algoritam spoljašnjeg sortiranja balansiranim spajanjem – primer 1

- DP sadrži slogove sa vrednostima atributa sortiranja
- 1, 5, 31, 7, 9, 3, 13, 6, 17, 8, 25, 27, 2, 10

- Unutrašnja memorija može da sortira tri sloga

- Na raspolaganju je šest pomoćnih sekvencijalnih datoteka

Algoritam spoljašnjeg sortiranja balansiranim spajanjem – primer 1

početna distribucija:
P1: 1, 5, 31, 8, 25, 27
P2: 3, 7, 9, 2, 10
P3: 6, 13, 17

spajanje i distribucija:
P4 : 1, 3, 5, 6, 7, 9, 13, 17, 31
P5 : 2, 8, 10, 25, 27
P6 : –

prepisivanje drugog u prvi pomoćni skup:
P1 : 1, 3, 5, 6, 7, 9, 13, 17, 31
P2 : 2, 8, 10, 25, 27
P3 : –

spajanje i distribucija:
P4 : 1, 2, 3, 5, 6, 7, 8, 9, 10, 13, 17, 25, 27, 31
P5 : –
P6 : –

prepisivanje drugog u prvi pomoćni skup:
P1 : 1, 2, 3, 5, 6, 7, 8, 9, 10, 13, 17, 25, 27, 31
P2 : –
P3 : –
DP := P1.

Algoritam spoljašnjeg sortiranja balansiranim spajanjem – primer 2

- DP sadrži slogove sa vrednostima atributa sortiranja
- 1, 5, 31, 7, 9, 3, 13, 6, 17, 35, 37, 38, 2, 10

- Unutrašnja memorija može da sortira tri sloga

- Na raspolaganju je šest pomoćnih sekvencijalnih datoteka

Algoritam spoljašnjeg sortiranja balansiranim spajanjem – primer 2

početna distribucija:
P1: 1, 5, 31, 35, 37, 38
P2: 3, 7, 9, 2, 10
P3: 6, 13, 17

spajanje i distribucija:
P4 : 1, 3, 5, 6, 7, 9, 13, 17, 31, 35, 37, 38
P5 : 2, 10
P6 : –

prepisivanje drugog u prvi pomoćni skup:
P1 : 1, 3, 5, 6, 7, 9, 13, 17, 31, 35, 37, 38
P2 : 2, 10
P3 : –

spajanje i distribucija:
P4 : 1, 2, 3, 5, 6, 7, 9, 10, 13, 17, 31, 35, 37, 38
P5 : –
P6 : –

prepisivanje drugog u prvi pomoćni skup:
P1 : : 1, 2, 3, 5, 6, 7, 9, 10, 13, 17, 31, 35, 37, 38
P2 : –
P3 : –

DP := P1.

Algoritam spoljašnjeg sortiranja balansiranim spajanjem

- Algoritam sortiranja balansiranim spajanjem
- ulaz: sekvencijalna datoteka DP, atribut sortiranja A;
- izlaz: sortirana datoteka DP;
- pomoćne strukture: sekvencijalne datoteke P1, . . . , P2*n*;
- BEGIN
  - izvršiti *početnu distribuciju* sekvenci iz DP u P1, . . . , P*n*;
  - REPEAT
    - izvršiti spajanje sekvenci iz datoteka P1, . . . , P*n* i njihovu distribuciju u datoteke P*n*+1, . . . , P2*n*;
    - prepisati datoteke {P*n*+1, . . . , P2*n*} u { P1, . . . , P*n*}
    - isprazniti datoteke {P*n*+1, . . . , P2*n*}
  - UNTIL u P1 samo jedna sekvenca, a ostale pomoćne datoteke prazne;
  - prepisati sadržaj datoteke P1 u datoteku DP
- END.

Algoritam spoljašnjeg sortiranja balansiranim spajanjem - složenost

- Vreme potrebno za sortiranje balansiranim spajanjem sekvencijalne datoteke DP sa n slogova je proporcionalno sa n^2.

- Postoje poboljšanja ovog algoritma koja se pre svega odnose na broj potrebnih pomoćnih datoteka, ali se složenost algoritma ne može bitno popraviti.

Sekvencijalna organizacija - jednorelacioni upiti

- Neuređena – gomila (heap): DP
- Uređena – sekvencijalna: DP, DPR

- Efikasnost izvršavanja jednorelacionih upita
- *Pretraživanje(DP, DPR)* pojedinačnog sloga (n slogova, m na stranici)*: O*(*t*n/m*)
- Organizacija neadekvatna
- Primer:
- SELECT STATUS, DRZAVA
- FROM I
- WHERE I_SIF = 'i2'

Sekvencijalna organizacija - jednorelacioni upiti

- Obrada svih slogova (DP, DPR) u proizvoljnom poretku: $O(t*n/m)$
- Adekvatna organizacija
- Primer
- SELECT NAZIV, STATUS
- FROM I

- Primer: grupna obrada sa restrikcijom
- SELECT NAZIV
- FROM I
- WHERE STATUS >= 20 AND DRZAVA = ’Srbija'

Sekvencijalna organizacija - jednorelacioni upiti

- Grupna obrada slogova u zadatom redosledu zadatog atributa:
  - Sortiranje DP datoteke u neuređenoj varijanti, odnosno DP i DPR datoteka (zajedno), u uređenoj varijanti, u zadatom redosledu zadatog atributa.
  - Uređena varijanta ima značajne prednosti ako je DP već sortirana po atributu koji određuje redosled
- Primer:

```
SELECT NAZIV, STATUS
FROM I
ORDER BY NAZIV
```

Efikasnost pretraživanja odražava i efikasnost brisanja odnosno ažuriranja

Sekvencijalna organizacija - jednorelacioni upiti

- Ažuriranje, brisanje i unošenje
  - ažuriranje i brisanje sloga analogno je izboru
  - vreme potrebno za ažuriranje i brisanje odgovara vremenu potrebnom za izbor
  - umesto fizičkog brisanja, slog se obeležava nevažećim
  - vreme uključuje i vreme periodične reorganizacije

- Unošenje novog sloga realizuje se kao hronološko dodavanje sloga DP odnosno DPR;
- u slučaju uređene varijante, vreme unošenja uključuje i vreme povremene reorganizacije, izračunato po pojedinačnom slogu

Sekvencijalna organizacija - višerelacioni upiti

- Višerelacioni upiti uključuju neki oblik operacije spajanja (uz restrikciju i projekciju)

- Osnovni algoritam izvršavanja dvorelacionih upita (analogno višerelacionih) ima dva globalna koraka:
  - *1. Sortirati* obe DP (i DPR, ako postoje) po atributima spajanja; u prvom prolazu kroz datoteke pri sortiranju izvršiti restrikcije i projekcije na izlazne atribute i atribute spajanja;
  - *2. Izvršiti sekvencijalno prirodno spajanje* sortiranih datoteka, uz istovremenu projekciju na izlazne atribute.

- Za relacije koje ne učestvuju u prvom koraku (npr. kod uređene varijante po atributu spajanja), pripadne restrikcije vrše se u drugom koraku, istovremeno sa spajanjem.

Algoritam sekvencijalnog prirodnog spajanja

- Algoritam sekvencijalnog prirodnog spajanja po atributima sortiranja
- ulaz: sortirane datoteke P1 i P2;
- izlaz: sortirani rezultat operacije prirodnog spajanja datoteka P1 i P2;

```
BEGIN

...

END
```

Algoritam sekvencijalnog prirodnog spajanja

```
BEGIN
  otvoriti datoteke P1 i P2 za čitanje;
  pročitati prvi slog s1 datoteke P1 i prvi slog s2 datoteke P2;
    REPEAT
      uporediti vrednosti atributa spajanja slogova s1 i s2;
      IF vrednost atributa spajanja sloga s1 manja od vrednosti atributa
         spajanja sloga s2
      THEN  BEGIN
                čitati slogove iz P1 dok imaju istu vrednost atributa
spajanja;
                pročitati sledeći slog s1 datoteke P1
                END
```

Algoritam sekvencijalnog prirodnog spajanja

ELSE IF vrednost atributa spajanja sloga *s*2 manja od vrednosti atributa spajanja sloga *s*1
THEN BEGIN
čitati slogove iz P2 dok imaju istu vrednost atributa spajanja;
pročitati sledeći slog *s*2 datoteke P2
END
ELSE BEGIN
(jednaki su) spojiti sve slogove iz datoteka P1, P2 sa istom vrednošću atributa spajanja kao *s*1 (tj. *s*2);
pročitati sledeći slog *s*1 datoteke P1 i sledeći slog *s*2 datoteke P2
END
UNTIL kraj jedne od datoteka P1, P2
END.

Višerelacioni upit - primer

```
SELECT NAZIV, K_SIF
FROM I JOIN KI ON I.I_SIF = KI.I_SIF
WHERE TIRAZ > 3000
```

- I, KI fizički organizovane sekvencijalno

- prvo se sortiraju
  - DP (i DPR, ako postoji) relacije I, uz projektovanje na atribute I_SIF, NAZIV
  - i DP relacije KI, uz restrikciju “TIRAZ > 3000” i projekciju na atribute K_SIF, I_SIF
- zatim se rezultati ovih operacija sekvencijalno prirodno spoje uz projekciju na atribute NAZIV, K_SIF

Sekvencijalna organizacija - višerelacioni upiti

- Alternativni algoritam spajanja: metod ugnježdenih petlji
- Izbor zavisi od učestalosti promena sekvencijalne datoteke

- Prednost uređene varijante u izvršavanju dvorelacionih (i višerelacionih) upita postoji samo u slučaju da je atribut spajanja baš atribut sortiranja bar jedne od relacija koje učestvuju u tom upitu
- U slučaju sekvencijalne reprezentacije potrebna su česta sortiranja, čak i kada se radi o uređenoj varijanti, s obzirom da je u njoj jedan od atributa privilegovan