Tehnici de multiindexare

Prof. dr. Ion SMEUREANU, asist. Marian DÂRDALĂ,
Catedra de Informatică Economică, A.S.E., Bucureşti

Articolul prezintă aspecte privind implementarea structurii de grid file pentru indexarea datelor multidimensionale.
***Cuvinte cheie**: grid file, partiţie grilă, director grilă, scală liniară*

Multiindexarea este una din metodele de organizare a datelor care permite regăsirea informaţiilor după mai multe chei. In funcţie de implementare, multiindexarea tratează diferenţiat sau nediferenţiat caracteristicile alese drept chei de indexare. Ne vom ocupa în continuare de o tehnică de multiindexare-*grid file,* care tratează nediferenţiat cheile de selecţie. Ea se bazează pe o reprezentare intuitivă a spaţiului *n*-dimensional al caracteristicilor, căreia i se asociază o hartă de biţi, matrice *n*-dimensională, semnalând prezenţa sau absenţa înregistrărilor care îndeplinesc cumulativ condiţiile de selecţie. Dimensiunea *bitmap*-ului va fi dată, deci, de numărul de atribute de identificare a înregistrărilor.

Presupunem un exemplu în care se doreşte identificarea înregistrărilor după trei atribute. Se va construi un masiv tridimensional, în concordanţă cu spaţiul atributelor (*S1,S2,S3*).

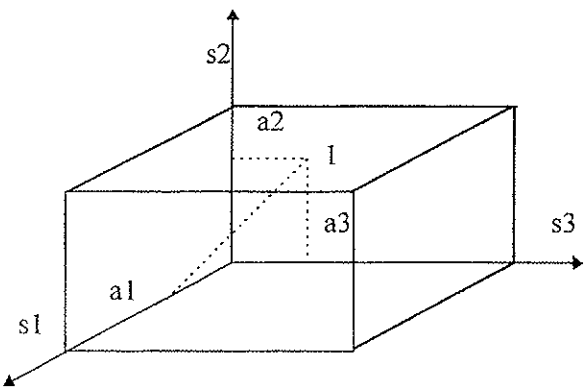

Fig. 1 Bitmap într-un spaţiu tridimensional

Dacă se doreşte găsirea înregistrării cu atributele (*a1,a2,a3*) se consultă bitul corespunzător masivului tridimensional B[*a1*][*a2*][*a3*] şi în caz că are valoare 1, înregistrarea există. Generalizând, se pot face operaţii de căutare a unei înregistrări, ceea ce se reduce la accesul direct în masiv a elementului [*a1*][*a2*][*a3*]...[*ak*]. Inserarea, respectiv ştergerea presupune poziţionarea bitului pe valoarea 1, respectiv 0. Această metodă presupune ocuparea unui spaţiu foarte mare de memorie, datorită varietăţii mari a valorilor de indexare, dezavantaj ce poate fi ameliorat aplicând diverse tehnici de compactare a masivelor.

Alt neajuns constă în faptul că în acest mod se verifică dacă există sau nu înregistrări care satisfac nişte atribute, dar nu se precizează exact poziţia lor în fişier. Pentru a elimina dezavantajul, se propune într-o primă etapă partiţionarea spaţiului căutării, prin alegerea unei diviziuni a domeniului. Alegerea diferitelor partiţii depinde de distribuirea datelor, ce se presupune a nu fi întotdeauna uniformă, în timp ce atributele aferente partiţiilor sunt independente.

O partiţionare, în cazul tridimensional, rezultă prin fixarea unor intervale pe fiecare axă (*X, Y, Z*) ce divizează spaţiul înregistrărilor sub forma:
$U = (U_0,U_1,...,U_k), V = (V_0,V_1,..., V_m)$ şi $W = (W_0, W_1, ... ,W_n)$.

În timpul operaţiilor de lucru cu fişiere, partiţiile trebuie să fie modificate în concordanţă cu inserările şi ştergerile. Partiţia $P = U * V * W$ este modificată prin alterarea unei componente la un moment de timp, datorită descompunerii unui interval în două, sau unirii a două intervale adiacente.

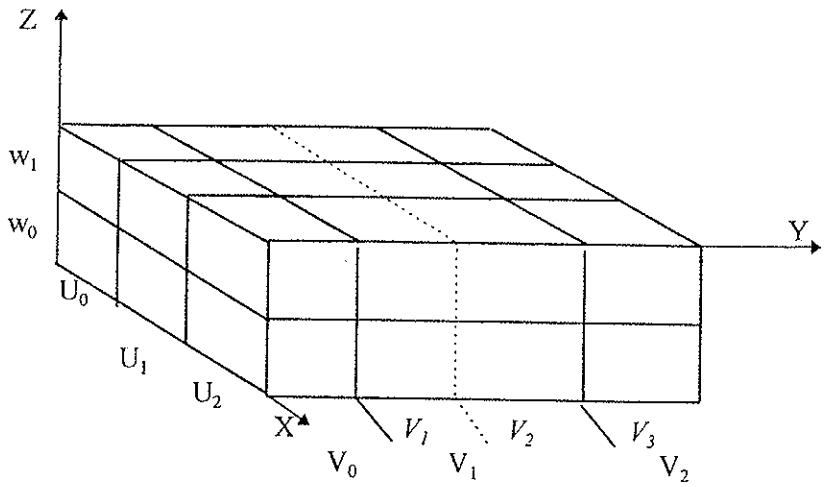

Fig. 2 Partițiile grilă

Metoda urmăreşte proiectarea unei structuri de fişier pentru stocarea şi regăsirea înregistrărilor după mai multe atribute; structura se numeşte *fişier grilă*. Unitatea de stocare a înregistrărilor este *bucket*-ul şi conține mai multe înregistrări. Pentru aceste fişiere, problema se reduce la definirea unei corespondențe între blocurile rezultate prin partiționare (blocuri grilă) şi *bucket*-uri. Această asignare o realizează un *director grilă*.

O posibilă structură de fişier grilă presupune:

- pentru o cerere punctuală să se folosească, în principal, două accese la disc;
- procesarea rapidă a cererii pe domenii mari, liniar ordonate;
- desfacerea şi unirea blocurilor să implice doar două blocuri;
- menținerea unui nivel rezonabil de umplere medie a *bucket*-ului.

Un rol important revine directorului grilă care este de fapt un sistem de management a *bucket*-ilor, ceea ce implică:

- definirea unei clase de legături a blocurilor grilă cu *bucket*-ii;
- alegerea unei structuri de date pentru director, care să conțină asignarea curentă;
- găsirea unui algoritm eficient pentru a actualiza acea structură în mod direct, la o schimbare a asignărilor.

În figura 2 este un exemplu de asignare convexă a blocurilor la *bucket*-i.

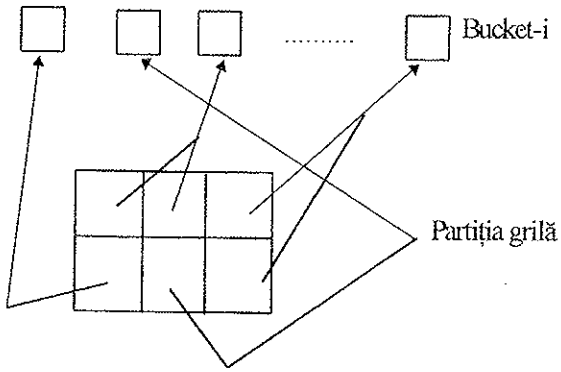

Fig. 3 Legătura între directorul grilă şi *bucket*-i

Fiecare partiție punctează un *bucket*. Există şi situații când mai multe partiții punctează pe acelaşi *bucket*.

Un director grilă conține două părți:

a) un masiv *k*-dimensional, care se numeşte masiv grilă, elementele lui fiind pointeri la *bucket*-i şi sunt într-o corespondență unu la unu cu blocurile din partiție.

b) *k* vectori, câte unul pentru fiecare atribut de indexare; un astfel de vector se numeşte *scală liniară*.

Pentru o mai bună înțelegere se va considera $k=2$, adică un spațiu al înregistrărilor, de forma $S = X * Y$. Un *director grilă* pentru un spațiu bidimensional se caracterizează prin:

- întregi $nx>0$, $ny>0$ ce precizează numărul de partiții a axei X, respectiv Y;
- întregi $0 \le cx < nx$, $0 \le cy < ny$ ce precizează elementul curent din director şi din blocul grilă curent;

- masivul bidimensional G (0.., nx-1, 0.., ny-1) numit masivul grilă;
- două masive unidimensionale X (0,...,nx) şi Y (0,...,ny), ce reprezintă câte o scală liniară aferentă fiecărei dimensiuni a spaţiului.

Operaţiile ce trebuie să se definească pe structura directorului grilă sunt:
- accesul direct $G(cx, cy)$;
- accesarea poziţiilor vecine:
  următorul / predecesorul x
  $cx = (cx \pm 1) \bmod nx$
  următorul/predecesorul y
  $cy = (cy \pm 1) \bmod ny$
- joncţiunea, care se poate face atât între două partiţii de pe axa x cât şi pe axa y. Astfel px, $1 \leq px < nx$, se uneşte cu următoarea partiţie ceea ce presupune renumirea tuturor elementelor care se află după px, cât şi ajustarea scalei x care va avea un element mai puţin.
- descompunerea se poate face ca şi operaţia precedentă pe scala x cât şi pe y. Având px, $0 \leq px \leq nx$, se va crea prin divizare un nou element px+1 şi se renumesc toate celulele ce succed lui px, scala x va avea un nou element.

Asignarea convexă a blocului grilă la *bucket* este exprimată prin următoarea relaţie: dacă $G(i', j')=G(i'', j'')$ atunci pentru toate valorile i, j cu proprietatea că $i' \leq i \leq i''$ şi $j' \leq j \leq j''$ avem:

$$G(i', j')=G(i,j)=G(i'', j'').$$

În continuare se va prezenta un exemplu de folosire a directorului grilă în accesarea unei înregistrări în funcţie de valorile a două atribute de indexare: *an* cu valorile 0-2000 şi *iniţială* cu domeniul a-z.

Se presupune că distribuţia înregistrărilor a cauzat următoarea partiţionare grilă:

x = (0,1000,1500,1750,1875,2000)

y = (a,f,k,p,t)

Având cele două scale putem căuta înregistrarea cu atributele [1980,w].

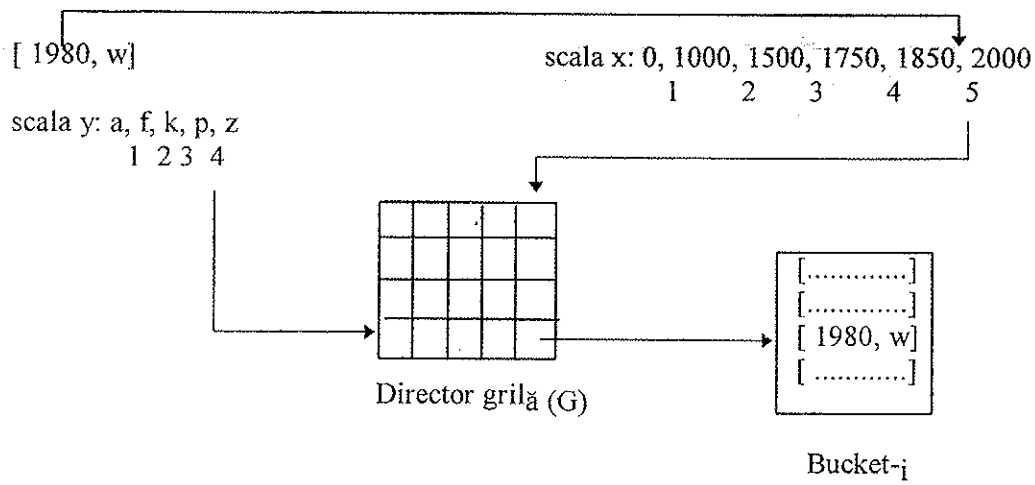

Fig. 4 Folosirea directorului grilă la identificarea *bucket*-ului pornind de la valorile cheilor

Determinarea *bucket*-ului care conţine înregistrările cu valorile precizate pentru atributele de indexare se face în următorii paşi:
- atributul 1980 este convertit în valoarea 5 din index printr-o căutare în scala x;
- atributul w este convertit în valoarea 4 din index printr-o căutare în scala y;
- indicii 4 şi 5 permit accesul direct la elementul concret al gridului director $G(4,5)$ de unde se extrage adresa *bucket*-ului ce conţine înregistrarea dorită. Crearea fişierului grilă presupune un proces dinamic de construire, sincronizat cu cea a directorului grilă. Pentru aceasta se presupune iniţial un singur *bucket* cu capacitatea de a reţine trei înregistrări. Directorul grilă va conţine, deci, un singur pointer la acest *bucket* unic.

Când *bucket*-ul *A* este plin (figura 5 a) şi se încearcă introducerea unei noi înregistrări, atunci spaţiul înregistrărilor este divizat şi un nou *bucket* este disponibilizat (figura 5 b).

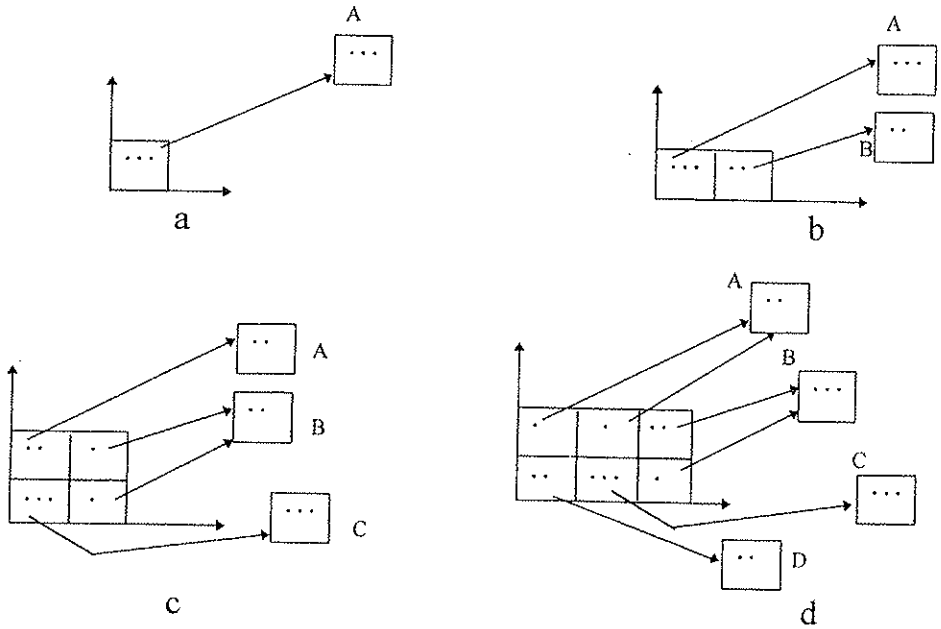

Fig. 5 Etapele de creare a fişierului grilă

Inregistrările sunt mutate din *bucket*ul vechi în cel nou, în funcţie de apartenenţa la noua partiţie ce se crează în directorul grilă. Dacă *bucket*ul *A* este din nou depăşit, blocul grilă (jumătatea stângă a spaţiului) este divizat după aceeaşi regulă (figura 5 c). Se presupune că se încarcă înregistrări şi că *bucket*ul *C*, existent, va fi incapabil de a mai stoca înregistrări. Astfel se va crea un nou *bucket* (*D*) ca în figura 5 d.

In implementarea acestui tip de fişier se urmăresc anumite probleme ce sunt legate de mediu şi ţin în mare măsură de modul de realizare a directorului grilă. El poate fi implementat ca masiv *k*-dimensional sau folosind liste înlănţuite.

Folosirea unui masiv are avantajul că accesarea elementelor se face mai uşor şi se economiseşte memorie. Dezavantajul constă în greutatea realizării operaţiei de joncţiune şi divizare.

Listele înlănţuite prezintă avantajul posibilităţii inserării, respectiv extragerii elementului din poziţii arbitrare şi deci a reconfigurării dinamice a structurii directorului grilă, dar presupun un spaţiu de memorie mai mare, necesar stocării pointerilor.

Pentru minimizarea timpului de răspuns la cereri, ultimile versiuni ale sistemelor de gestiune a bazelor de date (vezi Oracle Express Objects) au implementat şi conceptul de date multidimensionale. Structura ~~de~~ tip grid file oferă una dintre implementările cele mai peformante.

Bibliografie

1. **Freeston, M.,** The BANG file: A new kind of grid file, Proceedings of the ACM SIGMOD Conference on Management of Data, San Francisco, May, 1987.
2. **Nievergelt, J., Hinterberger, H., Sevcik, K.C.,** The Grid File: An Adaptable, Symetric Multikey File Structures, Readings in DATABASE SYSTEMS, Morgan Kaufmann Publishers, San Francisco, California, 1994.
3. **Salzberg, B.,** Grid file concurrency, Inf. Syst. 11,3,1986.