Gestione dei dati

Parte 6
Esercitazione sull'accesso ai file

Maurizio Lenzerini, Riccardo Rosati

Facoltà di Ingegneria
Sapienza Università di Roma
Anno Accademico 2012/2013

http://www.dis.uniroma1.it/~rosati/gd/

SAPIENZA
UNIVERSITÀ DI ROMA

Esercizio 1

Si consideri la relazione STATO(nome, superficie, capo). Le tuple della relazione STATO sono 7000, ed in ogni pagina entrano 20 tuple di tale relazione. Non si prevedono operazioni di inserimento o cancellazione sulla relazione STATO, mentre si sa che verrà molto spesso interrogata per calcolare tutti gli stati la cui superficie cade in un dato intervallo.

Si chiede di dire qual è il metodo migliore per rappresentare la relazione STATO, motivando la risposta. Si chiede anche di calcolare il costo medio dell'operazione che trova tutte le informazioni (nome, superfici e capo) degli stati con una certa superficie, sapendo che in media sono meno di 10 gli stati con una data superficie.

Soluzione esercizio 1

Fatto 1: **non si prevedono operazioni di inserimento o cancellazione** sulla relazione STATO

Fatto 2: si sa che la relazione STATO verrà molto spesso interrogata per calcolare tutti gli stati la cui superficie cade in un dato **intervallo**

Conseguenza: il metodo migliore per rappresentare la relazione STATO è **l'indice ISAM clusterizzato** costruito sulla search key "superficie".

Soluzione esercizio 1 (continua)

Per quanto riguarda il costo:

- sappiamo che B= 7000/20 = 350 (B=numero di pagine del file).
- chiamiamo F il numero di data entry o index entry per pagina
- segue che il numero di foglie nell'albero ISAM è 350/F
- Per calcolare il costo dell'operazione osserviamo che mediamente dobbiamo accedere solo alle pagine per localizzare il data entry, e ad una pagina del data file
- Il costo dell'operazione è quindi:

  $(\log_F 350/F + 1) =$

  $(\log_F 350 - \log_F F + 1) =$

  $(\log_F 350 - 1 + 1) = \log_F 350.$

Esercizio 2

Si consideri una base di dati contenente la relazione *Impiegato* con attributi *matricola, nome, dataNascita, citta, stipendio,* in cui le query di gran lunga più frequenti sono del seguente tipo: restituire nome, data di nascita e stipendio degli impiegati il cui stipendio è maggiore di un certo valore *k* e minore di un certo valore *h* (con *h* e *k* che variano per ogni query). Inoltre, la tabella *Impiegato* è soggetta a frequenti aggiornamenti (inserimenti e cancellazioni).

Sulla base di tali informazioni:

1. dire quale organizzazione di file è la più indicata per la relazione *Impiegato*, motivando la risposta;

2. sulla base dell'organizzazione di file scelta al punto precedente, illustrare la funzione che caratterizza il costo del calcolo della risposta a una query del tipo descritto precedentemente rispetto al numero di pagine necessarie per memorizzare la relazione.

Soluzione esercizio 2

- Fatto 1: la tabella Impiegato è soggetta ad aggiornamenti
- Fatto 2: la query più frequente corrisponde ad una ricerca per intervallo di valori
- la ricerca per intervallo richiede necessariamente o un file ordinato sulla chiave di ricerca oppure un indice di tipo clustered
- ma la presenza di aggiornamenti ci fa scartare l'organizzazione basata sul file ordinato
- tra gli indici, quelli di tipo hash in pratica non sono mai indici clustered (il problema è causato dalle pagine di overflow)
- tra gli indici ad albero, l'organizzazione da preferire è **B+-tree**, perché gli indici ISAM si degradano in presenza di aggiornamenti

Soluzione esercizio 2 (continua)

Per quanto riguarda il costo:

- assumiamo che nel caso medio le foglie del B+-tree siano occupate al 67%
- pertanto **nel caso medio** la ricerca per intervallo (rispetto al numero di pagine lette da memoria secondaria) ha costo

  $\log_F 1.5\ B$ + numero pagine contenenti record di interesse

  dove F è il fan-out del B+-tree

Esercizio 3

Si consideri una base di dati con le relazioni MUSEO(codice,codcitta) e CITTA(codcitta,numabitanti,sindaco), dove “codcitta” è una foreign key su CITTA. Sappiamo che il DBMS che gestisce tale base di dati esegue il join naturale tra relazioni di tipo R1(A,B) ed R2(B,C,D) scandendo le tuple di R1, e per ogni tupla t1 di R1 cercando la eventuale tupla t2 di R2 in join con t (ovvero, tale che t1.B=t2.B). Sappiamo anche che la relazione CITTA non è soggetta ad aggiornamenti, che le pagine della relazione MUSEO sono 100 e che ciascuna pagina contiene 100 record, e che le pagine della relazione CITTA sono 200 e che ciascuna pagina contiene 20 record.

Si chiede di:

- scegliere il metodo di rappresentazione della relazione CITTA che massimizza l’efficienza del join naturale tra le due relazioni, tenendo conto che il DBMS adotta l’algoritmo descritto sopra per eseguire l’operazione di join naturale;
- calcolare il numero di accessi a pagine necessari per l’esecuzione del join naturale (sempre mediante l’algoritmo descritto sopra) tra le due relazioni, assumendo che la relazione CITTA sia rappresentata secondo il metodo scelto.

Soluzione esercizio 3

E' evidente che per massimizzare l'efficienza dell'algoritmo occorre massimizzare la ricerca di un elemento nella relazione CITTA di cui si conosce la chiave (codcitta). Tenendo conto poi del fatto che CITTA non è soggetto ad aggiornamenti, il metodo di rappresentazione migliore per CITTA è dunque mediante un indice hash statico (ovvero: hashed file).

Il numero di accessi a pagine necessari per l'esecuzione dell'algoritmo di join naturale si calcola tenendo presente che:

- si deve accedere a tutte le 100 pagine di MUSEO
- per ogni record di MUSEO (il numero di tali record è $100 \times 100 = 10.000$) si può sfruttare l'indice hash statico e quindi operare un unico accesso alle pagine della relazione CITTA

Ne segue che il numero di accessi a pagine è:

$$100 + 10.000 \times 1 = 10.100$$